History of Flash at Toshiba

- He also understood that the market size for memory is more dependent on its bit cost than its user-friendliness. For instance, let's compare the market size between DRAMs and SRAMs.
- SRAM is faster, requires no refresh, and is very user-friendly, but the market size for DRAMs is much larger than that of SRAMs. The only reason why the DRAM market is larger is because the cost of DRAMs is much lower than that of SRAMs.
- This is the RAM story, but the ROM story is similar. Like SRAM, the byte-EEPROM is very user-friendly because it can erase and program a single byte. But its cost is so high that it cannot be widely adopted. A hard disk, which can also be considered to be a non-volatile memory, does not offer **byte reprogramming**, but does offer **sector reprogramming** (block reprogramming) and is widely used because of low cost.
- So, what is required is not the flexibility of byte reprogramming, but a low cost per bit.

History of Flash at Toshiba

- Based on the concepts above, Dr. Masuoka applied for a patent on simultaneously erasable EEPROMs in 1980. Although a conventional byte-EEPROM has **two transistors per cell**, a new memory cell which consists of **only one transistor**, was proposed to reduce cost. To realize a one-transistor cell, the **bit erase** scheme was dismissed and a **simultaneous erase** scheme was adopted.

- The development of a real test device was started in 1983 with Dr. Masuoka's colleagues: Mr. Asano and Mr. Iwahashi for the design, Mr. Tozawa, Mr. Komuro, Mr. Tanaka for the device technology, and supported by Mr. Suzuki, the memory senior manager. Fortunately, the device was verified be functional. In June of 1984, the first paper was submitted to IEDM. At that time, Dr. Masuoka recognized that it must be the first simultaneously erasable EEPROM in the world and thought about naming it with his colleagues. Mr. Ariizumi, one of his colleagues, proposed naming it "**Flash**" sometime in June of 1984, before submitting the IEDM paper. Why Mr. Ariizumi suggested the term "Flash" was because the device could erase a block simultaneously, which made him imagine the Flash of a camera. But no one, at the moment, could have dreamed that Flash memory would be used in digital cameras today. So what was first called simultaneously erasable EEPROM, was named "**Flash**" from 1984. The memory cell area for the first proposed Flash EEPROM was 64 sq. microns while a conventional **byte-erasable EEPROM cell** at that time occupied 272 sq. microns in the same lithography design rule of 2 microns.

History of Flash at Toshiba

- **In December of 1984**, the first paper for the **Flash EEPROM** was presented at IEDM in San Francisco. A subsequent the paper on a 256k bit Flash EEPROM was presented at ISSCC in San Francisco in February of 1985. After that, Dr. Masuoka was interviewed by “Business Week” and the Flash EEPROM was reported in Business Week on Mar.25, 1985. On the news, Dr. R.D. Pashley of Intel was interviewed to provide counterpoints. But afterwards, Intel stopped developing the UV-EPROM (ultraviolet erasable) and focused on Flash memory development. Dr. Pashley would become the General Manager of the Flash memory division.

- After Toshiba presented the **256k bit Flash EEPROM** at the ’85 ISSCC, Seeq developed a 128k bit Flash EEPROM and announced it at the ’87 ISSCC. Seeq’s memory cell was programmed by hot electron injection and erased by field emission from the floating gate to the drain. Therefore, Seeq’s cell could be realized by a dual polysilicon structure while Toshiba’s Flash EEPROM cell used a triple polysilicon structure due to the formation of the erase gate. Intel presented a 256k bit Flash EEPROM at the ’88 ISSCC. Intel adopted the same cell structure as that of the UV-EPROM.
- It is programmed by the hot electron injection like a UV-EPROM and erased by the field emission from the floating gate to the source. In principal, this concept is quite similar to that of the first proposed Flash EEPROM by Toshiba.

How Flash Works

- Like a UV-EPROM (ultraviolet erasable programmable read only memory) cell, a Flash EEPROM (electrically erasable programmable read only memory) cell has a dual gate structure in which a floating gate exists between a control gate and a silicon substrate of a **MOSFET**. A floating gate is perfectly isolated by the insulator, i.e. silicon dioxide, so that the injected electrons cannot leak out of the floating gate after power is removed. This is the basic storage mechanism for the Flash EEPROM non-volatile memory. The charge retention mechanism for Flash EEPROM is the same as conventional UV-EPROM and byte-erasable EEPROM. Like a UV-EPROM, a Flash EEPROM is programmed by a hot electron injection mechanism, and erased by field emission from a floating gate, like a byte-erasable EEPROM. The erase mechanism for a Flash EEPROM cell is the same as that for a byte-erasable **EEPROM** cell; however, their basic use as LSI memories are typically different. In a **Flash EEPROM**, the whole chip can be erased simultaneously, while a byte-erasable EEPROM is erased only one byte at a time. When the byte erase function is eliminated, an electrically re-programmable non-volatile memory can be realized by utilizing only one transistor per cell. A UV-EPROM also simultaneously erases all its bits by exposure to ultraviolet light, and is programmed by hot electron injection mechanism. In this sense, a UVEPROM is similar to a Flash EEPROM in functionality except that the erase operation is carried out by UV irradiation.

NAND vs NOR Flash

- Current semiconductor memories achieve random access by connecting the memory cells to the bit lines in parallel, as in NOR-type flash. In NOR-type flash, if any memory cell is turned on by the corresponding word line, the bit line goes low (see figure 1). Since the logic function is similar to a NOR gate, this cell arrangement results in NOR flash.

- However, speedy access is not always required in order to replace magnetic memory. The **NAND Flash** is a new flash configuration that reduces memory cell area so that a lower bit cost can be achieved. In 1987, Toshiba proposed the NAND Flash, and its NAND structured cell arranged as eight memory transistors in series. The NAND flash cell array, fabricated by using conventional self-aligned dual polysilicon gate technology, had only one memory transistor, one forth of a select transistor and one sixteenth of the contact hole area per bit. This technology realizes a small cell area without scaling down the device dimensions. The cell area per bit was half that of a DRAM using the same design rule of 1um (which was used for the 1M bit DRAM). As a result, Toshiba realized that it was possible for NAND Flash to be developed earlier than DRAM (for the same density) by one process generation. In comparison, conventional **EEPROM** was behind DRAM by one process generation at that time.

- The most important item regarding memories is the bit cost. In the case of a semiconductor memory, the bit cost is dependent on the memory cell area per bit. Since the cell area of NAND Flash is smaller than that of NOR Flash, NAND Flash always had the potential from the start to be less expensive than NOR Flash. However, it takes a rather long time for a NAND Flash to read out the first data byte compared to NOR Flash because of the resistance of the NAND cell array, although it is much faster than the seek time for a hard disc by several orders of magnitude. Therefore, the aim of NAND Flash is to replace hard disks.

NAND vs NOR Flash

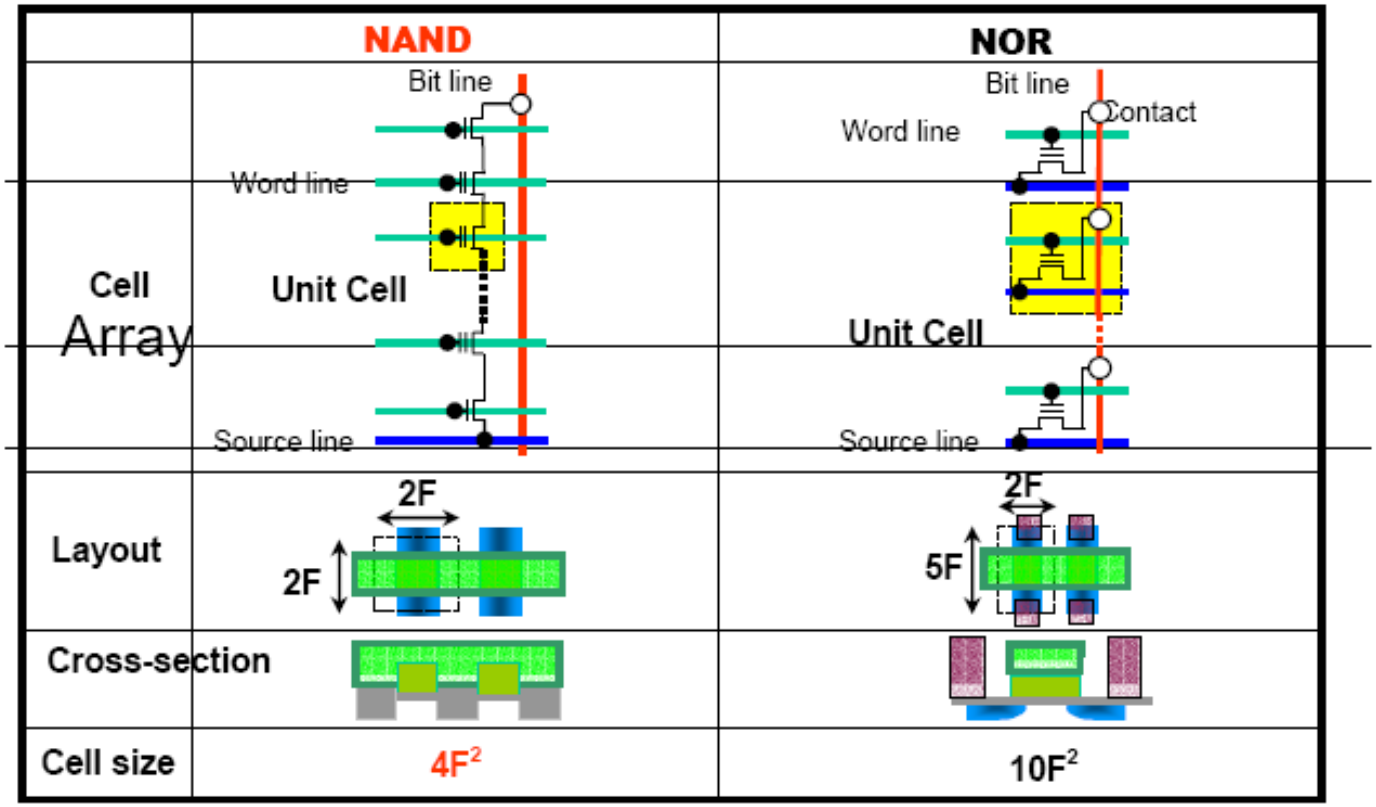


Figure 1. NAND Flash vs. NOR Flash

NAND vs NOR Flash

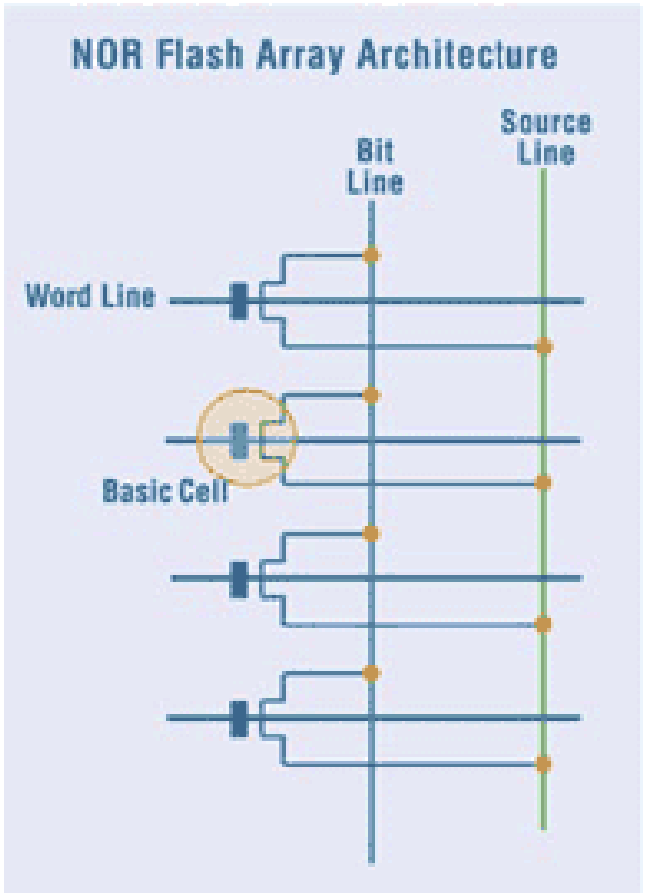
- The advantages of NAND Flash are that the erasing and programming times are short. The programming current is very small into the floating gate because NAND Flash uses Fowler-Nordheim tunneling for both erasing and programming. Therefore, the power consumption for programming does not significantly increase even as the number of memory cells being programmed is increased. As a result, many NAND Flash memory cells can be programmed simultaneously so that the programming time per byte becomes very short. Conversely, the NOR Flash can be programmed only by byte or word, and since it uses the hot electron injection mechanism for programming, it also consumes more power and the programming time per byte is longer. The programming time for NOR Flash is typically more than a order of magnitude greater than that of NAND Flash.

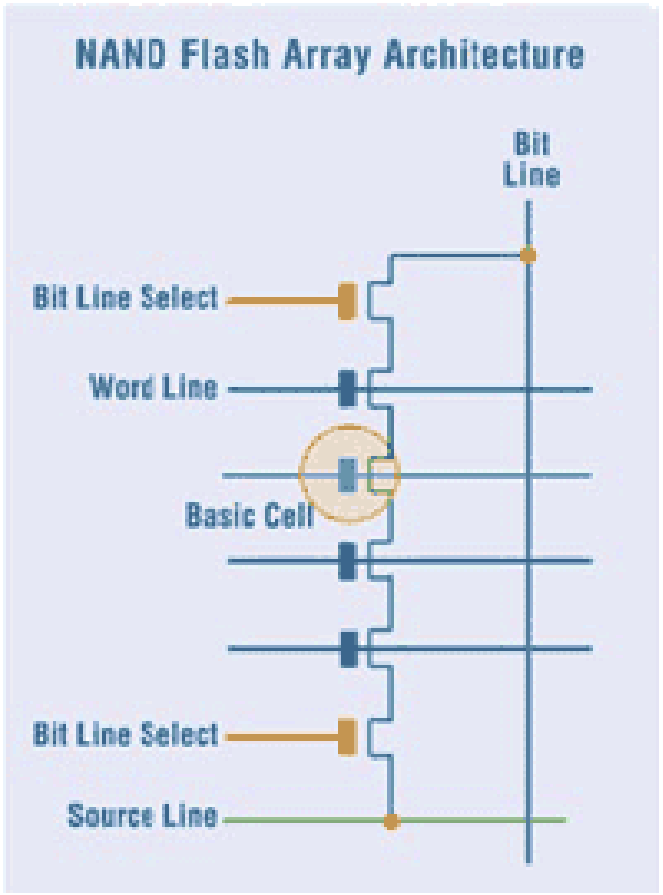
- The power consumption of NAND Flash or NOR Flash is about one tenth that of a hard disk drive. Also, the seek time for semiconductor memories is much faster than that of a hard disk. However, NAND Flash or NOR Flash must be erased before reprogramming while a hard disk requires no erasure. Therefore, in the case of continuous programming where the seek time is negligibly small, a hard disk drive can be programmed more quickly.

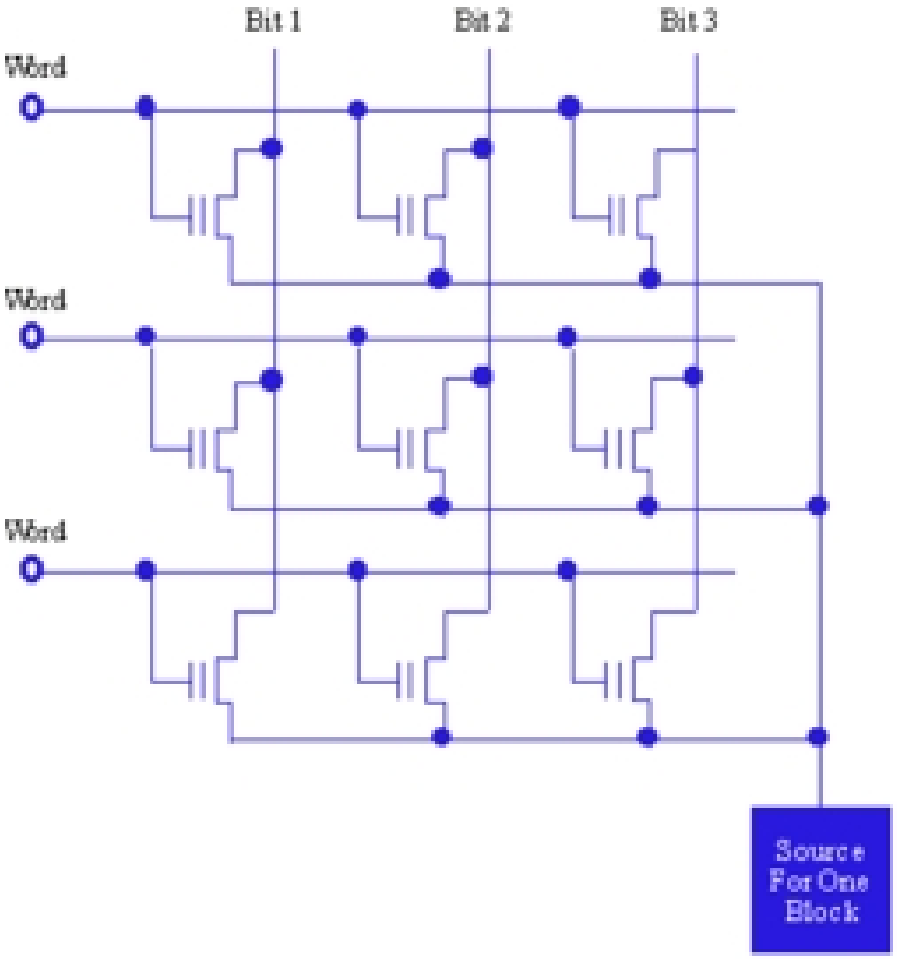
NAND vs NOR Flash

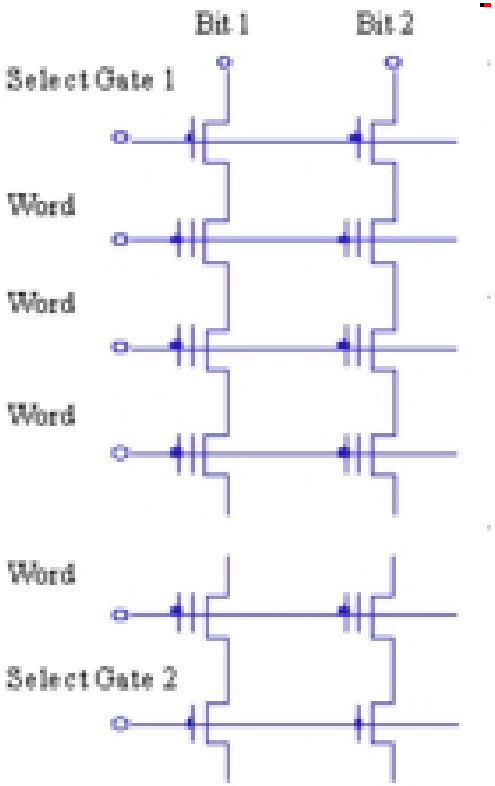
- For both for NOR Flash and NAND Flash, the endurance (which means the number of cycles a block or chip can be reprogrammed) is limited. In order to replace the UV-EPROM with Flash, and endurance of 1000 cycles was sufficient. It is estimated that at least 1,000,000 cycles are required to replace a hard disk drive. NOR Flash is typically limited to around 100,000 cycles. Since the electron flow-path due to the hot electron injection for programming is different from the one due to tunneling from the floating gate to the source for erasing, degradation is enhanced. However, in NAND Flash, both the programming and erasing is achieved by uniform Fowler-Nordheim tunneling between the floating gate and the substrate. This uniform programming and uniform erasing technology guarantees a wide cell threshold window even after 1,000,000 cycles. Therefore, NAND Flash has better characteristics with respect to program/erase endurance. In some recent scaled NOR Flash memories, their erasing scheme has been changed from source side erasing to uniform channel erasing, which is the same as the NAND Flash.

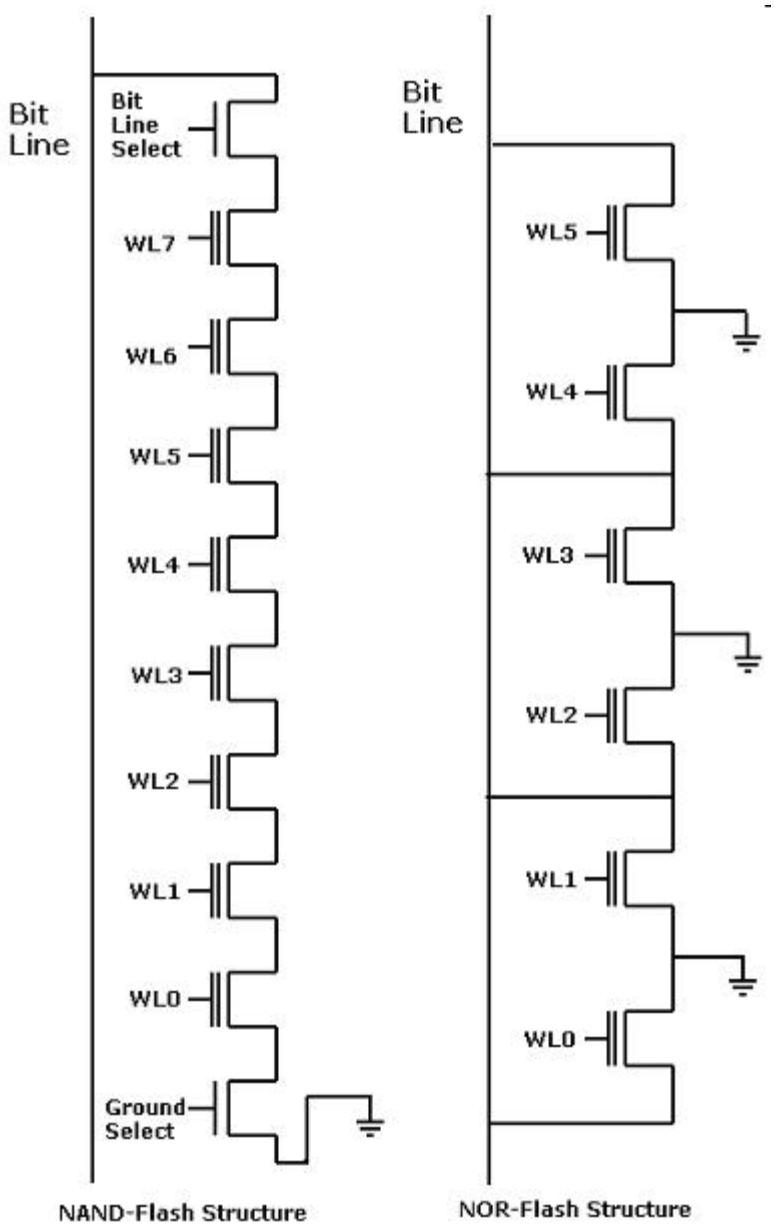
- From a practical standpoint, the biggest difference a designer will notice when comparing NAND Flash and NOR Flash is the interface. NOR Flash has a fully memory-mapped random access interface like an EPROM, with dedicated address lines and data lines. Because of this, it is easy to “boot” a system using NOR Flash. On the other hand, NAND Flash has no dedicated address lines. It is controlled using an indirect I/O-like interface and is controlled by sending commands and addresses through a 8 bit bus to an internal command and address register. For example, a typical read sequence consists of the following: writing to the command register the “read” command, writing to the address register 4 byte of address, waiting for the device to put the requested data in the output data register, and reading a page of data (typically 528 bytes) from the data register. The NAND Flash’s operation is similar to other I/O devices like the disk drive it was originally intended to replace. But because of its indirect interface, it is generally not possible to “boot” from NAND without using a dedicated state machine or controller. However, the indirect interfaces advantage is that the pinout does not change with different device densities since the address register is internal. Because NAND Flash is optimized for solid-state mass storage (low cost, high write speed, high erase speed, high endurance), it is the memory of choice for memory cards such as the SmartMediaTM, SDTM card, CompactFlashTM, and MemoryStickTM.


NOR Flash Array Architecture
Source Line
Bit Line
Word Line
Basic Cell

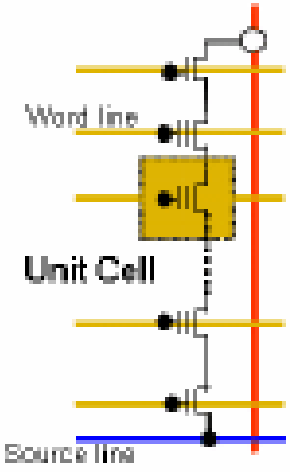
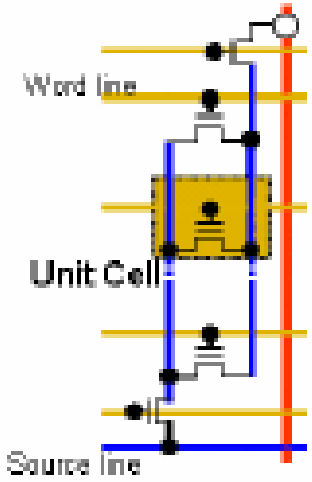
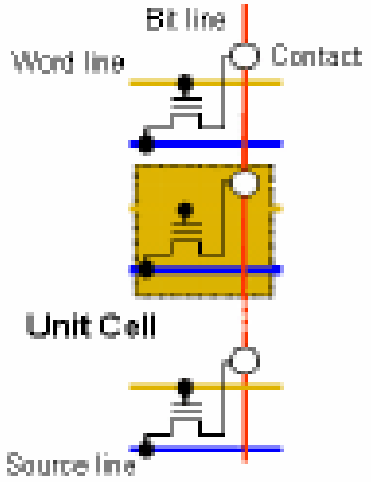
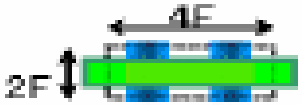
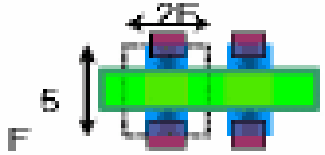
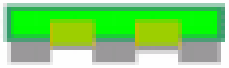
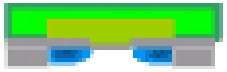

NAND Flash Array Architecture
Bit Line
Bit Line Select
Word Line
Basic Cell
Bit Line Select
Source Line


Bit 1
Bit 2
Bit 3
Word
Word
Word
Source For One Block


Bit 1
Bit 2
Select Gate 1
Word
Word
Word
Word
Select Gate 2

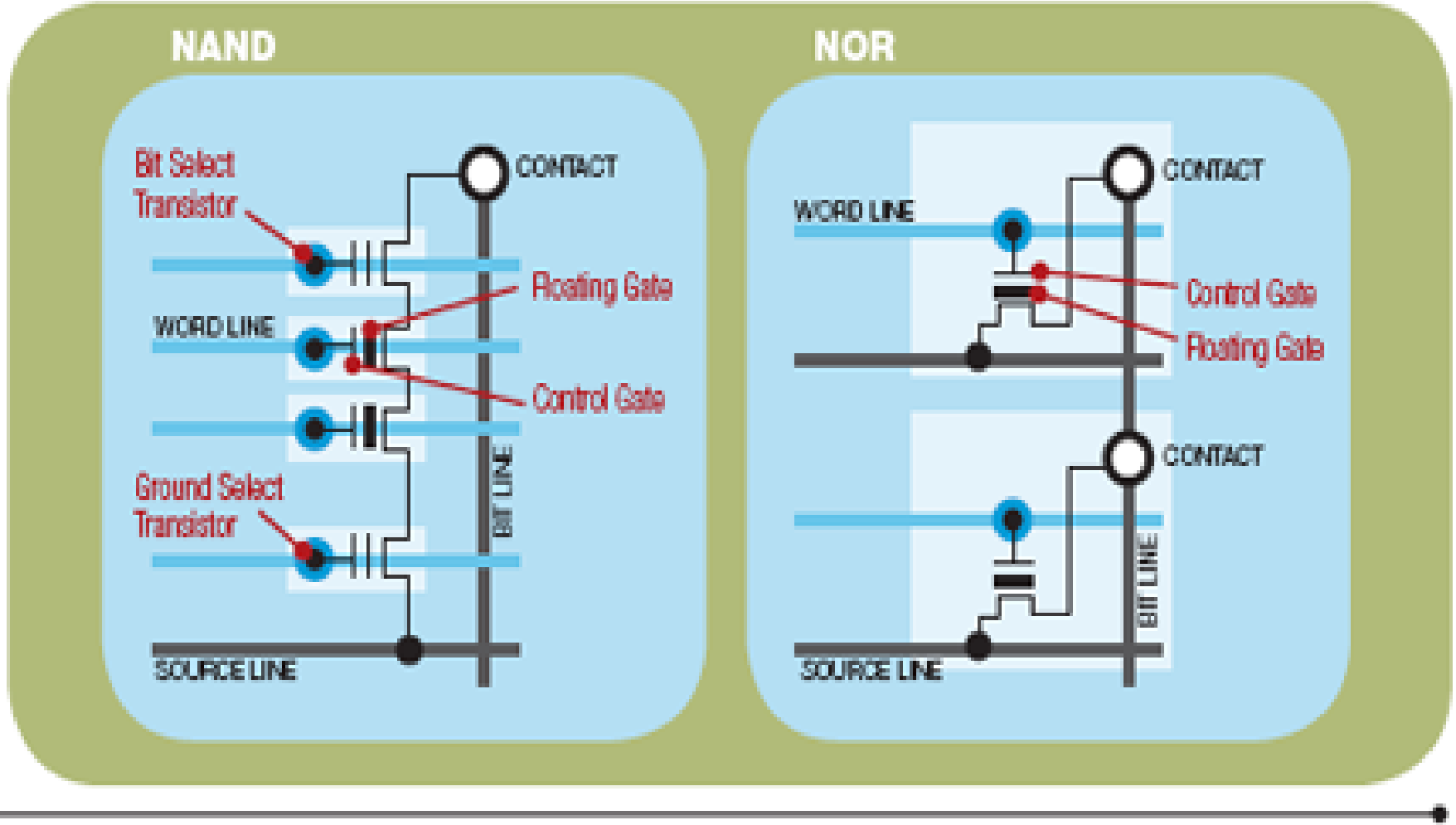

Bit Line
Bit Line Select
WL7
WL6
WL5
WL4
WL3
WL2
WL1
WL0
Ground Select
NAND-Flash Structure
Bit Line
WL5
WL4
WL3
WL2
WL1
WL0
NOR-Flash Structure

Nand and NOR flash

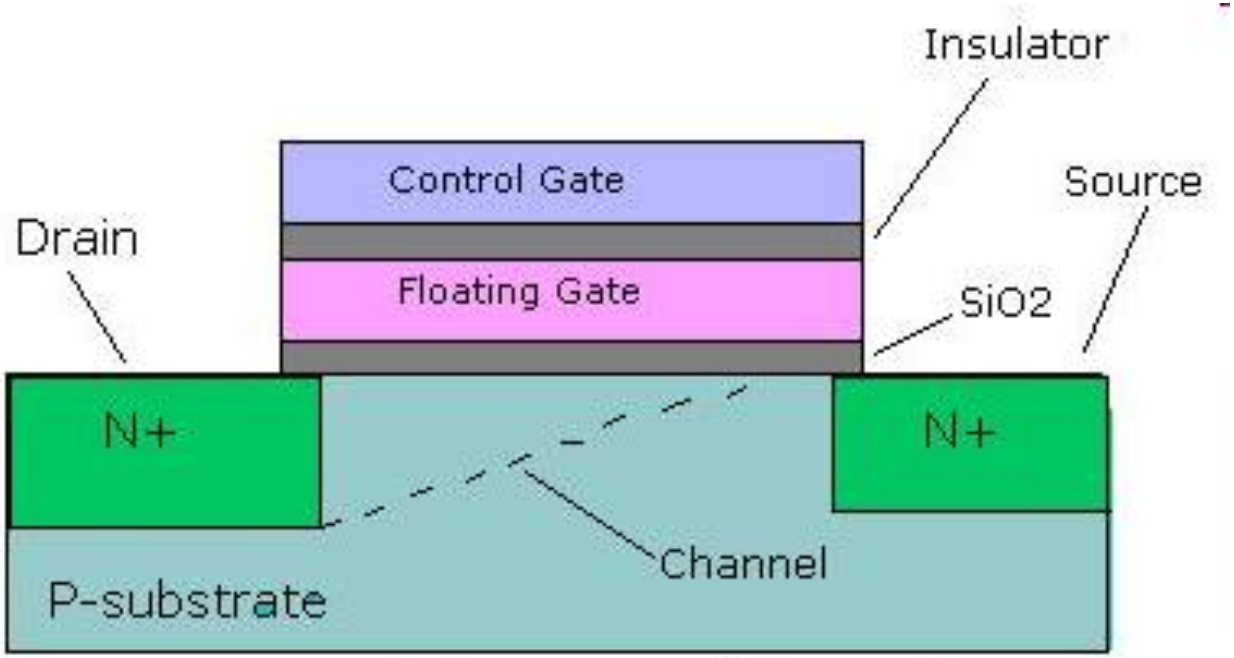
	NAND	AND	NOR
Cell Array			
Layout			
Cross-section			
Cell size	$4F^2$	$8F^2$	$10F^2$

COMPARISON

NAND vs. NOR flash memory cells

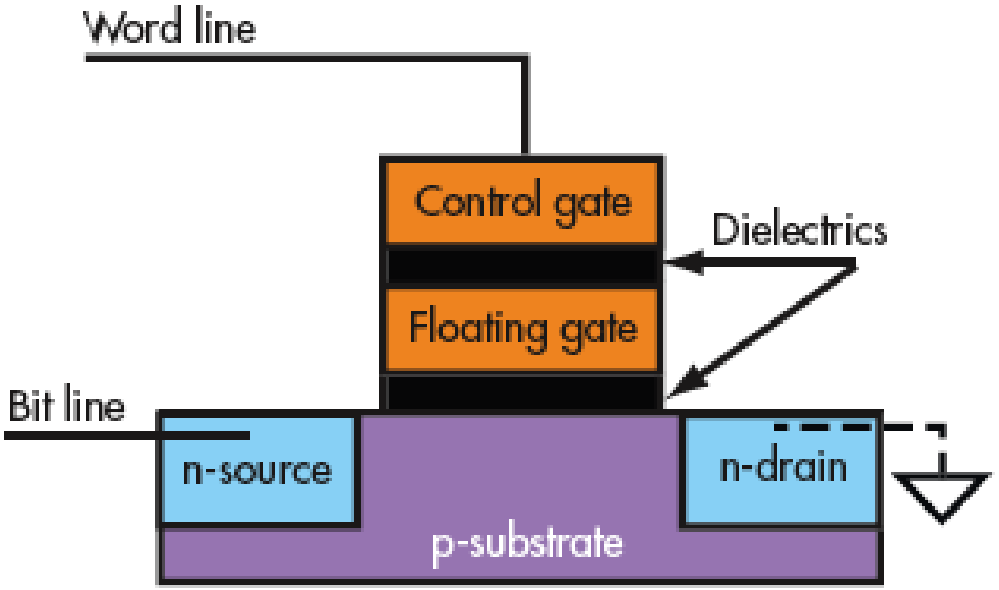


Floating Gate



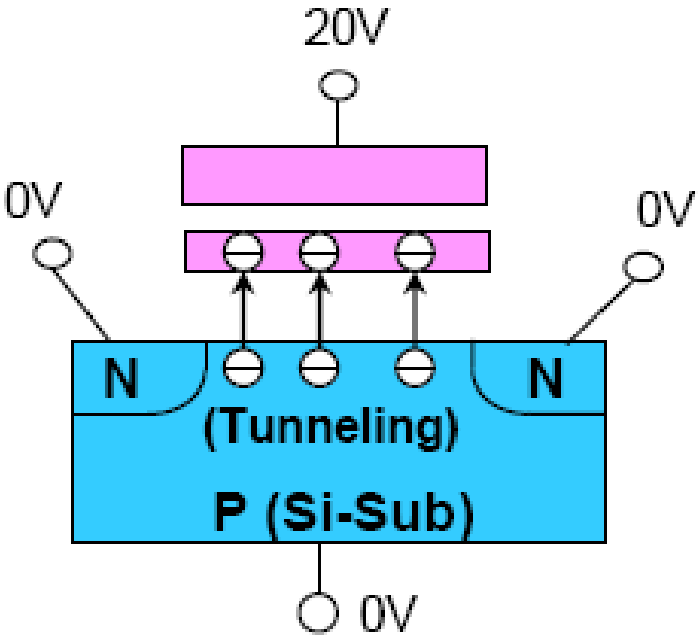
A flash memory cell (FGMOS)

Floating Gate

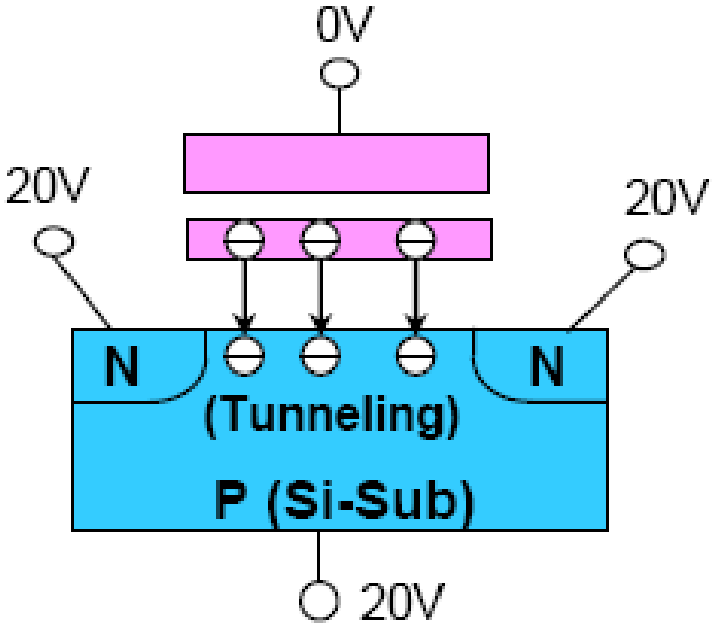


2. When a positive potential is applied to the control gate in a typical flash memory cell, negatively charged electrons are pushed through the thin oxide layer between the substrate and the floating gate and are trapped on the floating gate. This potential alters the threshold value.

NAND flash

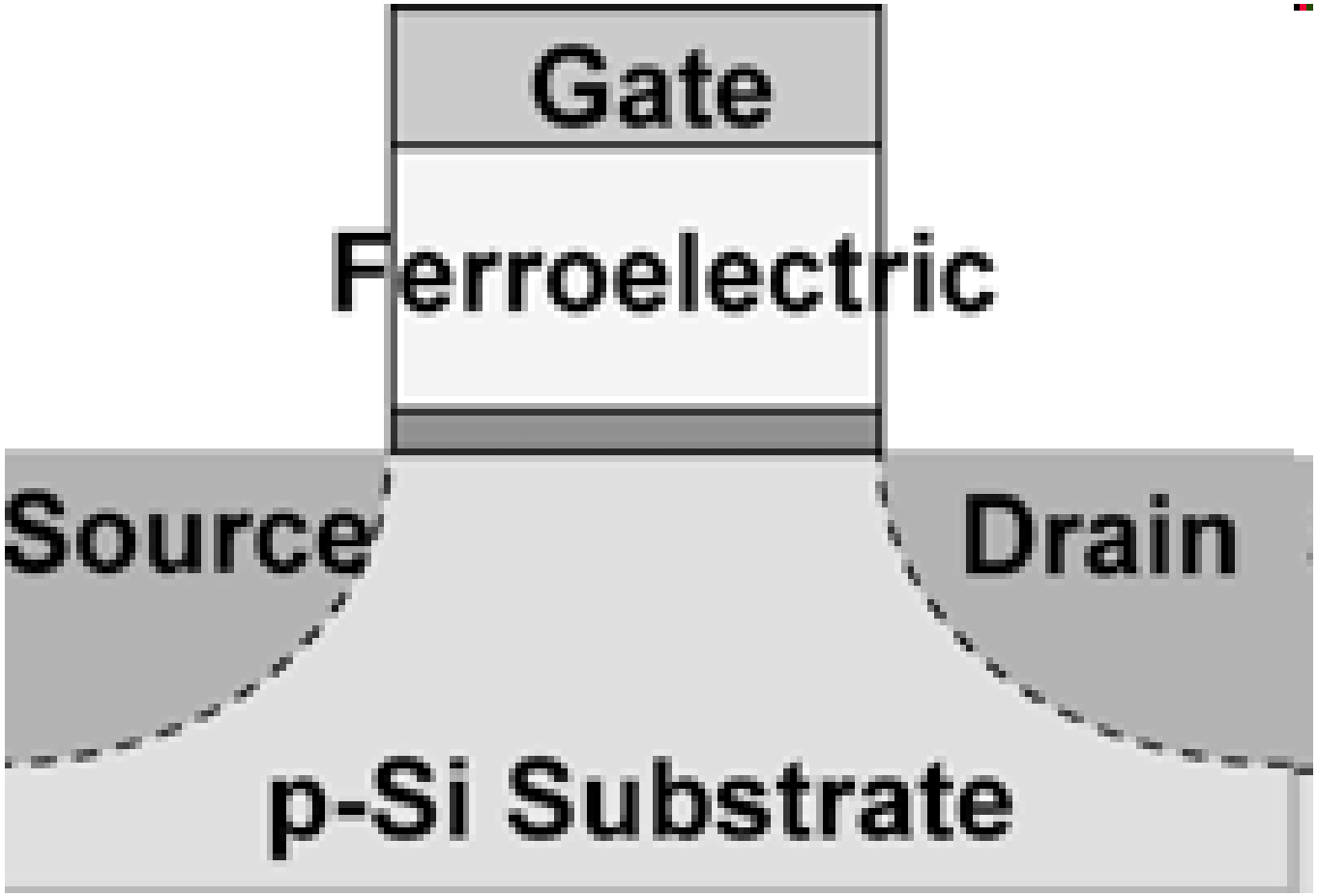


Programming

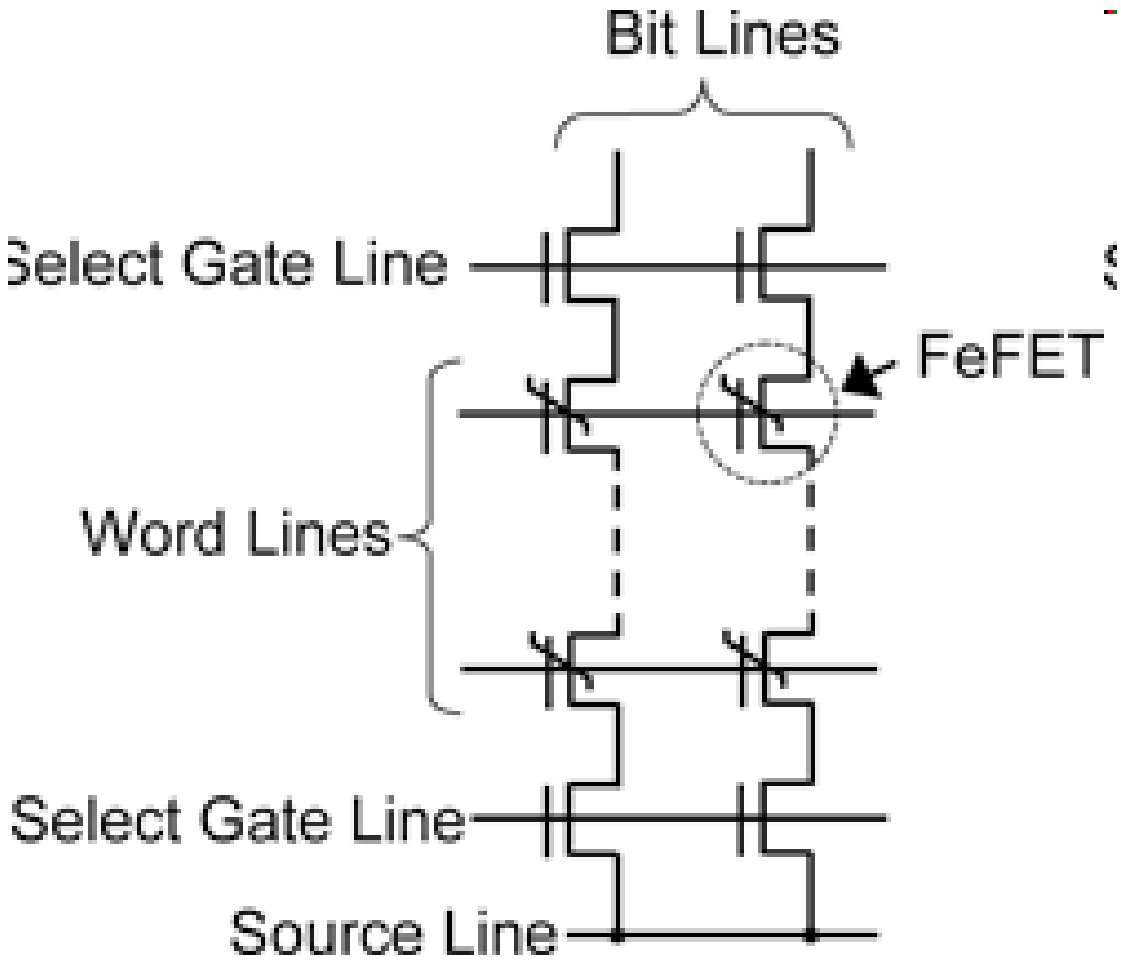


Erasing

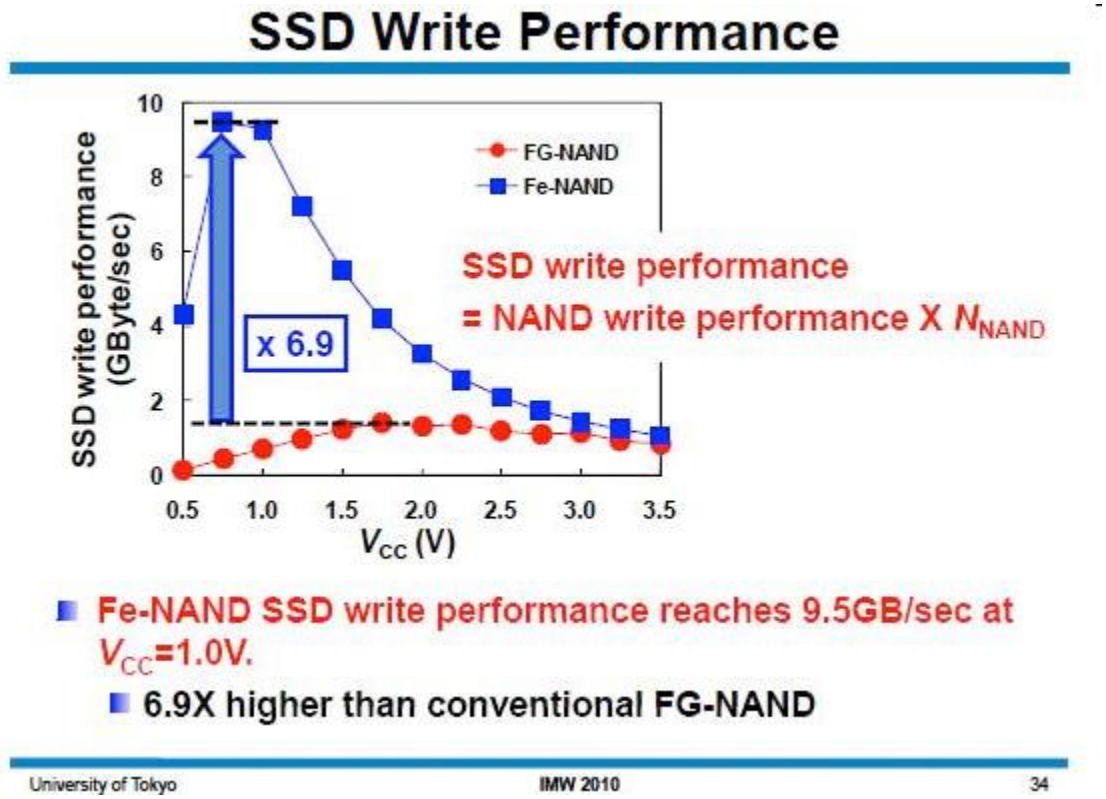
Ferroelectric Gate



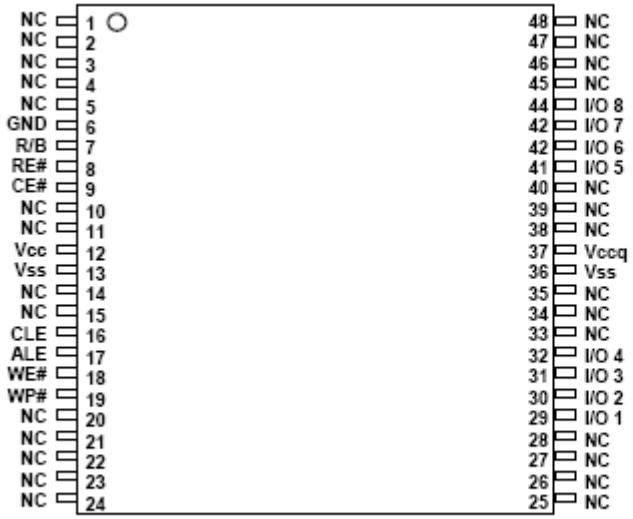
Ferroelectric Gate



SSD Write performances with Fe



The NAND Flash Interface



CLE: Command Latch Enable
ALE: Address Latch Enable
CE#: Chip Enable
WE#: Write Enable
RE#: Read Enable
WP#: Write Protect
R/B: Ready/Busy
GND: Test Input (grounded)
I/O: Input Output
Vcc: Positive Supply (core)
Vccq: Positive Supply (I/O)
Vss: Negative supply (ground)

Figure 3. NAND Flash Pinout.

ALE	CLE	Register Selected
0	0	Data register
0	1	Command register
1	0	Address register
1	1	Not defined

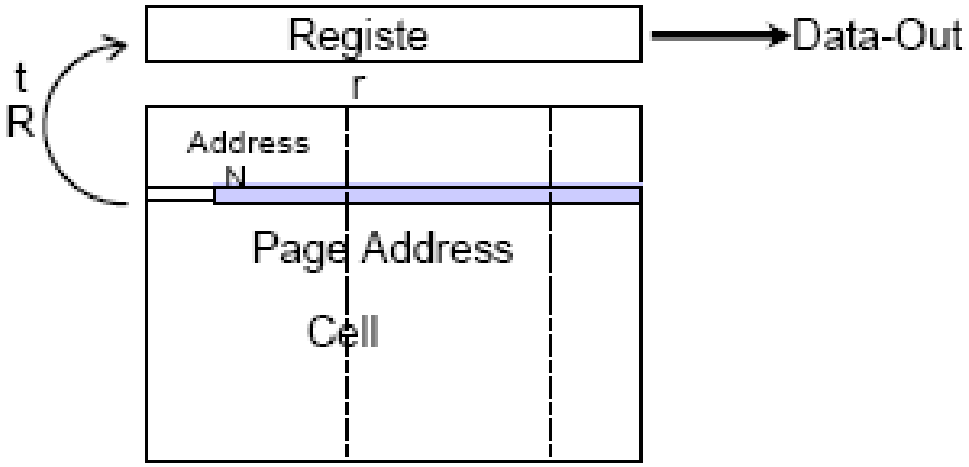
NAND flash

- The key to understanding how the NAND flash operates is the realization that in the NAND flash, the read and program operation takes place on a **page basis** (i.e. 528 bytes at a time for most NAND devices) rather than on a byte or word basis like NOR flash.
- A page is the size of the data register.
- The erase operation takes place on a block basis (for most NAND devices, the block size is 32 pages).
- **There are only 3 basic operations** in a NAND flash:
  - **read a page**
  - **program a page**
  - **erase a block**

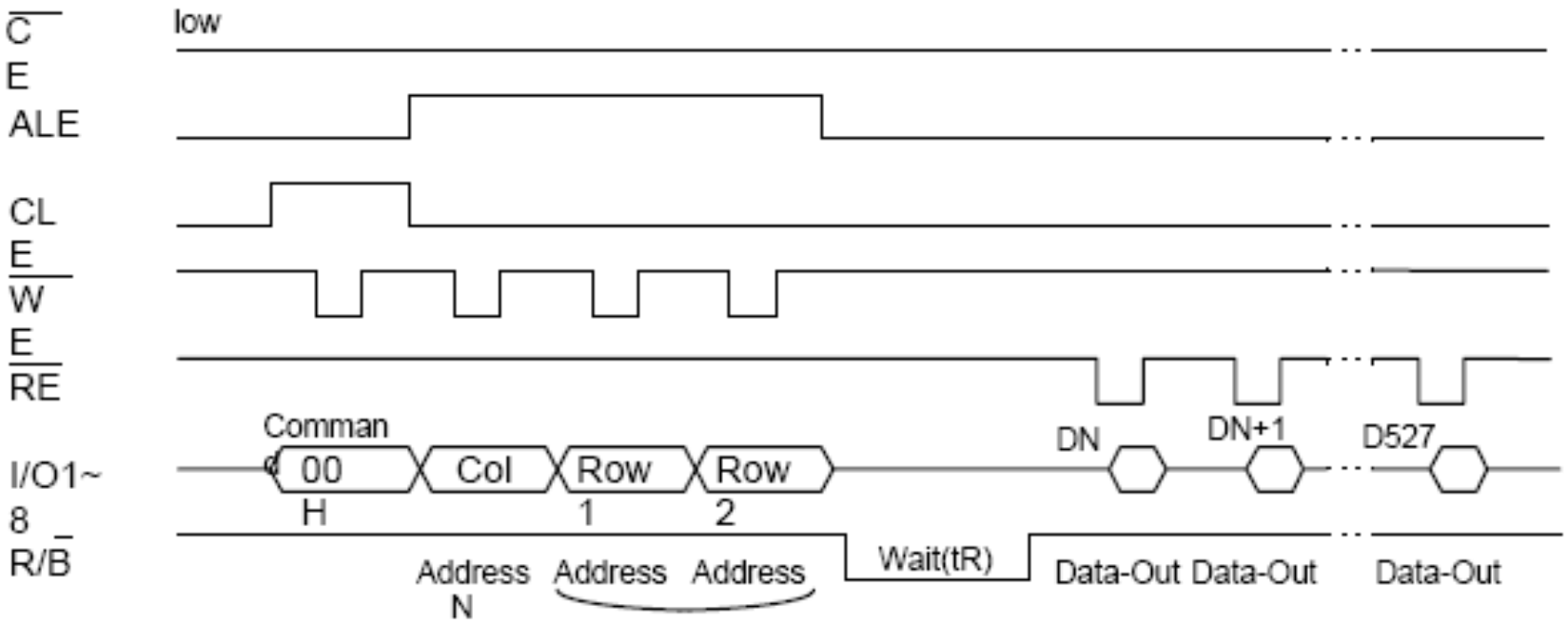
- Let's examine each of these operations in more detail.

Page read

- In a page read operation, a page of 528 bytes is transferred from memory into the data register for output. The sequence is as follows:

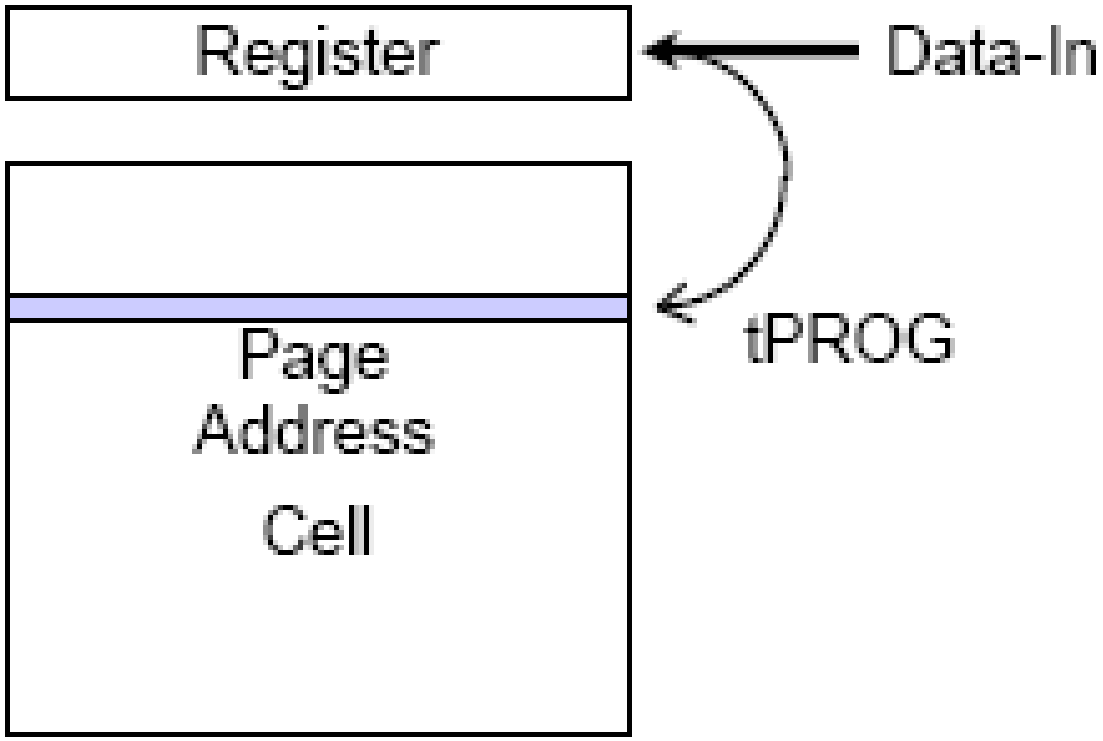


NAND flash-Page read

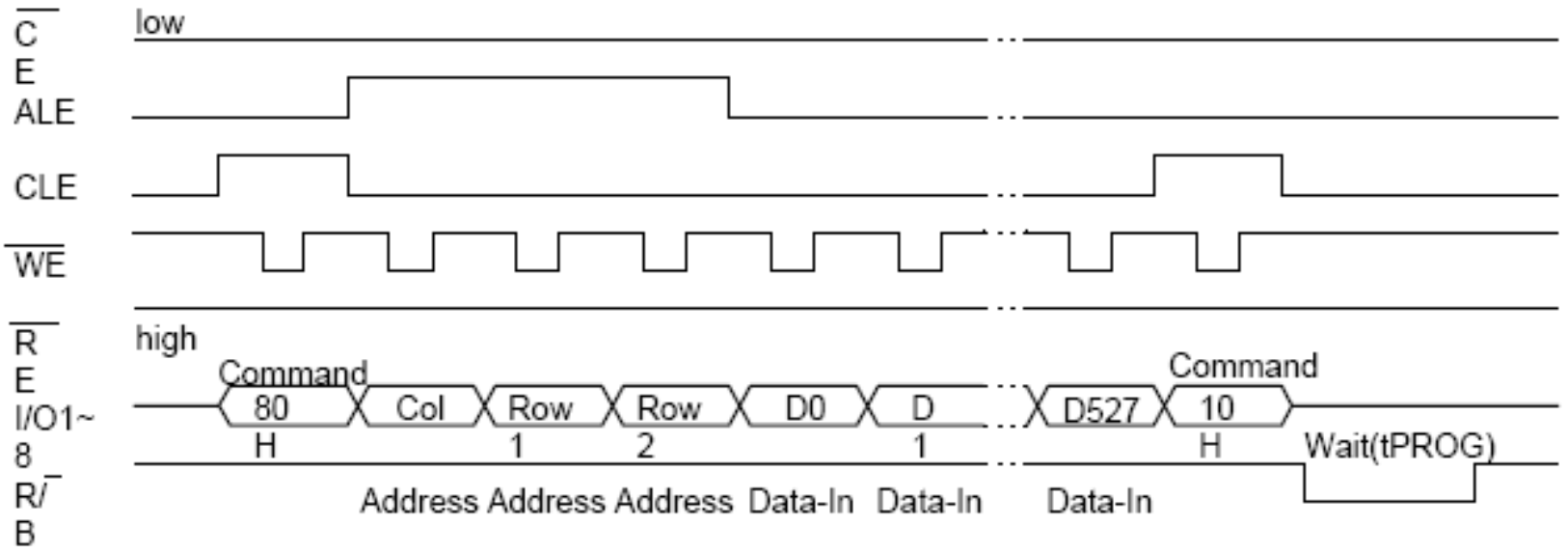


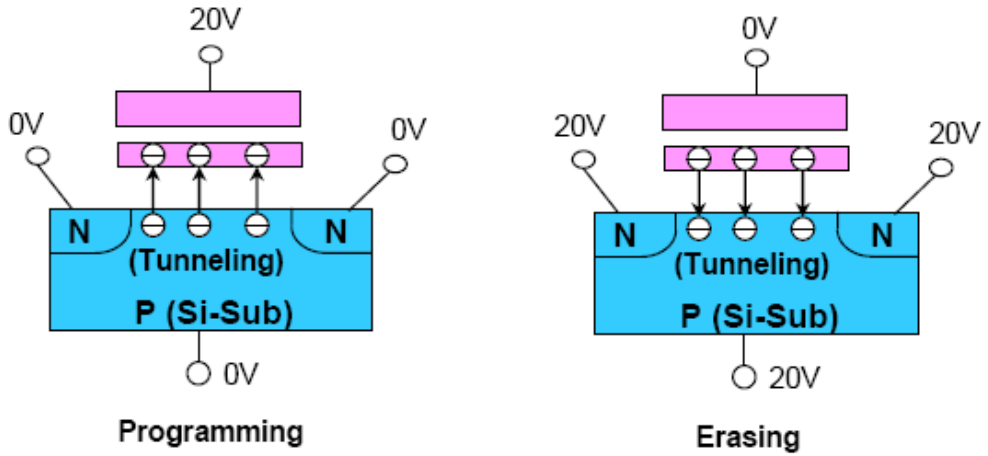
Page Program

- In a page program operation, a page of 528 bytes is written into the data register and then programmed into the memory array. The sequence is as follows:



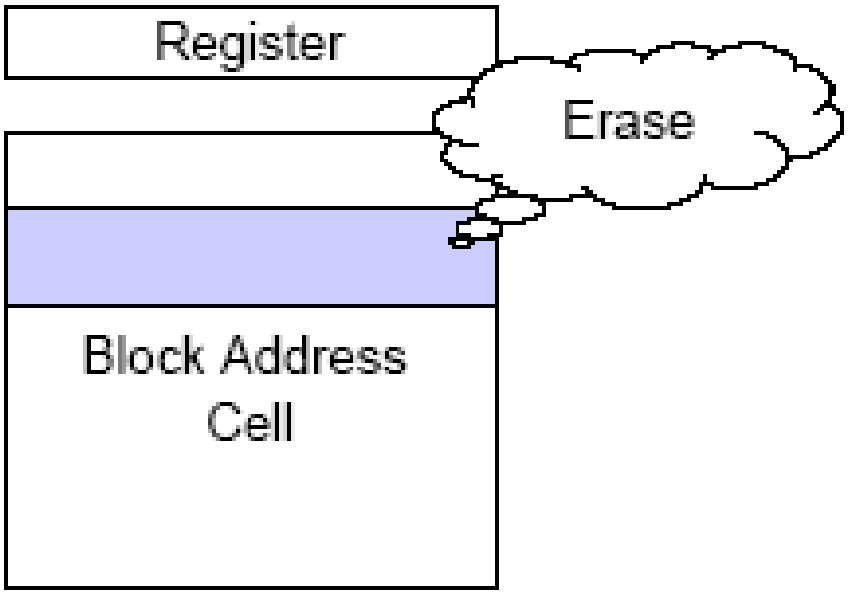
Page Program



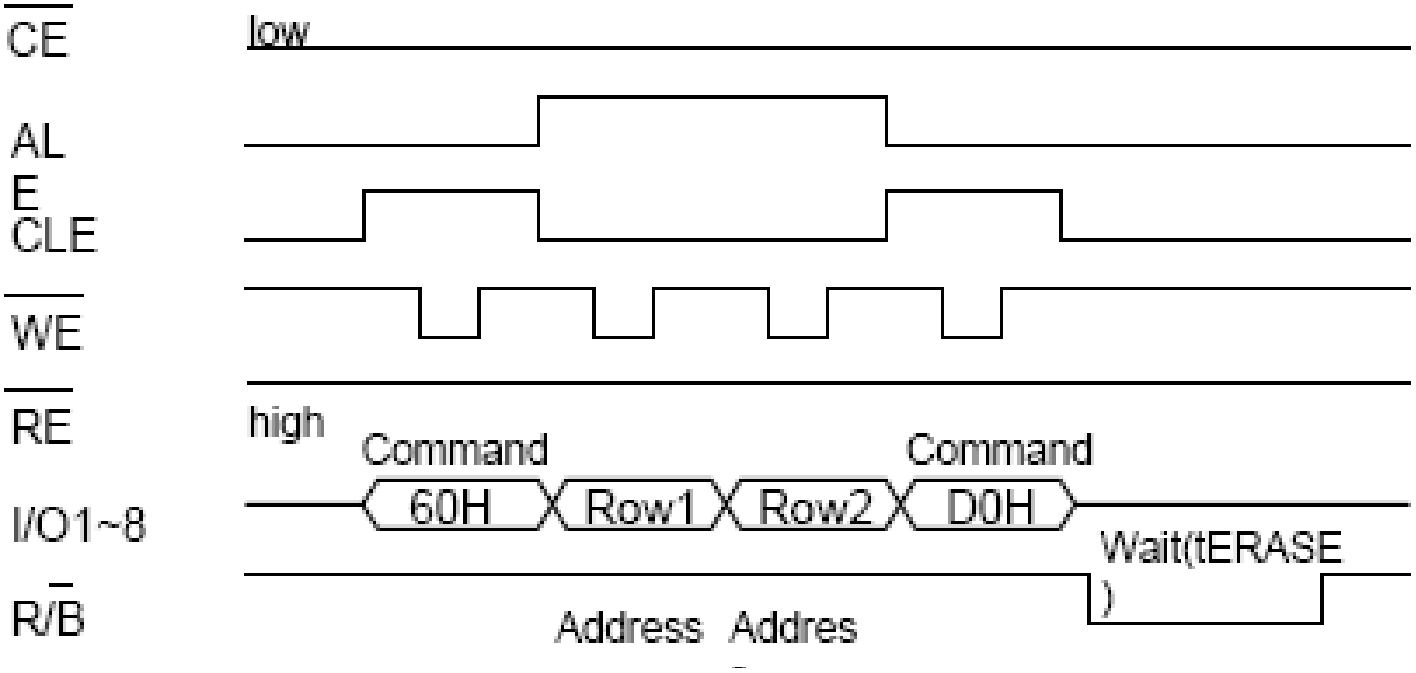


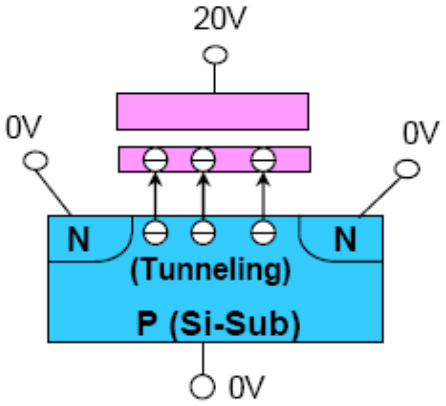
Block Erase

- In a block erase operation, a group of consecutive pages (typically 32) is erased in a single operation. While programming turns bits from “1” to “0”, block erasure is necessary to turn bits from “0” back to “1”. In a brand new device, all usable (good) blocks are in the erased state.

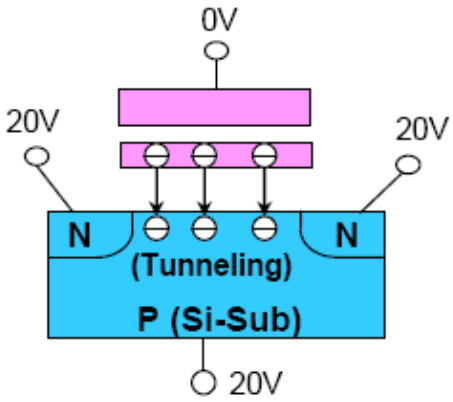


Block Erase





Programming



Erasing

Hardware Interfacing

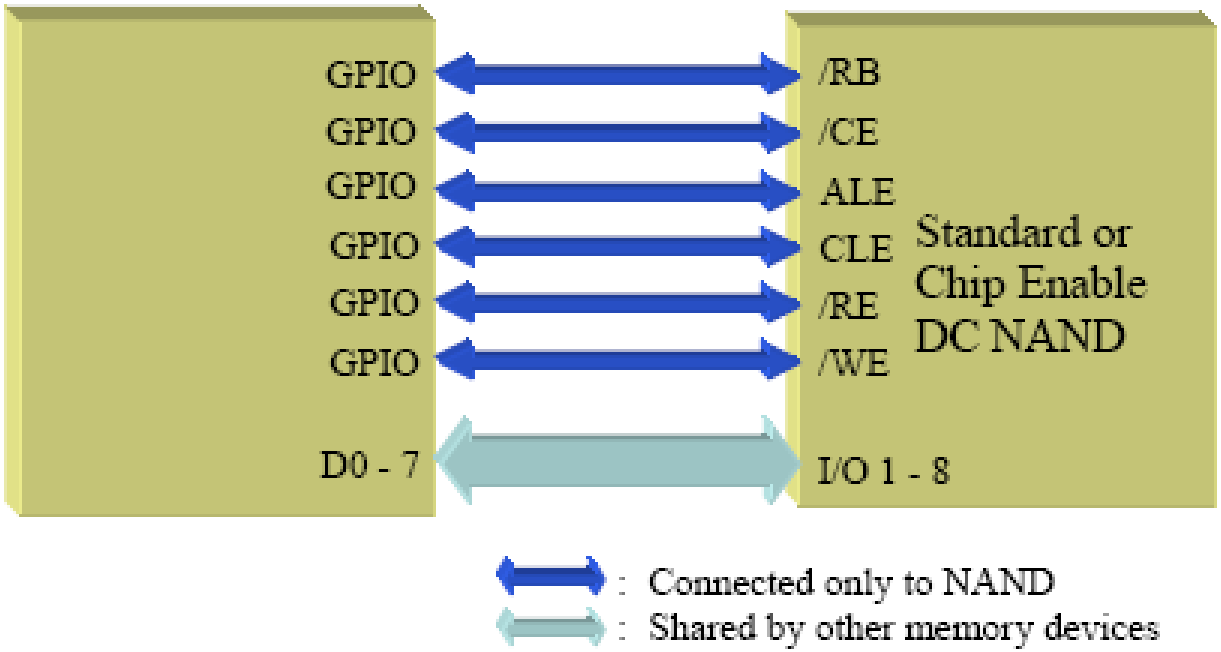


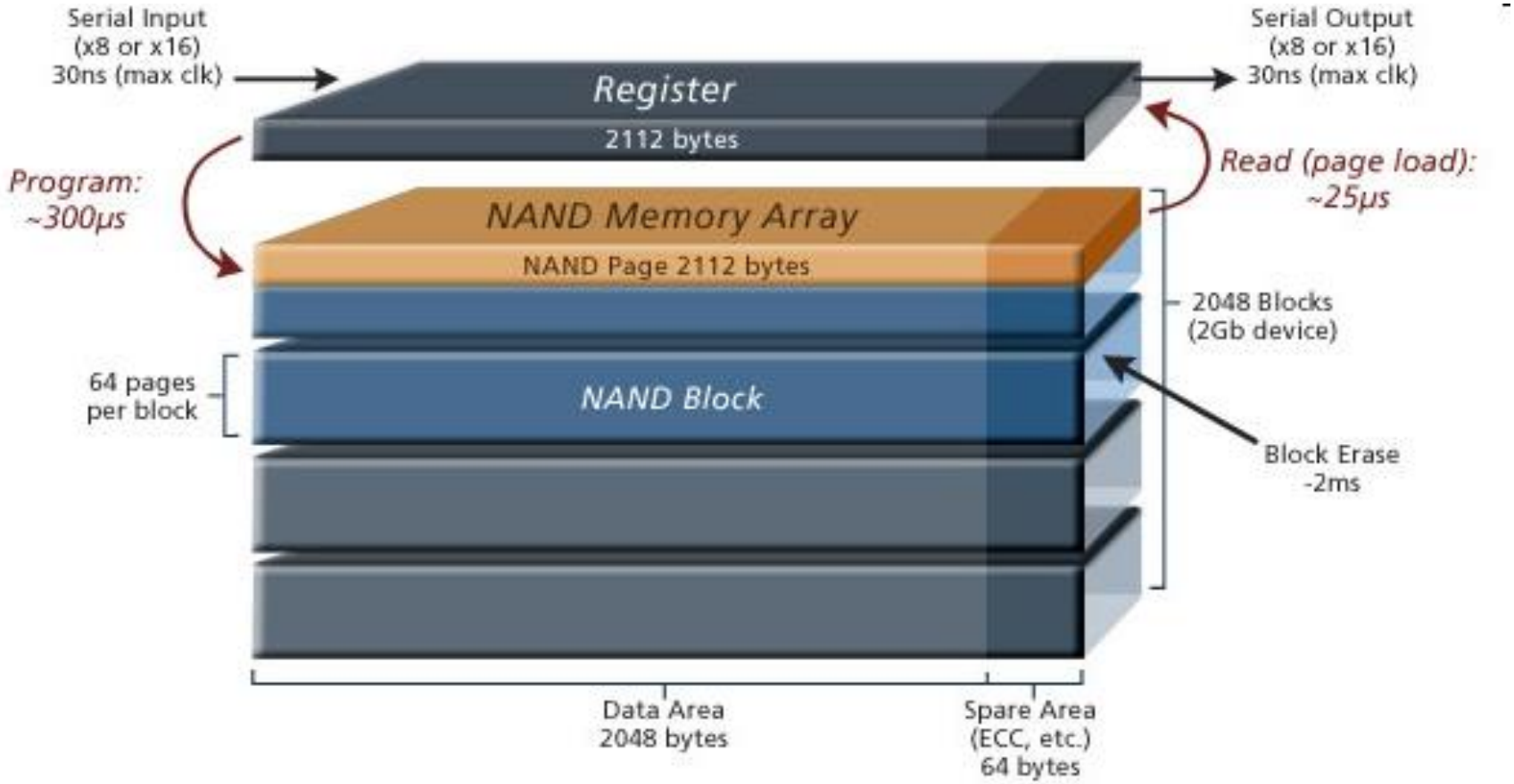
Figure 4-3. Physical Connection when GPIOs are used.

Large block vs. Small Block NAND

- In the current NAND architecture, each page consists of 528 bytes, and each block consists of 32 pages. Future NAND devices will use the large page/large block structure in which a **page will be 2112 bytes** (4 times larger) and a block will consist of **64 pages** (2 times larger) resulting in a block size that is 8 times larger. The first of these new large block NAND flash devices is the 1 Gbit TC58NVG0S3AFT00. Note that all large block devices will also have the chip enable don't care feature. The increased page and block size will enable faster program and erase speeds in future high density NAND flash.

Density	0.16 micron Small Page (528 B) Small Block (16kB)	0.13 micron Small Page (528 B) Small Block (16kB)	0.13 micron Large Page (2112 B) Large Block (128kB)
64 Mb	TC58V64BFT (standard)	N/A	N/A
128 Mb	TC58128AFT (standard) TC581282AXB (CEDC)	TC58DVM72A1FT00 (standard) TC58DVM72A1XB11 (CEDC)	N/A N/A
256 Mb	TC58256AFT (standard) TC58256AXB (CEDC)	TC58DVM82A1FT00 (standard) TC58DVM82A1XB11 (CEDC)	N/A N/A
512 Mb	TC58512FT (standard)	TC58DVM92A1FT00 (standard) TH58DVM92A1XB11 (CEDC)	N/A N/A
1 Gb	TH58100FT (standard)	TC58DVG02A1FT00 (standard)	TC58NVG0S3AFT00 (CEDC)
2 Gb	N/A	N/A	TH58NVG1S3AFT00 (CEDC)

Large blocks NAND flash



Large block vs. Small Block NAND

- The effective read speed of the large block NAND devices is similar to the small block devices:
  - **Read Time** = 6 cycles x 50ns + 25 µs + 2112 cycles x 50ns = 131 µs
  - **Read Speed** = 2112 bytes / 131 µs = 16.1 Mbytes /sec
- The effective write speed of the large block NAND devices is more than 3 times faster.
  - **Write Time** = 5 cycles x 50ns + 2112 cycles x 50ns + 1 cycle x 50ns + 200µs = 306 µs
  - **Write Speed** = 2112 bytes / 306µs = 6.9 Mbytes / sec
- The effective erase speed is nearly 8 times faster.
  - **Erase Time** = 4 cycles x 50ns + 2ms = 2ms
  - **Erase Speed** = 128kB / 2ms = 64 Mbytes / sec

Failure Modes Mechanism and Symptoms

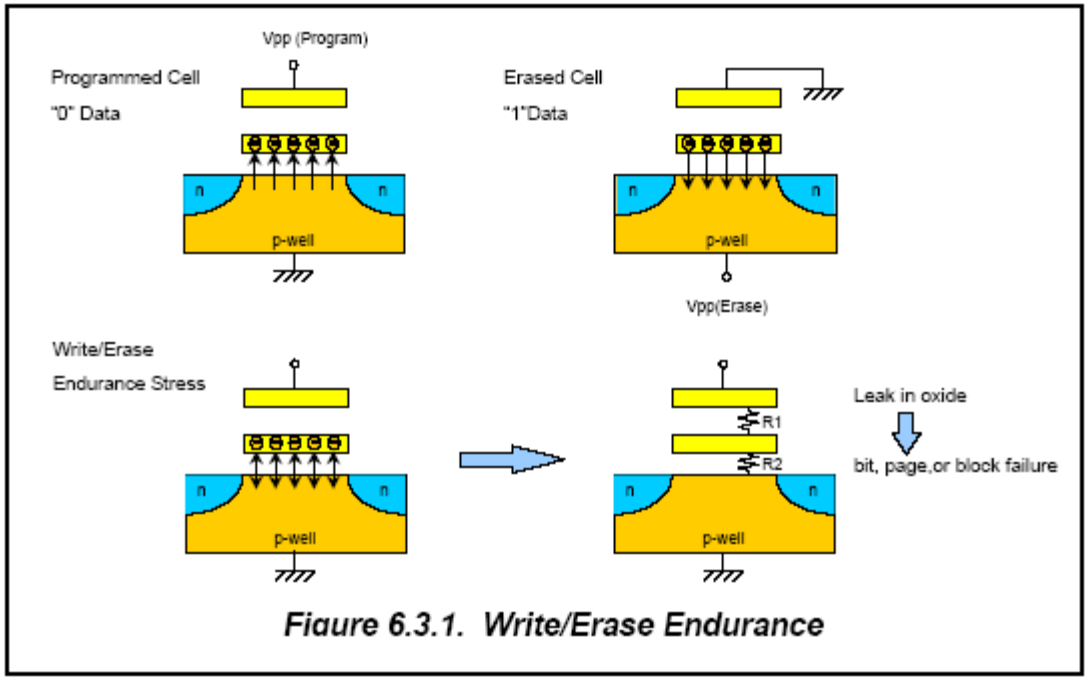


Figure 6.3.1. Write/Erase Endurance

Wear Leveling

- If flash memory had infinite write/erase endurance, wear leveling would not be necessary. However, unlike magnetic media, flash memory eventually wears out and no longer programs or erases in the allotted amount of time. Because the design of typical file systems assumed the characteristics of magnetic media, certain physical locations may be repeatedly rewritten. For example, in the DOS FAT file system, the FAT and directory areas must be modified multiple times each time a file is written or appended. When multiplied by the thousands of files in a typical file system, the FAT and directory areas of the disk will experience vastly more writes than any other area of the disk.

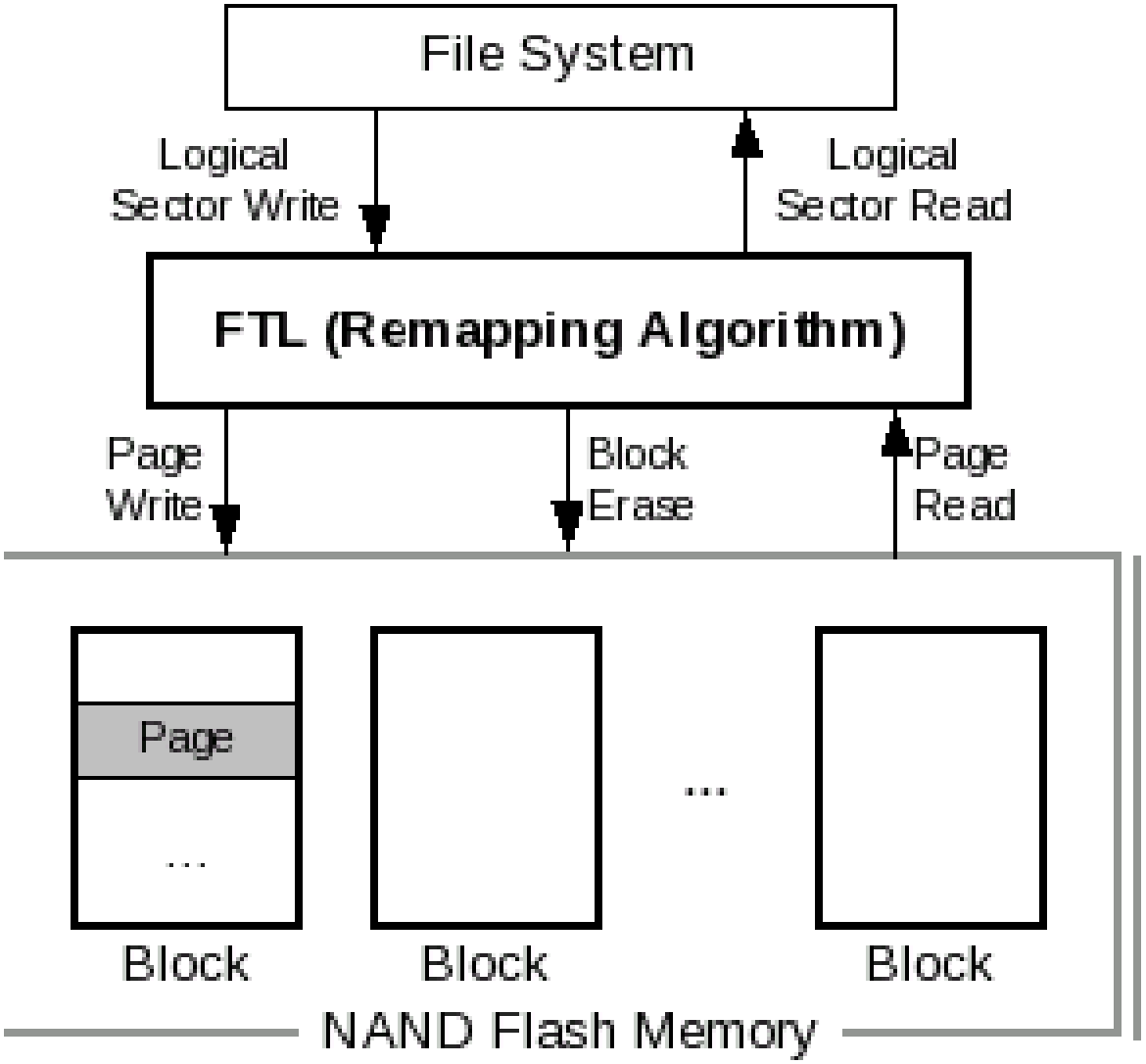
- When flash memory is used to emulate a disk drive, the physical areas of the flash that contain the FAT and directory would be worn out first, leading to early failure of the file system stored on the flash. In order to spread out the writes across as much of the flash as possible, a wear leveling algorithm is implemented by the controller (software or firmware in a hardware controller) which translates a logical address to different physical addresses for each write. Generally, this logical to physical lookup table is implemented in RAM and is initialized at power up by reading each physical block in the NAND flash to determine its logical block value.

- Ideally, wear leveling is intrinsic to the file system itself. Several new file system exist which write new data sequentially rather than overwriting a fixed location. These file systems use a technique known as journaling. For flash memory, JFFS2 (Journaling Flash File System 2) and YAFFS (Yet Another Flash File System) exist which automatically spread out wear by writing sequentially to free flash space.

Software Drivers

- Software drivers for managing NAND flash are becoming available from a variety of sources. There are open source developments such as JFFS2 and YAFFS, as well as a number of drivers available from third parties. The table below lists the sources of NAND flash driver software we are currently aware of or have discovered on the web.

Product Name	Company/Sponsor	Website
F1Pack Angel & Jet	Tokyo Electron	http://tmg-eng.teldevice.co.jp/f1pack.html
FlashFX	Datalight	http://www.datalight.com
JFFS2	Red Hat	http://sources.redhat.com/jffs2/
NAND File system	Kyoto Software Research	Contact Toshiba http://www.toshiba.com/taec/
smxFFS	Micro Digital	http://www.smxinfo.com
TargetFFS-NAND	Blunk Microsystems	http://www.blunkmicro.com/ffs
TrueFFS	Wind River Systems	http://www.windriver.com/products/true_ffs/
YAFFS	Toby Churchill	http://www.aleph1.co.uk/armlinux/projects/yaffs/


File System
Logical
Sector Write
Logical
Sector Read
FTL (Remapping Algorithm)
Page
Write
Block
Erase
Page
Read
Page
...
Block
Block
...
Block
NAND Flash Memory

Hardware Controllers

- There are a number of sources for hardware controllers for NAND flash. To date, the main application for these controllers have been for use inside flash memory cards such as CompactFlash, USB drives, or flash memory card reader/writers. Manufacturers include SST, Cypress, Standard Microsystems Corp., and many others.

What is SSDNow?

- SSD is a non-volatile, Flash-based data storage solution that offers an alternative to the traditional hard disk drive (HDD). SSDs are extremely stable and durable under rugged conditions. In contrast, HDDs have moving parts, which translate into increased probability of mechanical failures and vulnerability to excessive shock and vibration.
- Although Flash storage technology has existed and been applied in the computer industry for some time, only recently has this technology been used as a primary data storage solution.
- SSDs offer several advantages over hard disk drives in the areas of performance, reliability, power consumption and durability. From an economic standpoint, the HDD may appear to be a more cost-effective option because of its price tag. However, when assessing the total cost of ownership (TCO), some environments would prove SSD to be a more practical and advantageous data storage solution.
- *What is SSDNow?*
- Kingston's SSD*Now* is a high-performance SATA2 HDD replacement. In addition, unlike some DRAM-based SSD solutions, SSD*Now* relies on non-volatile NAND Flash memory chips to store data, making it highly reliable.

The Advantages of SSDNow

- With non-volatile NAND doing the work of moving parts, SSD*Now* has:
- • **Extreme Data Transfer Rates** — With its sustained read and write speeds, SSD*Now* is an impressive solution.
- • **Low Access Times** — SSD*Now* has access times that offer extremely fast application loads and speedy boot times.
- **Reduced Power Consumption** — With low active and idle power usage, SSD*Now* is excellent for an eco-friendly solution and/or for data center storage applications, especially where multiple drive configurations are required.
- • **Exceptional Durability / Reliability**
  - o **Vibration and Shock** — SSD*Now* can handle strenuous environments and/or accidents, such as the accidental dropping of a notebook computer.
  - o **Error Correction** — ECC is a common feature of HDD to protect against bit failures that lead to data corruption. A very similar technology is also used in Kingston's SSD*Now* to ensure data reliability.
  - o **Write Amplification** — Referring to the amount of data actually written to a drive for a given write request, SSD*Now*'s write-amplification factor is extremely efficient.
  - o **Wear Leveling Algorithm** — SSD*Now*'s wear-leveling efficiency prevents the overuse of cells by effectively distributing writes to all available cells without wasting write-erase cycles.
  - o **Capacity Provisions** — Additional capacity is included to provide spare blocks to replace any that become unusable and room for wear-leveling algorithms tooperate. While this spare area enhances the reliability of NAND Flash, a 128GB SSD may actually have 156GB, for example, of integrated NAND memory to provide headroom. SSD controllers manage these advanced features and help to make SSD*Now* suitable for high-capacity storage with the necessary three to five years of expected device life. Depending on the application for which the SSD is intended, the life expectancy could be extended further.
- **SSD*Now*'s cycling formula for write-amplification and wear-leveling efficiency:**
- *Cycles = (Host writes) * (Write amplification factor) * (Wear leveling factor) / (Drive capacity)*
- • **Managing Endurance** — Wear leveling and write amplification are factors to consider when calculating the life of any NAND Flash product. When management of these factors is done poorly, the drive will wear out quicker, but when done well (like that of SSD*Now)*, the aforementioned strategies and advanced techniques help overcome endurance limitations, therefore allowing a longer life.

The Disadvantages of HDD

- In a standard computer environment, HDDs are the main storage solution for all system and application software, as well as personal data (e.g., files, folders, pictures, etc .). For over 50 years, this has been accomplished with HDDs, which are made of spinning disks read by heads connected to moving arms (Figure 1). Unfortunately, these moving parts make the HDD susceptible to common HDD setbacks, including but not limited to:
  - • **HDD Failure** — Head crash, spindle motor failure and shock and vibration, to name a few.
  - • **High Operating Temperatures** — Reaching up to 60°C at times.
  - • **Increased Power Consumption** — Up to 5X more than that of SSD*Now*.
  - • **Increased Seek / Latency Time** — Due to the process in which the HDD needs to first locate files stored on platters after which spinning into action.
- SSD*Now* omits the moving parts, resulting in reduced failures, decreased temperatures and lower power consumption, while increasing performance.

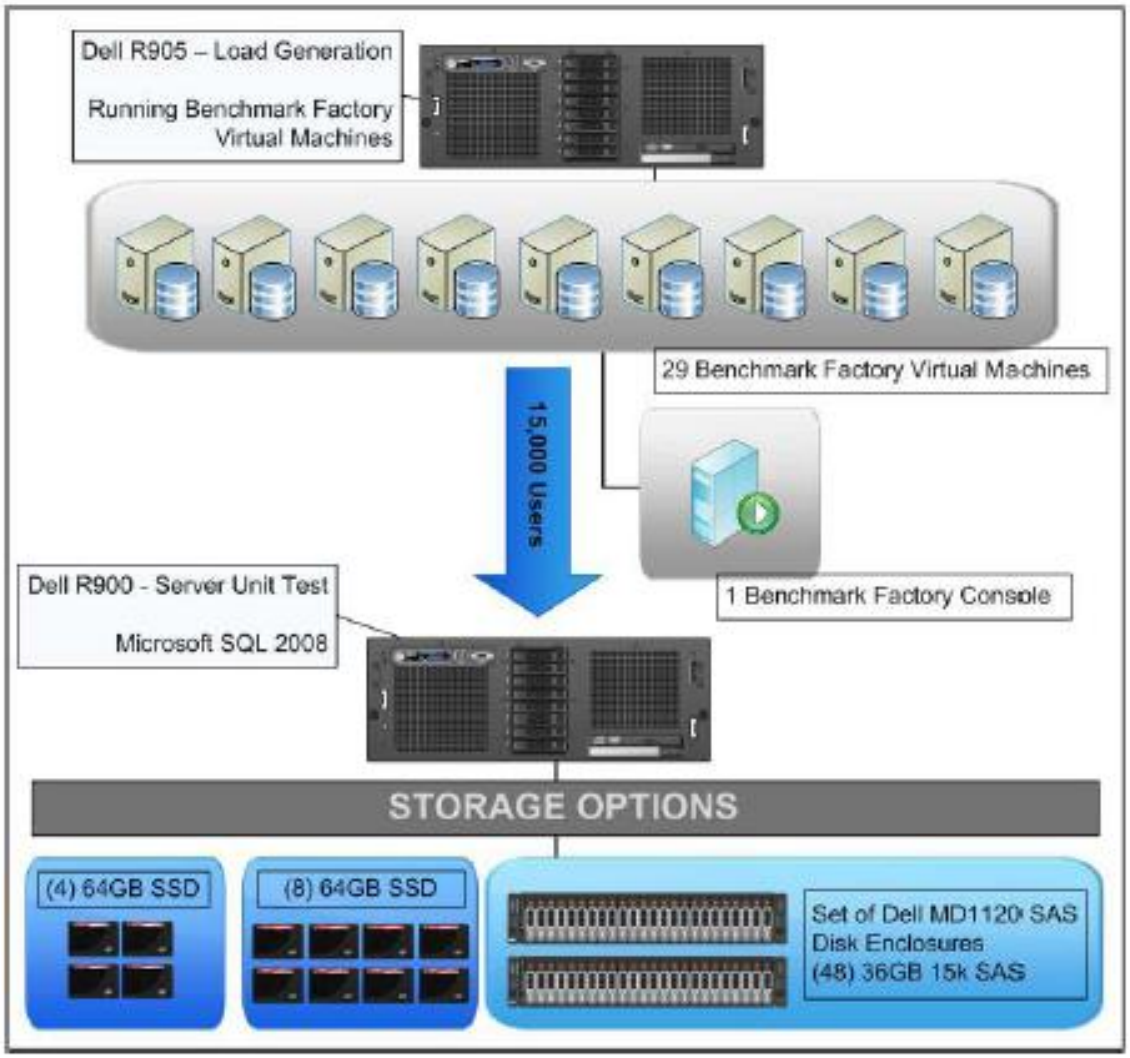
How SSD and HDD Stack Up

Feature	*SSDNow*	HDD
Reliability	No moving parts less chance for mechanical failures	Subject to mechanical failures
Performance	Near zero latency on seek means faster response times and better performance	Mechanical operation adds to seek times and limits response time
Durability	No moving parts makes SSD much more resistant to shock and vibration	Excessive shock and or vibration can cause mechanical failure
Power	Requires less power compared to HDD	More power required and higher performance HDDs will consume more power than consumer grade HDDs

Figure 2: SSD*Now* and HDD Stack Up

Effects of Solid-State Drives on Microsoft SQL Server 2008

Overview of testing infrastructure:



See Appendix A for details of the server infrastructure.

Effects of Solid-State Drives on Microsoft SQL Server 2008

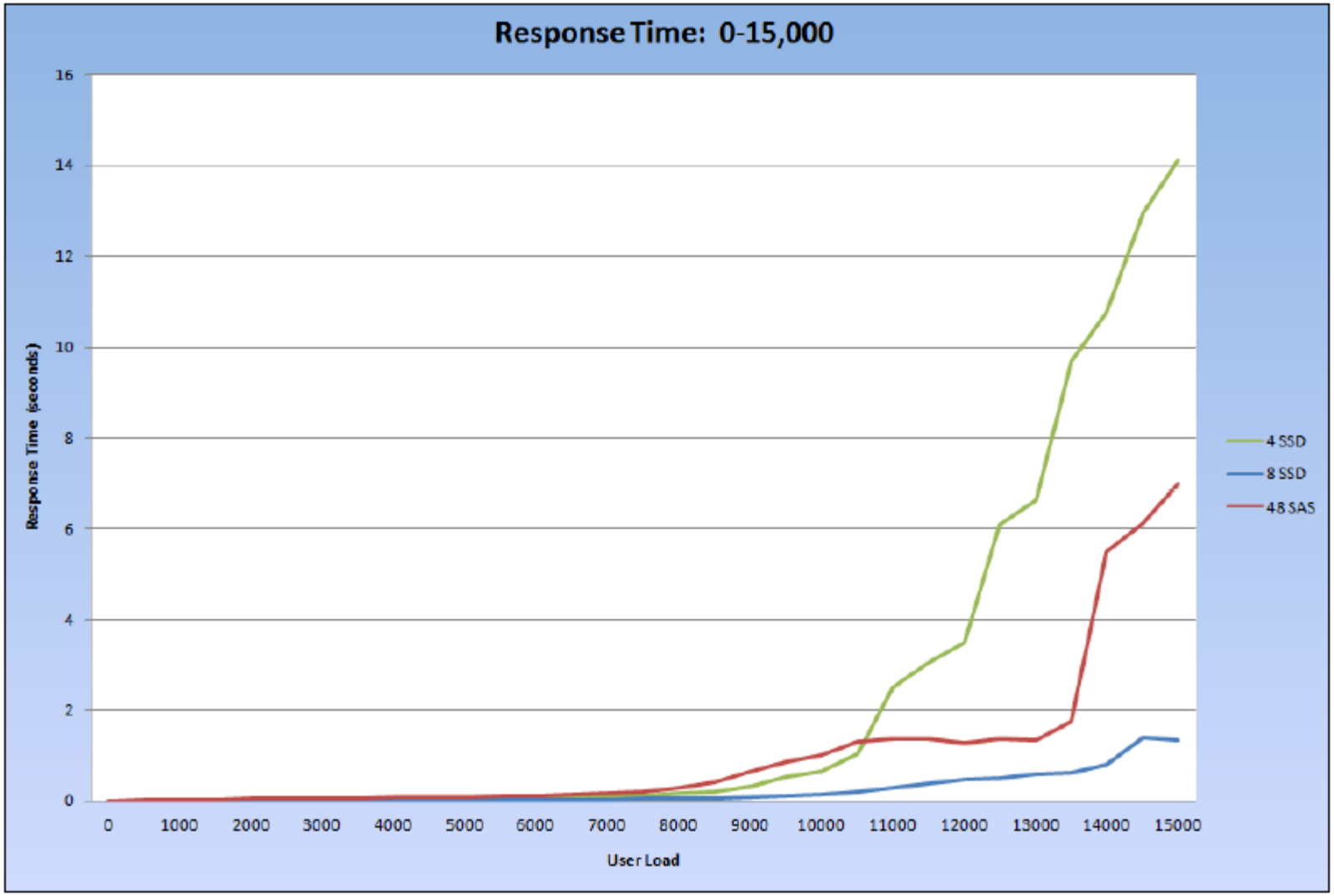
Table 1: Primary test storage options.

	48 Seagate 15k SAS	8 Kingston SNE125-S2/32GB	4 Kingston SNE125-S2/64GB
Drive Type/Connection	SAS/SAS, 15k	SSD/SATA	SSD/SATA
Storage Controller	Dell PERC 6/E	Dell PERC 6/E	Dell PERC 6/i
Controller Firmware	6.1.1-0047	6.1.1-0047	6.1.1-0047
Controller Driver	2.23.00.64	2.23.00.64	2.23.00.64
Firmware	D506	045C8621	045C8790
Total Drives in Test	48	8	4
RAID Level	RAID10	RAID 10	RAID 10
Raw Capacity per Drive	36GB	32GB	64GB
Formatted Volume	116GB	116GB	116GB

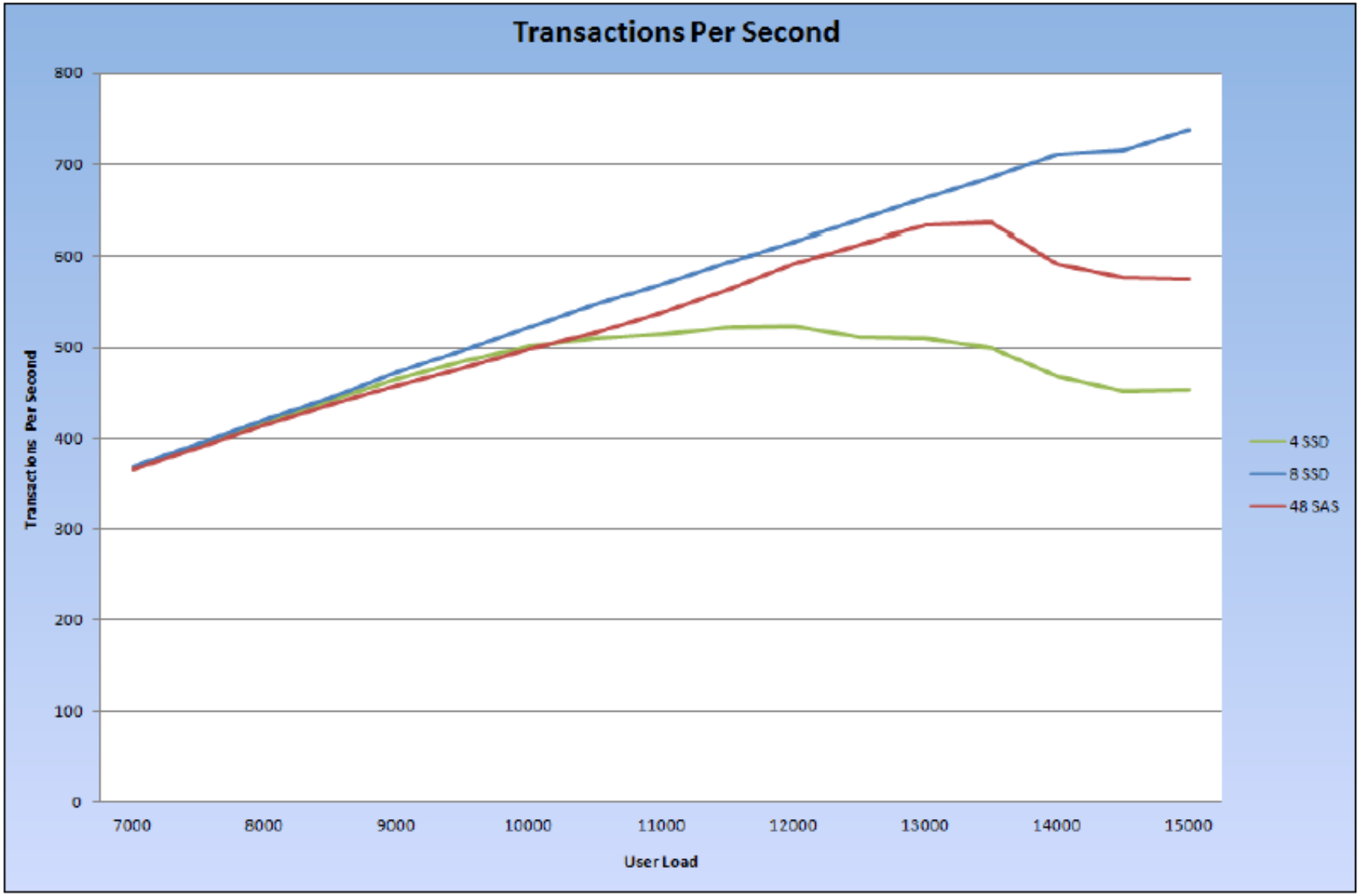
Effects of Solid-State Drives on Microsoft SQL Server 2008

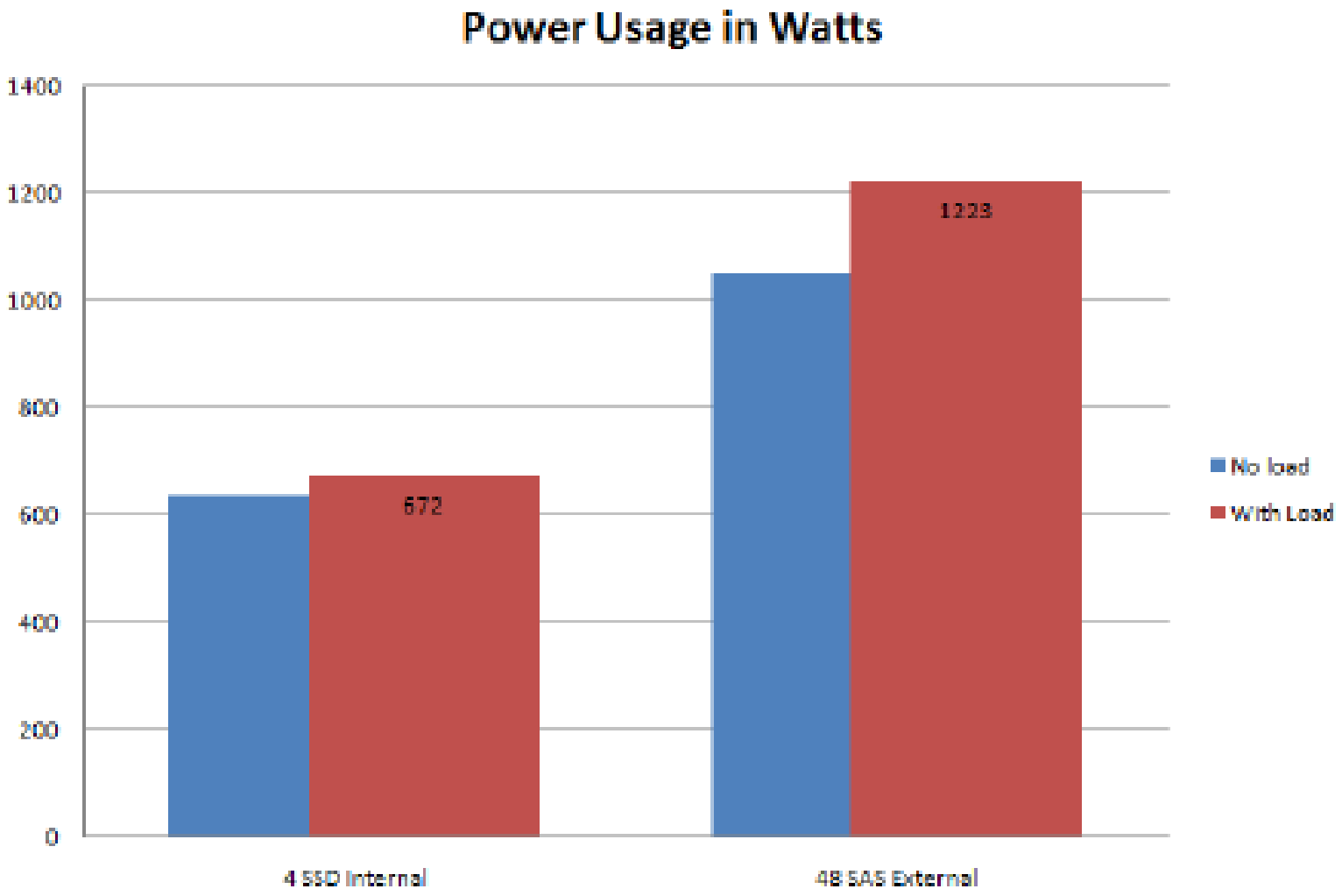
- Key Findings
- There were three main scenarios studied for the database storage subsystem:
- 1. 48 SAS drives configured as RAID10 external with a PERC 6/E RAID Controller.
- 2. 8 Solid-State drives configured as RAID10 external with a PERC 6/E RAID Controller.
- 3. 4 Solid-State drives configured as RAID10 internally with a PERC 6/I RAID Controller.

Effects of Solid-State Drives on Microsoft SQL Server 2008



Effects of Solid-State Drives on Microsoft SQL Server 2008




Power Usage in Watts
1400
1200
1000
800
600
400
200
0
672
1223
No load
With Load
4 SSD Internal
48 SAS External

Effects of Solid-State Drives on Microsoft SQL Server 2008

Advantages

- Faster start-up because no spin-up is required.
- Fast random access because there is no read/write head[14]
  - Low read latency times for RAM drives.[15] In applications where hard disk seeks are the limiting factor, this results in faster boot and application launch times (see Amdahl's law).[16]
  - Consistent read performance because physical location of data is irrelevant for SSDs.[17]
  - File fragmentation has negligible effect.
- Silent operation due to the lack of moving parts.
- Low capacity flash SSDs have a low power consumption and generate little heat when in use.
- High mechanical reliability, as the lack of moving parts almost eliminates the risk of "mechanical" failure.
- Ability to endure extreme shock, high altitude, vibration and extremes of temperature.[18][19] This makes SSDs useful for laptops, mobile computers, and devices that operate in extreme conditions (flash).[16]
- For low-capacity SSDs, lower weight and size: although size and weight per unit storage are still better for traditional hard drives, and microdrives allow up to 20 GB storage in a CompactFlash form-factor. As of 2008 SSDs up to 256 GB are lighter than hard drives of the same capacity.[18]
- Flash SSDs have twice the data density of HDDs (so far, with very recent and major developments of improving SSD densities), even up to 1TB disks[20][21] (currently more than 2TB is atypical even for HDDs)[22]). One example of this advantage is that portable devices such as a smartphone may hold as much as a typical person's desktop PC.
- Failures occur less frequently while writing/erasing data, which means there is a lower chance of irrecoverable data damage.[23]
- Defragmenting the SSD is unnecessary. Since SSDs are random access by nature and can perform parallel reads on multiple sections of the drive (as opposed to a HDD, which requires seek time for each fragment, assuming a single head assembly), a certain degree of fragmentation is actually better for reads, and wear leveling intrinsically induces fragmentation.[24] In fact, defragmenting a SSD is harmful since it adds wear to the SSD for no benefit.[25]

Disadvantages

- Flash-memory drives have limited lifetimes and will often wear out after 1,000,000 to 2,000,000 write cycles (1,000 to 10,000 per cell) for MLC, and up to 5,000,000 write cycles (100,000 per cell) for SLC.[26][27][28][29] Special file systems or firmware designs can mitigate this problem by spreading writes over the entire device, called wear leveling.[30]
- Wear leveling used on flash-based SSDs has security implications. For example, encryption of existing unencrypted data on flash-based SSDs cannot be performed securely due to the fact that wear leveling causes new encrypted drive sectors to be written to a physical location different from their original location—data remains unencrypted in the original physical location. It is also impossible to securely wipe files by overwriting their content on flash-based SSDs.[*citation needed*]
- As of early-2010, SSDs are still more expensive per gigabyte than hard drives. Whereas a normal flash drive is US$2 per gigabyte, hard drives are around US$0.10 per gigabyte for 3.5", or US$0.20 for 2.5".
- The capacity of SSDs is currently lower than that of hard drives. However, flash SSD capacity is predicted to increase rapidly, with drives of 1 TB already released for enterprise and industrial applications.[21][31][32][33][34]
- Asymmetric read vs. write performance can cause problems with certain functions where the read and write operations are expected to be completed in a similar timeframe. SSDs currently have a much slower write performance compared to their read performance.[35]
- Similarly, SSD write performance is significantly impacted by the availability of free, programmable blocks. Previously written data blocks that are no longer in use can be reclaimed by TRIM; however, even with TRIM, fewer free, programmable blocks translates into reduced performance.[36]
- As a result of wear leveling and write combining, the performance of SSDs degrades with use.[37][38]
- SATA-based SSDs generally exhibit much slower write speeds. As erase blocks on flash-based SSDs generally are quite large (e.g. 0.5 - 1 megabyte),[9] they are far slower than conventional disks during small writes (*write amplification* effect) and can suffer from write fragmentation.[39] Modern PCIe SSDs however have much faster write speeds than previously available.
- DRAM-based SSDs (but not flash-based SSDs) require more power than hard disks, when operating; they still use power when the computer is turned off, while hard disks do not.[40]

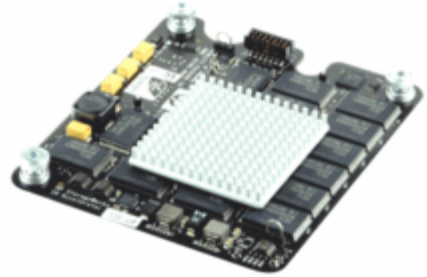
Cost and capacity

- Until recently,[*when?*] flash based solid-state drives were too costly for widespread use in mobile computing.[*citation needed*] As flash manufacturers transition from NOR flash to single-level cell (SLC) NAND flash and most recently to multi-level cell (MLC) NAND flash to maximize silicon die usage and reduce associated costs, "solid-state disks" are now being more accurately renamed "solid-state drives" – they have no disks but function as drives – for mobile computing in the enterprise and consumer electronics space.
- This technological trend is accompanied by an annual 50% decline in raw flash material costs, while capacities continue to double at the same rate. As a result, flash-based solid-state drives are becoming increasingly popular in markets such as notebook PCs and sub-notebooks for enterprises, Ultra-Mobile PCs (UMPC), and Tablet PCs for the healthcare and consumer electronics sectors. Major PC companies have now started to offer such technology.

Availability

- An SSD in standard 2.5-inch (64 mm) form-factor
- DDR SDRAM based SSD

- PCI attached IO Accelerator SSD



- PCI-E / DRAM / NAND based SSD

Hybrid drive

- A hybrid disk uses an SSD as a buffer for a larger hard disk drive. The hard disk may be spun down more of the time if data is available in the SSD.
- NAND Flash based SSDs offer a potential power saving; however, the typical pattern of usage of normal operations result in cache misses in the NAND Flash as well leading to continued spin of the drive platter or much longer latency if the drive needed to spin up.[*citation needed*] These devices would be slightly more energy efficient but could not prove to be any better in performance.[*citation needed*]
- DRAM-based SSDs may also work as a buffer cache mechanism (see hybrid RAM drive). When data is written to memory, the corresponding block in memory is marked as dirty, and all dirty blocks can be flushed to the actual hard drive based on the following criteria:
- Time (e.g., every 10 seconds, flush all dirty data);
- Threshold (when the ratio of dirty data to SSD size exceeds some predetermined value, flush the dirty data);
- Loss of power/computer shutdown.

Intel® X25-E SATA Solid-State Drive

3.1 Capacity

Table 2. User Addressable Sectors

Unformatted Capacity	Total User Addressable Sectors in LBA Mode
32 GB	62,500,000
64 GB	125,045,424

Notes:
1. 1 GB = 1,000,000,000 Byte and not all of the memory can be used for data storage.
2. 1 Sector = 512 Byte.

3.2 Performance

Table 3. Maximum Sustained Read and Write Bandwidth

Access Type	MB/s
Sequential Read	up to 250
Sequential Write	up to 170

Table 4. Random Read and Write Input/Output Operations per Second (IOPS)

Access Type	IOPS
4K Read	35,000
4K Write	3,300


TOSHIBA
SSD
Solid State Drive
128 GB