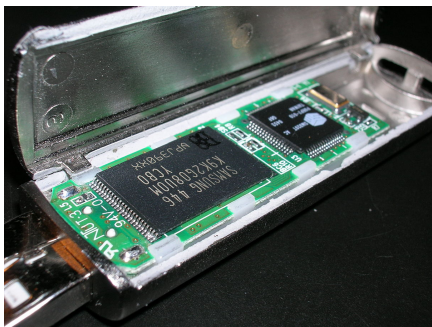


Flash memory



A USB flash drive. The chip on the left is the flash memory. The controller is on the right.

Computer memory types
Volatile
RAM
<ul style="list-style-type: none">DRAM (e.g., DDR SDRAM)SRAM
In development
<ul style="list-style-type: none">T-RAMZ-RAMTTRAM
Historical
<ul style="list-style-type: none">Williams-Kilburn tube (1946-47)Delay line memory (1947)Selectron tube (1953)Dekatron
Non-volatile
ROM
<ul style="list-style-type: none">Mask ROMPROMEPROMEEPROM
NVRAM
<ul style="list-style-type: none">Flash memory
Early stage NVRAM
<ul style="list-style-type: none">nvSRAMFeRAMMRAMPRAM
Mechanical
<ul style="list-style-type: none">Magnetic tapeHard driveOptical drive

In development
<ul style="list-style-type: none"> • CBRAM • SONOS • RRAM • Racetrack memory • NRAM • Millipede memory • FJG
Historical
<ul style="list-style-type: none"> • Paper data storage (1725) • Drum memory (1932) • Magnetic-core memory (1949) • Plated wire memory (1957) • Thin-film memory (1962) • Twistor memory (~1968) • Bubble memory (~1970)

Flash memory is an electronic non-volatile computer storage device that can be electrically erased and reprogrammed.

Flash memory developed from EEPROM (electrically erasable programmable read-only memory). There are two main types of flash memory, which are named after the NAND and NOR logic gates. The internal characteristics of the individual flash memory cells exhibit characteristics similar to those of the corresponding gates.

Whereas EPROMs had to be completely erased before being rewritten, NAND type flash memory may be written and read in blocks (or pages) which are generally much smaller than the entire device. The NOR type allows a single machine word (byte) to be written or read independently.

The NAND type is primarily used in main memory, memory cards, USB flash drives, solid-state drives, and similar products, for general storage and transfer of data. The NOR type, which allows true random access and therefore direct code execution, is used as a replacement for the older EPROM and as an alternative to certain kinds of ROM applications, whereas NOR flash memory may emulate ROM primarily at the machine code level; many digital designs need ROM (or PLA) structures for other uses, often at significantly higher speeds than (economical) flash memory may achieve.^[citation needed] NAND or NOR flash memory is also often used to store configuration data in numerous digital products, a task previously made possible by EEPROMs or battery-powered static RAM.

Example applications of both types of flash memory include personal computers, PDAs, digital audio players, digital cameras, mobile phones, synthesizers, video games, scientific instrumentation, industrial robotics, medical electronics, and so on. In addition to being non-volatile, flash memory offers fast read access times, as fast as dynamic RAM, although not as fast as static RAM or ROM. Its mechanical shock resistance helps explain its popularity over hard disks in portable devices, as does its high durability, being able to withstand high pressure, temperature, immersion in water, etc.

Although flash memory is technically a type of EEPROM, the term "EEPROM" is generally used to refer specifically to non-flash EEPROM which is erasable in small blocks, typically bytes. Because erase cycles are slow, the large block sizes used in flash memory erasing give it a significant speed advantage over old-style EEPROM when writing large amounts of data. As of 2013[1] flash memory costs much less than byte-programmable EEPROM and has become the dominant memory type wherever a system requires a significant amount of non-volatile, solid state storage.

History

Flash memory (both NOR and NAND types) was invented by Dr. Fujio Masuoka while working for Toshiba circa 1980. According to Toshiba, the name "flash" was suggested by Masuoka's colleague, Shōji Ariizumi, because the erasure process of the memory contents reminded him of the flash of a camera.^[citation needed] Masuoka and colleagues presented the invention at the *IEEE 1984 International Electron Devices Meeting* (IEDM) held in San Francisco.

Intel Corporation saw the massive potential of the invention and introduced the first commercial NOR type flash chip in 1988. NOR-based flash has long erase and write times, but provides full address and data buses, allowing random access to any memory location. This makes it a suitable replacement for older read-only memory (ROM) chips, which are used to store program code that rarely needs to be updated, such as a computer's BIOS or the firmware of set-top boxes. Its endurance may be from as little as 100 erase cycles for an on-chip flash memory, to a more typical 10,000 or 100,000 erase cycles, up to 1,000,000 erase cycles. NOR-based flash was the basis of early flash-based removable media; CompactFlash was originally based on it, though later cards moved to less expensive NAND flash.

NAND flash has reduced erase and write times, and requires less chip area per cell, thus allowing greater storage density and lower cost per bit than NOR flash; it also has up to ten times the endurance of NOR flash. However, the I/O interface of NAND flash does not provide a random-access external address bus. Rather, data must be read on a block-wise basis, with typical block sizes of hundreds to thousands of bits. This makes NAND flash unsuitable as a drop-in replacement for program ROM, since most microprocessors and microcontrollers required byte-level random access. In this regard, NAND flash is similar to other secondary data storage devices, such as hard disks and optical media, and is thus very suitable for use in mass-storage devices, such as memory cards. The first NAND-based removable media format was SmartMedia in 1995, and many others have followed, including MultiMediaCard, Secure Digital, Memory Stick and xD-Picture Card. A new generation of memory card formats, including RS-MMC, miniSD and microSD, and Intelligent Stick, feature extremely small form factors. For example, the microSD card has an area of just over 1.5 cm², with a thickness of less than 1 mm. microSD capacities range from 64 MB to 64 GB, as of May 2011.

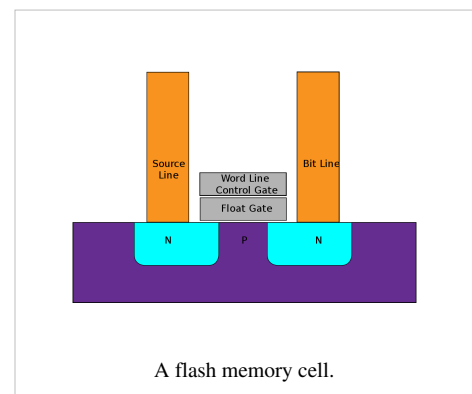
Principles of operation

Flash memory stores information in an array of memory cells made from floating-gate transistors. In traditional single-level cell (SLC) devices, each cell stores only one bit of information. Some newer flash memory, known as multi-level cell (MLC) devices, including triple-level cell (TLC) devices, can store more than one bit per cell by choosing between multiple levels of electrical charge to apply to the floating gates of its cells.

The floating gate may be conductive (typically polysilicon in most kinds of flash memory) or non-conductive (as in SONOS flash memory).^[2]

Floating-gate transistor

In flash memory, each memory cell resembles a standard MOSFET, except the transistor has two gates instead of one. On top is the control gate (CG), as in other MOS transistors, but below this there is a floating gate (FG) insulated all around by an oxide layer. The FG is interposed between the CG and the MOSFET channel. Because the FG is electrically isolated by its insulating layer, any electrons placed on it are trapped there and, under normal conditions, will not discharge for many years. When the FG holds a charge, it screens (partially cancels) the electric

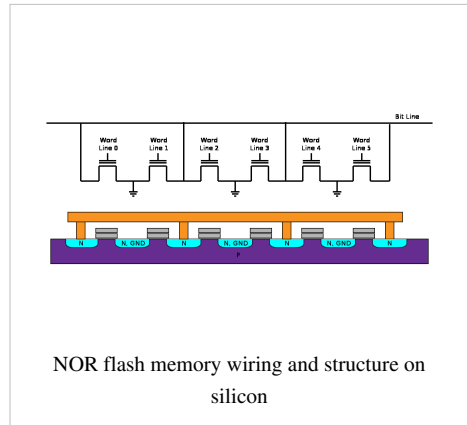


field from the CG, which modifies the threshold voltage (V_T) of the cell (more voltage has to be applied to the CG to make the channel conduct). For read-out, a voltage intermediate between the possible threshold voltages is applied to the CG, and the MOSFET channel's conductivity tested (if it's conducting or insulating), which is influenced by the FG. The current flow through the MOSFET channel is sensed and forms a binary code, reproducing the stored data. In a multi-level cell device, which stores more than one bit per cell, the amount of current flow is sensed (rather than simply its presence or absence), in order to determine more precisely the level of charge on the FG.

Internal charge pumps

Despite the need for high programming and erasing voltages, virtually all flash chips today require only a single supply voltage, and produce the high voltages via on-chip charge pumps.

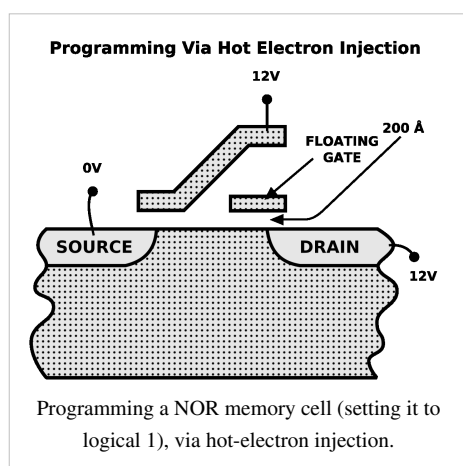
Over half the energy used by a 1.8 V NAND flash chip is lost in the charge pump itself. Since boost converters are inherently more efficient than charge pumps, researchers developing low-power SSDs have proposed returning to the dual V_{cc}/V_{pp} supply voltages used on all the early flash chips, driving the high V_{pp} voltage for all flash chips in a SSD with a single shared external boost converter.^{[3][4][5][6][7][8]}



In spacecraft and other high-radiation environments, the on-chip charge pump is the first part of the flash chip to fail. Flash memories continue to work at much higher radiation levels in read-only mode.^[9]

NOR flash

In NOR gate flash, each cell has one end connected directly to ground, and the other end connected directly to a bit line. This arrangement is called "NOR flash" because it acts like a NOR gate: when one of the word lines (connected to the cell's CG) is brought high, the corresponding storage transistor acts to pull the output bit line low. NOR flash continues to be the technology of choice for embedded applications requiring a discrete non-volatile memory device. The low read latencies characteristic of NOR devices allow for both direct code execution and data storage in a single memory product.



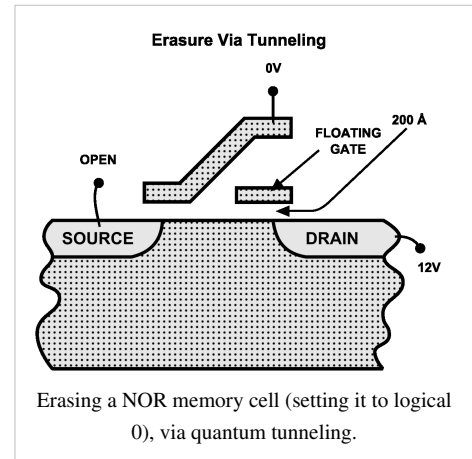
Programming

A single-level NOR flash cell in its default state is logically equivalent to a binary "1" value, because current will flow through the channel under application of an appropriate voltage to the control gate, so that the bitline voltage is pulled down. A NOR flash cell can be programmed, or set to a binary "0" value, by the following procedure:

- an elevated on-voltage (typically >5 V) is applied to the CG
- the channel is now turned on, so electrons can flow from the source to the drain (assuming an NMOS transistor)
- the source-drain current is sufficiently high to cause some high energy electrons to jump through the insulating layer onto the FG, via a process called hot-electron injection

Erasing

To erase a NOR flash cell (resetting it to the "1" state), a large voltage of *the opposite polarity* is applied between the CG and source terminal, pulling the electrons off the FG through quantum tunneling. Modern NOR flash memory chips are divided into erase segments (often called blocks or sectors). The erase operation can only be performed on a block-wise basis; all the cells in an erase segment must be erased together. Programming of NOR cells, however, can generally be performed one byte or word at a time.



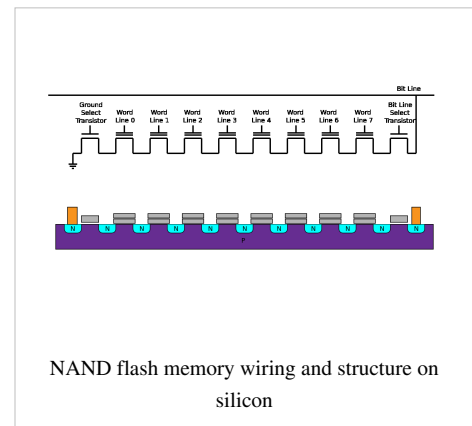
NAND

NAND flash also uses floating-gate transistors, but they are connected in a way that resembles a NAND gate: several transistors are connected in series, and only if all word lines are pulled high (above the transistors' V_T) is the bit line pulled low. These groups are then connected via some additional transistors to a NOR-style bit line array in the same way that single transistors are linked in NOR flash.

Compared to NOR flash, replacing single transistors with serial-linked groups adds an extra level of addressing. Whereas NOR flash might address memory by page then word, NAND flash might address it by page, word and bit. Bit-level addressing suits bit-serial applications (such as hard disk emulation), which access only 1 bit at a time. Execute-In-Place applications, on the other hand, require every bit in a word to be accessed simultaneously. This requires word-level addressing. In any case, both bit and word addressing modes are possible with either NOR or NAND flash.

To read, first the desired group is selected (in the same way that a single transistor is selected from a NOR array). Next, most of the word lines are pulled up above the V_T of a programmed bit, while one of them is pulled up to just over the V_T of an erased bit. The series group will conduct (and pull the bit line low) if the selected bit has not been programmed.

Despite the additional transistors, the reduction in ground wires and bit lines allows a denser layout and greater storage capacity per chip. (The ground wires and bit lines are actually much wider than the lines in the diagrams.) In addition, NAND flash is typically permitted to contain a certain number of faults (NOR flash, as is used for a BIOS ROM, is expected to be fault-free). Manufacturers try to maximize the amount of usable storage by shrinking the size of the transistor below the size where they can be made reliably, to the size where further reductions would increase the number of faults faster than it would increase the total storage available.



Writing and erasing

NAND flash uses tunnel injection for writing and tunnel release for erasing. NAND flash memory forms the core of the removable USB storage devices known as USB flash drives, as well as most memory card formats and solid-state drives available today.

Vertical NAND

Vertical NAND (V-NAND) memory stacks memory cells vertically and uses a charge trap geometry architecture. The vertical layers allow larger areal bit densities without requiring smaller individual cells.

Structure

V-NAND uses a charge trap flash geometry (pioneered in 2002 by AMD) that stores charge on an embedded silicon nitride film. Such a film is more robust against point defects and can be made thicker to hold larger numbers of electrons. V-NAND wraps a planar charge trap cell into a cylindrical form.

An individual memory cell is made up of one planar polysilicon layer containing a hole filled by multiple concentric vertical cylinders. The hole's polysilicon surface acts as the gate electrode. The outermost silicon dioxide cylinder acts as the gate dielectric, enclosing a silicon nitride cylinder that stores charge, in turn enclosing a silicon dioxide cylinder as the tunnel dielectric that surrounds a central rod of conducting polysilicon which acts as the conducting channel.

Memory cells in different vertical layers do not interfere with each other, as the charges cannot move vertically through the silicon nitride storage medium, and the electric fields associated with the gates are closely confined within each layer. The vertical collection is electrically identical to the serial-linked groups in which conventional NAND flash memory is configured.

Construction

Growth of a group of V-NAND cells begins with an alternating stack of conducting (doped) polysilicon layers and insulating silicon dioxide layers.

The next step is to form a cylindrical hole through these layers. In practice, a 128 Gbit V-NAND chip with 24 layers of memory cells requires about 2.9 billion such holes. Next the hole's inner surface receives multiple coatings, first silicon dioxide, then silicon nitride, then a second layer of silicon dioxide. Finally, the hole is filled with conducting (doped) polysilicon.

Performance

As of 2013 V-NAND flash reads and writes twice as fast as conventional NAND. It lasts 10 times longer. V-NAND consumes 50 percent less power. As of 2013 they offer comparable physical bit density, using 10-nm lithography, but potentially offer as much as two orders of magnitude of increased density.

Limitations

Block erasure

One limitation of flash memory is that, although it can be read or programmed a byte or a word at a time in a random access fashion, it can only be erased a "block" at a time. This generally sets all bits in the block to 1. Starting with a freshly erased block, any location within that block can be programmed. However, once a bit has been set to 0, only by erasing the entire block can it be changed back to 1. In other words, flash memory (specifically NOR flash) offers random-access read and programming operations, but does not offer arbitrary random-access rewrite or erase operations. A location can, however, be rewritten as long as the new value's 0 bits are a superset of the over-written values. For example, a nibble value may be erased to 1111, then written e.g. as 1110. Successive writes to that nibble

can change it to 1010, then 0010, and finally 0000. Essentially, erasure sets all bits to 1, and programming can only clear bits to 0. File systems designed for flash devices can make use of this capability, for example to represent sector metadata.

Although data structures in flash memory cannot be updated in completely general ways, this allows members to be "removed" by marking them as invalid. This technique may need to be modified for multi-level cell devices, where one memory cell holds more than one bit.

Common flash devices such as USB flash drives and memory cards provide only a block-level interface, or flash translation layer (FTL), which writes to a different cell each time to wear-level the device. This prevents incremental writing within a block, however it does not help the device from being prematurely worn out by poorly designed systems. For example, nearly all consumer devices ship formatted with the MS-FAT file system, which pre-dates flash memory, having been designed for DOS and disk media.

Memory wear

Another limitation is that flash memory has a finite number of program-erase cycles (typically written as P/E cycles). Most commercially available flash products are guaranteed to withstand around 100,000 P/E cycles before the wear begins to deteriorate the integrity of the storage. Micron Technology and Sun Microsystems announced an SLC NAND flash memory chip rated for 1,000,000 P/E cycles on 17 December 2008.

The guaranteed cycle count may apply only to block zero (as is the case with TSOP NAND devices), or to all blocks (as in NOR). This effect is partially offset in some chip firmware or file system drivers by counting the writes and dynamically remapping blocks in order to spread write operations between sectors; this technique is called wear leveling. Another approach is to perform write verification and remapping to spare sectors in case of write failure, a technique called bad block management (BBM). For portable consumer devices, these wearout management techniques typically extend the life of the flash memory beyond the life of the device itself, and some data loss may be acceptable in these applications. For high reliability data storage, however, it is not advisable to use flash memory that would have to go through a large number of programming cycles. This limitation is meaningless for 'read-only' applications such as thin clients and routers, which are programmed only once or at most a few times during their lifetimes.

Recently, Taiwanese engineers from Macronix revealed their intention to announce at the 2012 IEEE International Electron Devices Meeting that it has figured out how to improve NAND flash storage read/write cycles from 10,000 to 100 million cycles using a "self-healing" process that uses a flash chip with "onboard heaters that could anneal small groups of memory cells."^[10]

The built-in thermal annealing replaces the usual erase cycle with a local high temperature process that not only erases the stored charge, but also repairs the electron induced stress in the chip, giving write cycles of at least 100 million.^[11]

The result is a chip that can be erased and rewritten on over and over, even when it should theoretically break down. As promising as Macronix's breakthrough could be for the mobile industry, however, there are no plans for a commercial product to be released any time in the near future.^[12]

Read disturb

The method used to read NAND flash memory can cause nearby cells in the same memory block to change over time (become programmed). This is known as read disturb. The threshold number of reads is generally in the hundreds of thousands of reads between intervening erase operations. If reading continually from one cell, that cell will not fail but rather one of the surrounding cells on a subsequent read. To avoid the read disturb problem the flash controller will typically count the total number of reads to a block since the last erase. When the count exceeds a target limit, the affected block is copied over to a new block, erased, then released to the block pool. The original block is as good as new after the erase. If the flash controller does not intervene in time, however, a **read disturb**

error will occur with possible data loss if the errors are too numerous to correct with ECC.

Low-level access

The low-level interface to flash memory chips differs from those of other memory types such as DRAM, ROM, and EEPROM, which support bit-alterability (both zero to one and one to zero) and random access via externally accessible address buses.

NOR memory has an external address bus for reading and programming. For NOR memory, reading and programming are random-access, and unlocking and erasing are block-wise. For NAND memory, reading and programming are page-wise, and unlocking and erasing are block-wise.

NOR memories

Reading from NOR flash is similar to reading from random-access memory, provided the address and data bus are mapped correctly. Because of this, most microprocessors can use NOR flash memory as execute in place (XIP) memory, meaning that programs stored in NOR flash can be executed directly from the NOR flash without needing to be copied into RAM first. NOR flash may be programmed in a random-access manner similar to reading. Programming changes bits from a logical one to a zero. Bits that are already zero are left unchanged. Erasure must happen a block at a time, and resets all the bits in the erased block back to one. Typical block sizes are 64, 128, or 256 KB.

Bad block management is a relatively new feature in NOR chips. In older NOR devices not supporting bad block management, the software or device driver controlling the memory chip must correct for blocks that wear out, or the device will cease to work reliably.

The specific commands used to lock, unlock, program, or erase NOR memories differ for each manufacturer. To avoid needing unique driver software for every device made, special Common Flash Memory Interface (CFI) commands allow the device to identify itself and its critical operating parameters.

Besides its use as random-access ROM, NOR flash can also be used as a storage device, by taking advantage of random-access programming. Some devices offer read-while-write functionality so that code continues to execute even while a program or erase operation is occurring in the background. For sequential data writes, NOR flash chips typically have slow write speeds, compared with NAND flash.

NAND memories

NAND flash architecture was introduced by Toshiba in 1989. These memories are accessed much like block devices, such as hard disks or memory cards. Each block consists of a number of pages. The pages are typically 512 or 2,048 or 4,096 bytes in size. Associated with each page are a few bytes (typically 1/32 of the data size) that can be used for storage of an error correcting code (ECC) checksum.

Typical block sizes include:

- 32 pages of 512+16 bytes each for a block size of 16 KB
- 64 pages of 2,048+64 bytes each for a block size of 128 KB^[13]
- 64 pages of 4,096+128 bytes each for a block size of 256 KB^[14]
- 128 pages of 4,096+128 bytes each for a block size of 512 KB.

While reading and programming is performed on a page basis, erasure can only be performed on a block basis. Number of Operations (NOPs) is the number of times the pages can be programmed. So far, this number for MLC flash is always one while, for SLC flash, it is four.^[citation needed]

NAND devices also require bad block management by the device driver software, or by a separate controller chip. SD cards, for example, include controller circuitry to perform bad block management and wear leveling. When a logical block is accessed by high-level software, it is mapped to a physical block by the device driver or controller. A

number of blocks on the flash chip may be set aside for storing mapping tables to deal with bad blocks, or the system may simply check each block at power-up to create a bad block map in RAM. The overall memory capacity gradually shrinks as more blocks are marked as bad.

NAND relies on ECC to compensate for bits that may spontaneously fail during normal device operation. A typical ECC will correct a one-bit error in each 2048 bits (256 bytes) using 22 bits of ECC code, or a one-bit error in each 4096 bits (512 bytes) using 24 bits of ECC code. If the ECC cannot correct the error during read, it may still detect the error. When doing erase or program operations, the device can detect blocks that fail to program or erase and mark them bad. The data is then written to a different, good block, and the bad block map is updated.

Most NAND devices are shipped from the factory with some bad blocks. These are typically marked according to a specified bad block marking strategy. By allowing some bad blocks, the manufacturers achieve far higher yields than would be possible if all blocks had to be verified good. This significantly reduces NAND flash costs and only slightly decreases the storage capacity of the parts.

When executing software from NAND memories, virtual memory strategies are often used: memory contents must first be paged or copied into memory-mapped RAM and executed there (leading to the common combination of NAND + RAM). A memory management unit (MMU) in the system is helpful, but this can also be accomplished with overlays. For this reason, some systems will use a combination of NOR and NAND memories, where a smaller NOR memory is used as software ROM and a larger NAND memory is partitioned with a file system for use as a non-volatile data storage area.

NAND sacrifices the random-access and execute-in-place advantages of NOR. NAND is best suited to systems requiring high capacity data storage. It offers higher densities, larger capacities, and lower cost. It has faster erases, sequential writes, and sequential reads.

Standardization

A group called the Open NAND Flash Interface Working Group (ONFI) has developed a standardized low-level interface for NAND flash chips. This allows interoperability between conforming NAND devices from different vendors. The ONFI specification version 1.0 was released on 28 December 2006. It specifies:

- a standard physical interface (pinout) for NAND flash in TSOP-48, WSOP-48, LGA-52, and BGA-63 packages
- a standard command set for reading, writing, and erasing NAND flash chips
- a mechanism for self-identification (comparable to the serial presence detection feature of SDRAM memory modules)

The ONFI group is supported by major NAND flash manufacturers, including Hynix, Intel, Micron Technology, and Numonyx, as well as by major manufacturers of devices incorporating NAND flash chips.^[15]

A group of vendors, including Intel, Dell, and Microsoft, formed a Non-Volatile Memory Host Controller Interface (NVMHCI) Working Group. The goal of the group is to provide standard software and hardware programming interfaces for nonvolatile memory subsystems, including the "flash cache" device connected to the PCI Express bus.

Distinction between NOR and NAND flash

NOR and NAND flash differ in two important ways:

- the connections of the individual memory cells are different
- the interface provided for reading and writing the memory is different (NOR allows random-access for reading, NAND allows only page access)

These two are linked by the design choices made in the development of NAND flash. A goal of NAND flash development was to reduce the chip area required to implement a given capacity of flash memory, and thereby to reduce cost per bit and increase maximum chip capacity so that flash memory could compete with magnetic storage devices like hard disks.^[citation needed]

NOR and NAND flash get their names from the structure of the interconnections between memory cells.^[16] In NOR flash, cells are connected in parallel to the bit lines, allowing cells to be read and programmed individually. The parallel connection of cells resembles the parallel connection of transistors in a CMOS NOR gate. In NAND flash, cells are connected in series, resembling a NAND gate. The series connections consume less space than parallel ones, reducing the cost of NAND flash. It does not, by itself, prevent NAND cells from being read and programmed individually.

When NOR flash was developed, it was envisioned as a more economical and conveniently rewritable ROM than contemporary EPROM and EEPROM memories. Thus random-access reading circuitry was necessary. However, it was expected that NOR flash ROM would be read much more often than written, so the write circuitry included was fairly slow and could only erase in a block-wise fashion. On the other hand, applications that use flash as a replacement for disk drives do not require word-level write address, which would only add to the complexity and cost unnecessarily.^[citation needed]

Because of the series connection and removal of wordline contacts, a large grid of NAND flash memory cells will occupy perhaps only 60% of the area of equivalent NOR cells (assuming the same CMOS process resolution, for example, 130 nm, 90 nm, or 65 nm). NAND flash's designers realized that the area of a NAND chip, and thus the cost, could be further reduced by removing the external address and data bus circuitry. Instead, external devices could communicate with NAND flash via sequential-accessed command and data registers, which would internally retrieve and output the necessary data. This design choice made random-access of NAND flash memory impossible, but the goal of NAND flash was to replace hard disks, not to replace ROMs.

Write endurance

The write endurance of SLC floating-gate NOR flash is typically equal to or greater than that of NAND flash, while MLC NOR and NAND flash have similar endurance capabilities. Example Endurance cycle ratings listed in datasheets for NAND and NOR flash are provided.

- SLC NAND flash is typically rated at about 100k cycles (Samsung OneNAND KFW4G16Q2M)
- MLC NAND flash is typically rated at about 5k – 10k cycles for medium-capacity applications (Samsung K9G8G08U0M) and 1k – 3k cycles for high-capacity applications
- TLC NAND flash is typically rated at about 1k cycles (Samsung 840)
- SLC floating-gate NOR flash has typical endurance rating of 100k to 1M cycles (Numonyx M58BW 100k; Spansion S29CD016J 1,000k)
- MLC floating-gate NOR flash has typical endurance rating of 100k cycles (Numonyx J3 flash)

However, by applying certain algorithms and design paradigms such as wear leveling and memory over-provisioning, the endurance of a storage system can be tuned to serve specific requirements.

Flash file systems

Because of the particular characteristics of flash memory, it is best used with either a controller to perform wear leveling and error correction or specifically designed flash file systems, which spread writes over the media and deal with the long erase times of NOR flash blocks. The basic concept behind flash file systems is the following: when the flash store is to be updated, the file system will write a new copy of the changed data to a fresh block, remap the file pointers, then erase the old block later when it has time.

In practice, flash file systems are only used for memory technology devices (MTDs), which are embedded flash memories that do not have a controller. Removable flash memory cards and USB flash drives have built-in controllers to perform wear leveling and error correction so use of a specific flash file system does not add any benefit.

Capacity

Multiple chips are often arrayed to achieve higher capacities for use in consumer electronic devices such as multimedia players or GPSs. The capacity of flash chips generally follows Moore's Law because they are manufactured with many of the same integrated circuits techniques and equipment.

Consumer flash storage devices typically are advertised with usable sizes expressed as a small integral power of two (2, 4, 8, etc.) and a designation of megabytes (MB) or gigabytes (GB); e.g., 512 MB, 8 GB. Device packaging uses "decimal prefixes," meaning 1,000,000 bytes and 1,000,000,000 bytes, respectively.^[citation needed] This includes SSDs marketed as hard drive replacements, in accordance with traditional hard drives, which also use decimal prefixes. Thus, an SSD marked as "64 GB" is actually at least $64 \times 1,000^3$ bytes (64 GB), or often a bit more. Most users will have slightly less capacity than this available for their files, due to the space taken by file system metadata. The flash memory chips inside them are sized in strict binary multiples, but the actual total capacity of the chips is not usable at the drive interface. It is considerably larger than the advertised capacity in order to allow for distribution of writes (wear leveling), for sparing, for error correction codes, and for other metadata needed by the device's internal firmware.

In 2005, Toshiba and SanDisk developed a NAND flash chip capable of storing 1 GB of data using multi-level cell (MLC) technology, capable of storing two bits of data per cell. In September 2005, Samsung Electronics announced that it had developed the world's first 2 GB chip.

In March 2006, Samsung announced flash hard drives with a capacity of 4 GB, essentially the same order of magnitude as smaller laptop hard drives, and in September 2006, Samsung announced an 8 GB chip produced using a 40 nm manufacturing process. In January 2008, Sandisk announced availability of their 16 GB MicroSDHC and 32 GB SDHC Plus cards.^{[17][18]}

More recent flash drives (as of 2012) have much greater capacities, holding 64, 128, and 256 GB.^[19]

There are still flash chips manufactured with capacities under or around 1 MB, e.g., for BIOS-ROMs and embedded applications.

Transfer rates

NAND flash memory cards are much faster at reading than writing so it is the maximum read speed that is commonly advertised.

As a chip wears out, its erase/program operations slow down considerably,^[citation needed] requiring more retries and bad block remapping. Transferring multiple small files, each smaller than the chip-specific block size, could lead to a much lower rate. Access latency also influences performance, but less so than with their hard drive counterpart.

The speed is sometimes quoted in MB/s (megabytes per second), or as a multiple of that of a legacy single speed CD-ROM, such as 60×, 100× or 150×. Here 1× is equivalent to 150 kB/s. For example, a 100× memory card gives

150 kB/s \times 100 = 15,000 kB/s.^[citation needed]

Performance also depends on the quality of memory controllers. Even when the only change to manufacturing is die-shrink, the absence of an appropriate controller can result in degraded speeds.^[20]

Applications

Serial flash

Serial flash is a small, low-power flash memory that uses a serial interface, typically Serial Peripheral Interface Bus (SPI), for sequential data access. When incorporated into an embedded system, serial flash requires fewer wires on the PCB than parallel flash memories, since it transmits and receives data one bit at a time. This may permit a reduction in board space, power consumption, and total system cost.

There are several reasons why a serial device, with fewer external pins than a parallel device, can significantly reduce overall cost:

- Many ASICs are pad-limited, meaning that the size of the die is constrained by the number of wire bond pads, rather than the complexity and number of gates used for the device logic. Eliminating bond pads thus permits a more compact integrated circuit, on a smaller die; this increases the number of dies that may be fabricated on a wafer, and thus reduces the cost per die.
- Reducing the number of external pins also reduces assembly and packaging costs. A serial device may be packaged in a smaller and simpler package than a parallel device.
- Smaller and lower pin-count packages occupy less PCB area.
- Lower pin-count devices simplify PCB routing.

There are two major SPI flash types. The first type is characterized by small pages and one or more internal SRAM page buffers allowing a complete page to be read to the buffer, partially modified, and then written back (for example, the Atmel AT45 *DataFlash* or the Micron Technology Page Erase NOR Flash). The second type has larger sectors. The smallest sectors typically found in an SPI flash are 4 kB, but they can be as large as 64 kB. Since the SPI flash lacks an internal SRAM buffer, the complete page must be read out and modified before being written back, making it slow to manage. *SPI flash* is cheaper than *DataFlash* and is therefore a good choice when the application is code shadowing.

The two types are not easily exchangeable, since they do not have the same pinout, and the command sets are incompatible.

Firmware storage

With the increasing speed of modern CPUs, parallel flash devices are often much slower than the memory bus of the computer they are connected to. Conversely, modern SRAM offers access times below 10 ns, while DDR2 SDRAM offers access times below 20 ns. Because of this, it is often desirable to shadow code stored in flash into RAM; that is, the code is copied from flash into RAM before execution, so that the CPU may access it at full speed. Device firmware may be stored in a serial flash device, and then copied into SDRAM or SRAM when the device is powered-up.^[21] Using an external serial flash device rather than on-chip flash removes the need for significant process compromise (a process that is good for high-speed logic is generally not good for flash and vice-versa). Once it is decided to read the firmware in as one big block it is common to add compression to allow a smaller flash chip to be used. Typical applications for serial flash include storing firmware for hard drives, Ethernet controllers, DSL modems, wireless network devices, etc.

Flash memory as a replacement for hard drives

One more recent application for flash memory is as a replacement for hard disks. Flash memory does not have the mechanical limitations and latencies of hard drives, so a solid-state drive (SSD) is attractive when considering speed, noise, power consumption, and reliability. Flash drives are gaining traction as mobile device secondary storage devices; they are also used as substitutes for hard drives in high-performance desktop computers and some servers with RAID and SAN architectures.

There remain some aspects of flash-based SSDs that make them unattractive. The cost per gigabyte of flash memory remains significantly higher than that of hard disks. Also flash memory has a finite number of P/E cycles, but this seems to be currently under control since warranties on flash-based SSDs are approaching those of current hard drives.

For relational databases or other systems that require ACID transactions, even a modest amount of Flash storage can offer vast speedups over arrays of disk drives.^{[22][23]}

In June 2006, Samsung Electronics released the first flash-memory based PCs, the Q1-SSD and Q30-SSD, both of which used 32 GB SSDs, and were at least initially available only in South Korea.

A solid-state drive was offered as an option with the first Macbook Air introduced in 2008, and from 2010 onwards, all Macbook Air laptops shipped with an SSD. Starting in late 2011, as part of Intel's Ultrabook initiative, an increasing number of ultra thin laptops are being shipped with SSDs standard.

There are also hybrid techniques such as hybrid drive and ReadyBoost that attempt to combine the advantages of both technologies, using flash as a high-speed non-volatile cache for files on the disk that are often referenced, but rarely modified, such as application and operating system executable files.

Flash memory as RAM

As of 2012, there are attempts to use flash memory as the main computer memory, DRAM.^[24] In this role, it is slower than conventional DRAM, but uses up to ten times less power and is also significantly cheaper.^[citation needed]

Archival or long-term storage

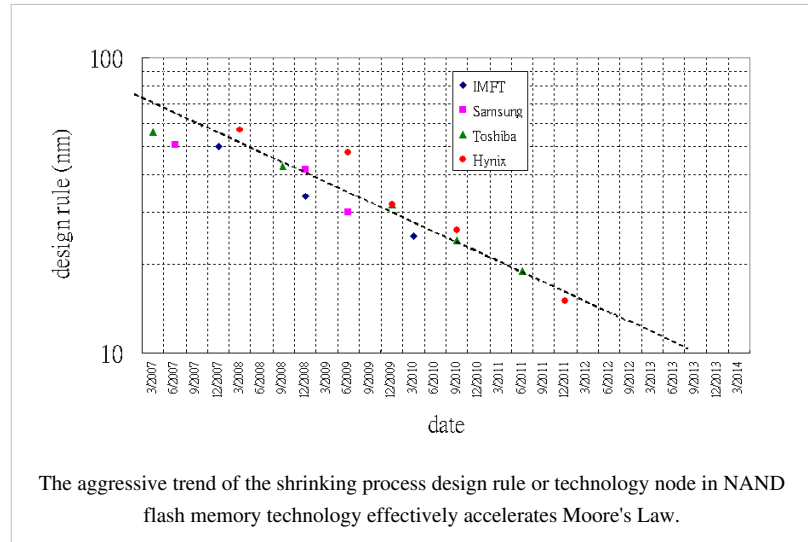
It is unclear how long flash memory will persist under archival conditions—i.e., benign temperature and humidity with infrequent access with or without prophylactic rewrite. Anecdotal evidence suggests that the technology is reasonably robust on the scale of years.

Industry

One source states that, in 2008, the flash memory industry includes about US\$9.1 billion in production and sales. Other sources put the flash memory market at a size of more than US\$20 billion in 2006, accounting for more than eight percent of the overall semiconductor market and more than 34 percent of the total semiconductor memory market. In 2012, the market was estimated at \$26.8 billion.^[25]

Flash scalability

Due to its relatively simple structure and high demand for higher capacity, NAND flash memory is the most aggressively scaled technology among electronic devices. The heavy competition among the top few manufacturers only adds to the aggressiveness in shrinking the design rule or process technology node. While the expected shrink timeline is a factor of two every three years per original version of Moore's law, this has recently been accelerated in the case of NAND flash to a factor of two every two years. In November 2012, Samsung announced that it had started production of 10 nm scale chips, which implies a minimum geometry of between 10 and 19 nm.^{[26][27]}



As the feature size of flash memory cells reaches the minimum limit, further flash density increases will be driven by greater levels of MLC, possibly 3-D stacking of transistors, and improvements to the manufacturing process. The decrease in endurance and increase in uncorrectable bit error rates that accompany feature size shrinking can be compensated by improved error correction mechanisms. Even with these advances, it may be impossible to economically scale flash to smaller and smaller dimensions. Many promising new technologies (such as FeRAM, MRAM, PMC, PCM, ReRAM, and others) are under investigation and development as possible more scalable replacements for flash.

References

- [1] http://en.wikipedia.org/w/index.php?title=Flash_memory&action=edit
- [2] "PSoC Designer(TM) Device Selection Guide – AN2209" (<http://www.cypress.com/?rID=2750>): "... The PSoC ... utilizes a unique Flash process: SONOS"
- [3] (abstract) (<http://adsabs.harvard.edu/abs/2010IEITE..93..317Y>).
- [4] In particular, pp 515-536: K. Takeuchi. "Low power 3D-integrated SSD" (http://link.springer.com/content/pdf/10.1007/978-90-481-9431-5_18.pdf)
- [5] Tadashi Yasufuku et. al., "Inductor and TSV Design of 20-V Boost Converter for Low Power 3D Solid State Drive with NAND Flash Memories" (http://www.researchgate.net/publication/220240029_Inductor_and_TSV_Design_of_20-V_Boost_Converter_for_Low_Power_3D_Solid_State_Drive_with_NAND_Flash_Memories). 2010.
- [6] Hatanaka, T. and Takeuchi, K. "4-times faster rising VPASS (10V), 15% lower power VPGM (20V), wide output voltage range voltage generator system for 4-times faster 3D-integrated solid-state drives" (<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5986104>). 2011.
- [7] Takeuchi, K., "Low power 3D-integrated Solid-State Drive (SSD) with adaptive voltage generator" (<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5488397>). 2010.

- [8] Ishida, K. et. al., "1.8 V Low-Transient-Energy Adaptive Program-Voltage Generator Based on Boost Converter for 3D-Integrated NAND Flash SSD" (<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5759723>). 2011.
- [9] A. H. Johnston, "Space Radiation Effects in Advanced Flash Memories" (<http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/13431/1/01-2369.pdf>). NASA Electronic Parts and Packaging Program (NEPP). 2001. "... internal transistors used for the charge pump and erase/write control have much thicker oxides because of the requirement for high voltage. This causes flash devices to be considerably more sensitive to total dose damage compared to other ULSI technologies. It also implies that write and erase functions will be the first parameters to fail from total dose. ... Flash memories will work at much higher radiation levels in the read mode. ... The charge pumps that are required to generate the high voltage for erasing and writing are usually the most sensitive circuit functions, usually failing below 10 krad(SI)."
- [10] "Flash memory breakthrough could lead to even more reliable data storage." (<http://phys.org/news/2012-12-taiwan-defeat-limits-memory.html>)
- [11] "Flash memory made immortal by fiery heat." (http://www.theregister.co.uk/2012/12/03/macronix_thermal_annealing_extends_life_of_flash_memory/)
- [12] "Flash memory breakthrough could lead to even more reliable data storage." (<http://news.yahoo.com/flash-memory-breakthrough-could-lead-even-more-reliable-124049340.html/>)
- [13] TN-29-07: Small-Block vs. Large-Block NAND flash Devices (<http://download.micron.com/pdf/technotes/nand/tn2907.pdf>) Explains 512+16 and 2048+64-byte blocks
- [14] AN10860 LPC313x NAND flash data and bad block management (http://www.nxp.com/documents/application_note/AN10860.pdf) Explains 4096+128-byte blocks.
- [15] A list of ONFi members is available at <http://onfi.org/membership/>.
- [16] See pages 5–7 of Toshiba's "NAND Applications Design Guide" under External links.
- [17] 12 GB MicroSDHC (<http://www.sandisk.com/Corporate/PressRoom/PressReleases/PressRelease.aspx?ID=4079>)
- [18] 32 GB SDHC Plus (<http://www.sandisk.com/Corporate/PressRoom/PressReleases/PressRelease.aspx?ID=4091>)
- [19] http://www.pcworld.com/businesscenter/article/225370/look_out_for_the_256gb_thumb_drive_and_the_128gb_tablet.html; <http://techcrunch.com/2009/07/20/kingston-outs-the-first-256gb-flash-drive/> 20 July 2009, Kingston DataTraveler 300 is 256 GB.
- [20] Samsung Confirms 32nm Flash Problems, Working on New SSD Controller (<http://www.dailytech.com/article.aspx?newsid=16407>)
- [21] Many serial flash devices implement a *bulk read* mode and incorporate an internal address counter, so that it is trivial to configure them to transfer their entire contents to RAM on power-up. When clocked at 50 MHz, for example, a serial flash could transfer a 64 Mbit firmware image in less than two seconds.
- [22] "Intel SSD 910 vs HDD RAID in tpcc-mysql benchmark." (<http://www.mysqlperformanceblog.com/2012/09/11/intel-ssd-910-vs-hdd-raid-in-tpcc-mysql-benchmark/>)
- [23] Matsunobu, Yoshinori. "SSD Deployment Strategies for MySQL." (<http://www.slideshare.net/matsunobu/ssd-deployment-strategies-for-mysql>) *Sun Microsystems*, 15 April 2010.
- [24] Douglas Perry (2012) (<http://www.tomshardware.com/news/fusio-io-flash-ssdalloc-memory-ram,16352.html>) Princeton: Replacing RAM with Flash Can Save Massive Power.
- [25] NAND memory market rockets (<http://www.tgdaily.com/hardware-brief/71015-nand-memory-market-rockets>), April 17, 2013, Nermin Hajdarbegovic, *TG Daily*, retrieved at 18 april 2013
- [26] Humphries, Matthew (15 November 2012). "Samsung starts producing 10nm NAND memory chips" (<http://www.geek.com/articles/chips/samsung-starts-producing-10nm-nand-memory-chips-20121115/>) Geek.com.Retrieved 18 November 2012.
- [27] Clarke, Peter (20 November 2012). "Samsung takes NAND memory below 20-nm" (<http://www.eetimes.com/design/memory-design/4401791/Samsung-takes-NAND-memory-below-20-nm>) eetimes.com. Retrieved 21 December 2012.

External links

- New Pulse Measurement System For Semiconductor Device Characterization (<http://www.photonics.com/Content/ReadArticle.aspx?ArticleID=22453>)
- Semiconductor Characterization System has diverse functions (<http://news.thomasnet.com/fullstory/547012>)
- NAND Flash Applications Design Guide by Toshiba, April 2003 v. 1.0 (<http://139.138.48.19/pdf/NAND/Toshiba/NandDesignGuide.pdf.pdf>)
- Understanding and selecting higher performance NAND architectures (<http://www.eetimes.com/design/memory-design/4211387/Understanding-and-selecting-higher-performance-NAND-architectures?Ecosystem=memory-design>)
- How flash storage works presentation by David Woodhouse from Intel (<http://blip.tv/linuxconfau/how-flash-storage-works-4738604>)

Article Sources and Contributors

Flash memory *Source:* <http://en.wikipedia.org/w/index.php?oldid=571781780> *Contributors:* .:Ajvol.:., 220 of Borg, 2404, 3000farad, A13ean, Adam1213, Adamarnesen, Admiral Valdemar, Adrian Sampson, Aeluwas, Agentbla, Ahl0003, Ahoerstemeier, Ailanto, Akadewboy, Alberrosidus, Aldie, Alecv, Alex4601, AlexMS, Alexander Maksakov, Alexandria, Alexius08, Alistair1978, AlistairMcMillan, Alphaman, Alvis, Ambrosia-, An Sealgair, Anaxial, Andonic, Andrewman327, Andrewpmk, Andrewru, Andygui, Animum, Ankurdave, Antonio Lopez, AntrygG.Revok, Anwar saatad, Arjayay, Ark25, Artur Buchhorn, Arunvel007, Ashley Pomeroy, Astral, Audiomastermind, Audriusa, Auntof6, Awesome Truck Ramp, AxelBoldt, Aziz1005, Bctwriter, Ben Standeven, BenRG, Beno1000, BertK, Bhadani, BigDunc, Birge, Bitbit, Bk0, Bluemoose, Bobkart, Bog500, Bolton Lookalike, Brandon.irwin, Brianski, Brolin Empey, Bronyraur777, Bruce1ee, Bubba73, CRGreathouse, CSWarren, Callmebill, CaptSnuffy, Caramesc, Cewesley, Chaitanya.lala, CharlBarnard, Chowbok, Chris the speller, Christian List, Chroniccommand, Clayton H S, Cmayn, Colin Marquardt, Collinong, ComSpex, CommonsDelinker, Corti, Corwin8, Craig Sunderland, Crazycomputers, CrazyM108, Crispmuncher, Cyberdbo, Cybertai, Cyferz, Damian Yerrick, DanMonkey, Daniel.Cardenas, DardanAeneas, Darin-0, Davewujcik, David.Monniaux, DavidCary, Davidgumberg, Dbarnets, Dbrunner, Dcamp314, Dcjhhuang, Dcoetzee, Declan4Mac, Defladamouse, Delirium, Demixc, Den fjättrade ankan, Dennis Valeev, Dennisrd587, DerHexer, Desplesda, Destructules, Destynova, Diceman, Dickdock, Dicklyon, DigitalGuy, Digitalp, Dimawik, Discospinster, Diza, Dondervogel 2, Donhalcon, DoorsAjar, Dougher, Dragonjimmyy2k, Dspark76, Dylan.griffith, E Wing, EJSawyer, ELCleanup, EOZyo, ESKog, Eagleal, EdgeOfEpsilon, Editrevert, Edward, Elauer, Electron9, Elockard, Elsender, Emhoo, Emufarmers, EncMstr, Enkep, Erianna, Evice, Exity, Falcon8765, Favonian, Faxm0dem, Ferdinand Pienaar, Ffierling, Phaigia, FiddyCent, Flasher, FleetCommand, Flibirgit, Flyer22, Flyguy649, Foxygirltamara, Frankatca, Frap, Frappucino, Frodet, Frozen-Coke-Rocks, Fuzzform, Garas, Gatortpk, Gene Nygaard, Geopgeop, Ggrossman, Ghnrd, Giftlite, Gioto, Glenn, Glrx, God of War, Golbez, Gosgood, GraemeL, Greg L, Gregbard, GregorB, Grstain, Guiding light, Guy.H.Atkinson, Haham hanuka, Haiviet, Hardikp12, Harry491, Harryboyles, Hcobb, Hda3ku, Heron, Hibernian, Hooperbloom, Hounder4, Hulagutten, HumphreyW, Hunangarden, Hunting dog, ICE77, II MusLiM HyBRiD II, ISC PB, Ice Cold Beer, Imroy, Iqinn, Iridescent, Itai, Iterator12n, J.delanoy, JFreeman, JGrochow, JMRodrigues, JSpung, Ja 62, Jackson Peebles, Jacob Poon, Jake Wartenberg, Jaymax, Jbrisport, Jearroll, Jeeter07, Jefe2000, Jeh, Jerryseinfeld, JidGom, Jimothytrotter, Jjrepko, Joe fat porter, Joel Saks, John Duncan, JohnClarknew, Jondel, Jondelac, Jordanleegauci, JordoCo, JosephJohnCox, Jujutacular, Julesd, KDerbyshire, Kaldosh, Karebear 1022, Keilana, Kelvie, Kenny sh, Kevins, Kjkolb, Koremore, Kozuch, Kuirong Wang, LOL, Lamarrj, Lanov, Lansedebei, LapoLuchini, Larytet, Lfstevens, Lightmouse, Lingwitt, Linkerator, Lkesteloot, Looxix, Losik, Lowellian, Lozeldafan, Luigiacruz, Lysdexia, M a s, MER-C, Mac, MadGuy7023, Magus732, Mahlum, Mako098765, Mange01, Mangojuice, Marcok, Marek69, Markwads, MartinPackerIBM, Materialscientist, Matt Britt, Matthiaspaul, Matusz, Maury Markowitz, Mearling, Mckoss, Mclean007, Mditto, Mdwh, Meersan, Memorytn, Michael Broquist, Michael Hardy, Michal.Pohorelsky, Microdialtd, Mikeblas, Miltonhowe, Mindmatrix, Minimac, Mix Bouda-Lycaon, Moe Epsilon, Moog10000, Morcheeba, Mortense, Moxfyre, MrBurns, Mrcaseyj, Music Sorter, Mwaisberg, Mwarren us, Nakon, Nandexpert, Nehrams2020, Neutrality, NightMonkey, Nixdorf, Norm mit, Norvy, Novacatz, Nrbelex, Nsaa, Numbermaniac, Numbo3, OccamzRazor, Oda Mari, Odie5533, Ohconfucius, Oli Filth, Olivier, Omicronperse18, Onco p53, Oneiros, Oni Lukos, OwenX, Oğuz Ergin, Panu-Kristian Poiksalo, Paste, Patrick, Patrick.franklin, Pedro, Pekaje, Perry Bebbington, Peter1978, Peyre, PhotoBox, PierreCA22, Pietdesomere, Plugwash, Pnm, Postlewaight, Pot, Potatoswatter, Preslethe, Prince of Strings, PsyberS, Pxma, Pyromaniac4382, Pádraic MacUídhir, Qdr, Quibik, R. A. Wilson, RCHenningsgard, Racexr11, Rainer Wasserfuhr, RastaKins, Ravedave, Rchandra, RedWolf, Reelrt, Retroneo, Richi, Rigby27, Rjwilmsi, Rmallins, Rockfang, RodC, RonaldDuncan, Ronaldo Guevara, Russpav, Russvdw, Rwalker, ST47, Saberwyn, Salamurai, Salivum, Salsa Shark, Sander123, Seidenstud, Semicolons, Sfoskett, Shaddack, SharkD, Sharpaaa, Shentino, Shreshth91, SimonLyall, Simonckenyon, SI, Smack, Smithbrenon, Smt52, Sneha.chavan, Sonicyouth86, Spidern, Spur, StableKeep, Stefrance690, Stephan Leeds, StephenFalken, StevenDH, Stevenmitchell, StuartBrady, Subversive.sound, Suffusion of Yellow, Surfer43, Taz69x, Tankiona, Tbfly, TechWriterNJ, Technopat, Technopilgrim, Tek022, TestSubject528491, Thaninj, The Random Editor, The Thing That Should Not Be, TheGiantHogweed, TheImitation, Thingg, Thue, Thunderbird2, Tobias Bergemann, Tom5760, Transisto, Tree Biting Conspiracy, Turian, TuukkaH, Tvancourt, Tysonclugg, Urod, Vafa95, Vbishtei, Vegaswikian, Vickypathak1990, Vlado4, VncSG, Vodomar, Vrenator, Wadahh77, Waloddyca, Wapcaplet, Wbm1058, Wereon, Wernher, Willsmith, Winston Chuen-Shih Yang, Winterst, Wjw0111, Wmahan, Woohookitty, Work permit, Writtenonsand, Wtshymanski, Yankeguy, Yankohipara, Yialanliu, Yuzhip, ZeWrestler, ZeeFBI, Zeno Gantner, Zephyris, Zerpent, Zillog Jones, Zodon, ZoeB, ^musaz, தருமரெட்டி, 1043 anonymous edits

Image Sources, Licenses and Contributors

Image:USB flash drive.JPG *Source:* http://en.wikipedia.org/w/index.php?title=File:USB_flash_drive.JPG *License:* GNU Free Documentation License *Contributors:* User:Nrbelex

Image:Flash cell structure.svg *Source:* http://en.wikipedia.org/w/index.php?title=File:Flash_cell_structure.svg *License:* GNU Free Documentation License *Contributors:* Cyferz

Image:NOR flash layout.svg *Source:* http://en.wikipedia.org/w/index.php?title=File:NOR_flash_layout.svg *License:* GNU Free Documentation License *Contributors:* Cyferz (talk)

Image:Flash-Programming.svg *Source:* <http://en.wikipedia.org/w/index.php?title=File:Flash-Programming.svg> *License:* Public Domain *Contributors:* David W.

Image:Flash erase.svg *Source:* http://en.wikipedia.org/w/index.php?title=File:Flash_erase.svg *License:* Public Domain *Contributors:* David W.

Image:Nand flash structure.svg *Source:* http://en.wikipedia.org/w/index.php?title=File:Nand_flash_structure.svg *License:* GNU Free Documentation License *Contributors:* Cyferz

Image:NAND scaling timeline.png *Source:* http://en.wikipedia.org/w/index.php?title=File:NAND_scaling_timeline.png *License:* Creative Commons Attribution-Sharealike 3.0,2.5,2.0,1.0 *Contributors:* Cepheiden, Magog the Ogre, 1 anonymous edits

License