



NAND Flash FAQ

- What is NAND Flash?**
- What is the major difference between NAND Flash and other Memory?**
- Structural differences between NAND Flash and NOR Flash**
- What does NAND Flash controller do?**
- How to send command to NAND Flash?**
- How data are stored in NAND Flash?**
- How to program a NAND Flash device?**
- How to read data from NAND Flash device?**
- What are the commands supported by NAND Flash devices?**
- What is MLC and SLC?**
- What is ONFI?**
- Large block vs. small block?**
- What is spare column in NAND Flash?**
- What is Error Correction Code (ECC)?**
- What type of hardware ECC to use?**
- What is bad block?**
- What is wear leveling?**
- What is RAM shadowing?**
- Can NAND Flash used as boot rom?**
- What is eMMC and eSD?**

What is NAND Flash?

NAND Flash is a type of memory device called nonvolatile memory. It is by far the most common nonvolatile memory used for mass storage. The following table classifies different types of silicon memories:

Type	Sub-type	Example
Volatile Retain data only when power is on. Loss data when power is off	Static memory Retain data indefinitely as long as power is on. Consumes none or very little power to retain data	SRAM such as CPU cache
	Dynamic memory Retain data for a small period of time when power is on. Require periodic refresh to retain data. Consumes power during refresh	SDRAM



Type	Sub-type	Example
Nonvolatile Retains data regardless of power supply	Programmable memory Data can be written into the device many times.	NAND Flash NOR Flash
	One-time programmable memory Data can be written into the device only during the manufacturing process.	Mask programmable ROM

What is the major difference between NAND Flash and other Memory?

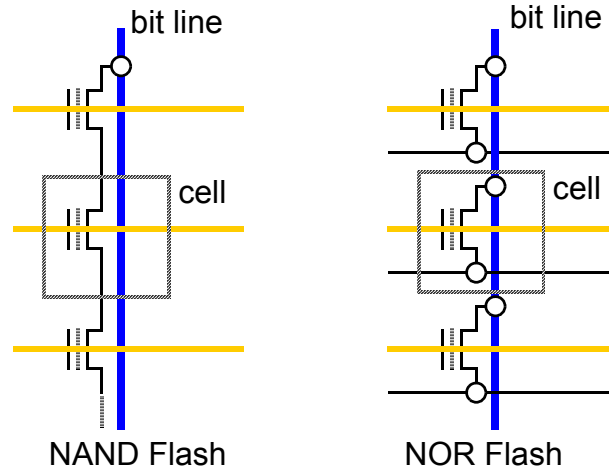
From the system designer’s perspective, the biggest difference is that NAND Flash is a serial storage device while most other memories are random access memory (RAM). Random access device can be designed easily as the primary storage (main memory) of a system. Typical instruction and data fetch from the CPU involved an address phase and data phase on the CPU bus. Random access device can retrieve the required data easily. Subsequently and different address locations can also be accessed with little penalty. To the contrary, serial storage device requires long access time for the initial data and subsequent access to any nonconsecutive location also requires large penalty. As a result, serial storage device such as NAND Flash requires special NAND Flash controller to access data and is seldom used as the main memory of the system.

Structural differences between NAND Flash and NOR Flash?

The biggest differences between NAND Flash and NOR Flash devices is way the memory cells are arranged. Both NAND and NOR Flash cells uses single transistor memory cell. In NOR Flash, these cells are arranged in parallel with all the source node of the cells connected to the bit line, similar to the way that NMOS transistors are arranged in building a NOR gate. In NAND Flash, the cells are arranged in series with the adjacent cells sharing source and drain, similar to the way NMOS transistors are arranged in building a NAND gate. The sharing of the source and drain of adjacent cells eliminated the need for metal contact and tremendously reduces the die size. NAND Flash cells can be



packed much closer together, with a 60% saving cell size over NOR Flash.



What does NAND Flash controller do?

The serial cell structure used in NAND Flash allows very high storage density but data from the memory block can only be read serially. The disadvantage, as compared to RAM, is that data cannot be randomly accessed. But once a page of memory is opened for read, data can be shifted out from the memory quickly. The NAND Flash interface also requires that commands to the NAND Flash be sent serially to the device as a command packet, instead of the parallel "address" and "data" signals in typical RAM. These are the major reason that make interfacing with NAND Flash memory much more complicated than interfacing with typical SRAM or NOR Flash devices. A NAND Flash controller is design specifically to handle all required tasks of accessing NAND Flash device efficiently.

How to send command to NAND Flash?

NAND Flash devices has 5 control signals (CS, ALE, CLE, RD, WE), 8 or 16 data signals and one response signal (R/B). Commands are shifted in to the device through the data pins when CLE is active. If the command contains address, the address information are shifted into the device through the data signal when ALE is active. RD and WE are used to shift data into or out of the NAND Flash device. For command that requires long time to process, such as erase, or program, the R/B signal indicates when the device is busy.



How data are stored in NAND Flash?

The basic unit of operation for NAND Flash device is one page of data with some commands affecting the whole block (multiple pages) or the whole chip. Before programming, a page should be erased which sets all data bits to "1". Afterwards only the value "0" can be programmed into each cell. Programming effectively stores a new data with the value equal to the logical "AND" of the existing data and the program data.

How to program a NAND Flash device?

NAND Flash devices are programmed on a page by page basis. The command word and the page address are first shifted into the device followed by the programming data. The exact programming steps of a NAND Flash device varies depending on whether it is large block or small block device and whether features such as two plane programming or cache mode are used. During programming, the R/B signal is low to indicate that the device is busy. Typical programming time is a few hundred micro second per page. Variations such as two plane programming, cache programming and random programming are supported by some NAND Flash devices.

How to read data from a NAND Flash device?

NAND Flash devices are read by shifting in the command and address. After the command and address are shifted in, it takes a few tens of micro second to open a page. After a page is opened, data can be shifted out of the device by using the RD command signal. Some devices support random data read within a page and some supports cache read mode.

What are the commands supported by NAND Flash devices?

There are many commands, some manufacture specific and supported only by a few devices while some commands are universal to all NAND Flash manufacturers. The most common commands are RESET, ERASE, PROGRAM, PROGRAM CONFIRMATION, READ DATA, READ STATUS and READ ID. Device specific commands include RANDOM READ, RANDOM WRITE, PAGE CACHE READ, PAGE CACHE WRITE, INTERNAL DATA MOVE, TWO-PLANES READ, TWO-PLANE WRITE, and others.

What is MLC and SLC?

Single Level Cell (SLC) NAND Flash stores only one bit of information per memory cell while Multi Level Cell (MLC) NAND Flash stores multiple bits of information per memory cell. MLC offers higher storage density but in general is slower and less robust. SLC can be used for up to 100,000 erase/program cycles while MLC allows about 10,000 erase/program cycles. Most manufacturer also require 4-bit error correction



(4 bits out of each 512 byte of data) on MLC while only 1-bit error correction is required for SLC. When design with NAND Flash, it is important to know whether MLC or SLC is used so that the appropriate error correction scheme is used.

What is ONFI?

ONFI stands for Open Nand Flash Interface. It is promoted by an industrial consortium as the open standard for device level NAND Flash interface. Early NAND Flash devices from different manufacturers use similar interface but an open standard did not exist. As a result, subtle differences exist among devices from different vendors. ONFI standard aims to provide a common standard so different devices can be used interchangeably and sets the stage for future standard NAND Flash development.

Large block vs. small block?

NAND Flash devices can also be categorized by large block and small block devices. From the user's perspective, it is more correct to call them as large page and small page devices. Large block device typically has 2048 bytes of data and 64 bytes of spare data per page while small block device has 512 bytes of data and 16 bytes of spare data per page. Commands sequence for large block and small block devices are different so the controller must be aware of which type of device is being used.

What is spare column in NAND Flash?

NAND Flash devices organize 512bytes or 2048 bytes of data into a page. There are also 16 or 64 bytes of extra data called the spare column associated with each page. The spare columns are fully addressable by the user and is typically used for storing Error Correction Code (ECC) and other management information to improve data integrity.

What is Error Correction Code (ECC)?

Due to manufacturing issues (yield), usage (wear and tear) and environmental factors, data stored in NAND Flash may not retain its data value as written. However, the probability for this happening is very small so normally only very few number of bits contain the wrong value and the vast majority of the data bits is still correct. ECC is a good way to recover the wrong value from the remaining good data bits. NAND Flash manufacturers recommend the use of ECC for most NAND Flash applications.

ECC can be implemented by software or hardware. If it is implemented by software, instead of storing raw data into the NAND Flash, software pre-process data to embed ECC information into data before writing



into NAND Flash. More CPU power is needed in this method but no special hardware is needed.

If ECC is implemented by hardware, the NAND flash controller generates ECC code when each byte of data is written through the controller. ECC information is stored in the spare column area by the controller on write access and retrieved from the spare column area in read access.

What type of hardware ECC to use?

For 1-bit ECC correction per 512 bytes (as required in SLC), 24 ECC bits are required for each 512 bytes of data. Typically Hamming code is used for 1-bit correction and the ECC bits are generated by hardware and stored in the spare column area so it does not consume the normal storage area. For 4-bit ECC correction required by MLC, "Bose, Chaudhuri, and Hocquenghem" (BCH) code is used. 52 ECC bits are required. Again the ECC bits are stored in the spare column area. For 8-bit ECC correction, 104-bit BCH code is used.

A different ECC scheme, ECC protection per word, is also possible. It requires 7 ECC bits to be stored along with each 32-bits of data. This allows the ECC circuit to correct 1 bad bit out of the 39 bits (32-bit data + 7-bit ECC) and detect two bad bits out of the 39 bits. This ECC scheme requires 25% overhead of memory storage and requires an extra NAND Flash device dedicated for ECC storage. This scheme is not as commonly used as ECC protection per page but is suitable for certain applications.

The EP501 NAND Flash controller from Eureka Technology supports all these types of ECC.

What is bad block?

Like all large memory devices, not all the memory cells are fully functional due to yield related issues. Most memory devices use some kind of repair method to repair or remap the bad cells so that the memory device is fully functional from the user's perspective. NAND Flash devices handles this problem by declaring some blocks (a block is 64 pages of memory) as bad block. During production testing after wafer fabrication, each memory die is tested and bad blocks, if any, are marked at the spare column of the first 2 pages of the block. It is the responsibility of the user to read the bad block markings and to avoid using the bad blocks. Over the life time of the NAND Flash, additional bad blocks may develop due to repeated use of the memory cells so the user must be equipped to handle the new bad blocks. Typically bad block management is done by system software to remap data to avoid using bad blocks in the memory device.



What is wear leveling?

NAND Flash cells, regardless how carefully they are designed and tested, have only limited lifetime. Typically, NAND Flash cell can be erased or programmed only for limited times (100,000 times for SLC and 10,000 times for MLC) before it fails. If a particular page of data stored in NAND Flash is updated often, cells within that page can fail after very short time, rendering the entire system to fail. Wear-leveling is a technique that spreads the memory use evenly to different physical pages so that the entire NAND Flash devices is used evenly to maximize the life span of the device and the system. Similar to bad block management, wear leveling involve the remapping of logical address to different physical address of the device. Wear leveling and bad block management are typically done by software called Flash Transaction Layer (FTL). Many software vendors supplies FTL software.

What is RAM shadowing?

Because of the limited erase/program cycle and slow speed (compared to SDRAM) of NAND Flash devices, shadowing is a technique that system designers use to increase performance of the system. Portions of the NAND Flash device or the entire device is copied (shadowed) to SDRAM or SRAM during system initialization. Once copied, the system operates directly out of the SDRAM/SRAM. Data are copied back to the NAND Flash device only when nonvolatile storage is needed.

Can NAND Flash used as boot rom?

Yes, a well designed NAND Flash controller should support this function. There are two basic methods to handle this function. The simpler method requires the controller to open a page for read immediately after reset. A special boot up code is written to the first page so that the CPU can fetch all the initial instructions within this page. The CPU may "think" it is doing initial code fetch from different locations as needed but all the required data is actually provided sequentially from the first page of the NAND Flash. The initial code must contain instructions for the CPU to access the remaining pages from the NAND Flash or to shadow the NAND Flash to SDRAM/SRAM.

The second method requires the NAND Flash controller to automatically shadow the initial page(s) from the NAND Flash to the main memory of the system or to some random access memory immediately after reset. CPU would then boot from the random access memory.

What is eMMC of eSD?

eMMC or eSD are NAND Flash based storage chip that features eMMC or eSD interface instead of the typical NAND Flash or ONFI interface. MultiMedia Card (MMC) and Secure Digital (SD) are well established interfaces that are used in removable memory cards. eMMC and eSD can be



removable or board mounted devices. To interface with these device, an SD host controller such as the EP550 SD IP core is needed instead of a NAND Flash controller.