



Eureka Technology

Fundamentals of NAND Flash Device Usage

by

Eureka Technology Inc.

August 26th, 2010

Copyright (C)

All Rights Reserved



Introduction

This white paper presents very important information for managers, engineers, and system architects who are working with NAND Flash memory. NAND Flash memory devices have very different operating characteristics from other memory devices. One must understand these characteristics in order to harness the power of this technology and deploy it wisely into new designs and applications.

What is NAND Flash?

NAND Flash is a type of nonvolatile memory device. It is the most commonly used nonvolatile memory for mass storage today. The following table classifies different types of silicon memory technologies:

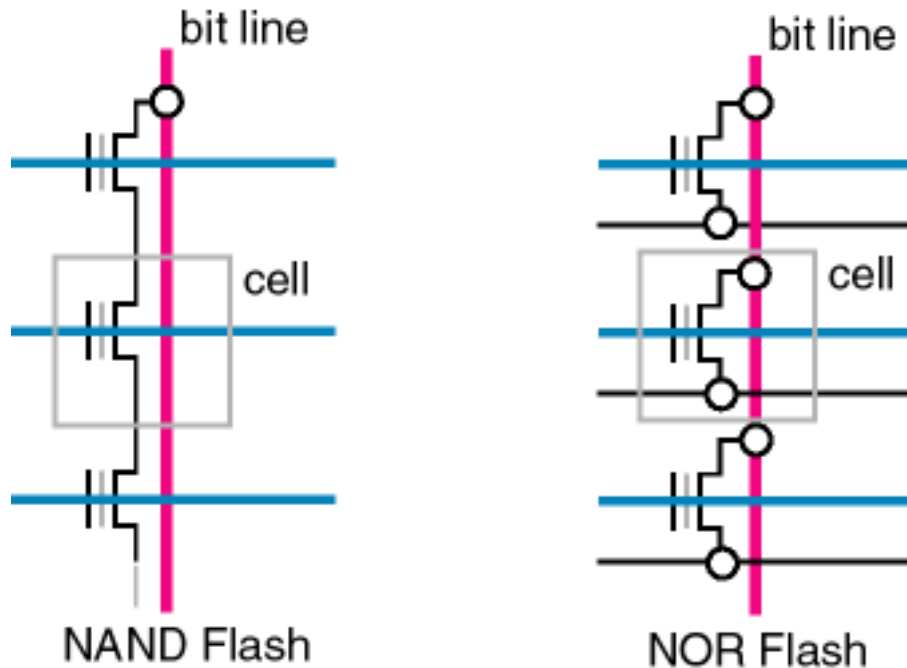
Type	Subtype	Example
Volatile Retain data only when power is applied. Loss data when power is removed	Static memory Retain data indefinitely as long as power is applied. Typically consumes very little or no power to retain data.	SRAM, such as cache memory in CPU
	Dynamic memory Retain data for limited time when power is on. Require periodic refresh to retain data. Consumes power during refresh	SDRAM, such as main memory used in personal computers.
Nonvolatile Retains data even when no power is removed	Programmable memory Data can be written into the device many times.	NAND Flash NOR Flash CMOS NVM
	One-time programmable memory Data can be written into the device only during the manufacturing process.	Mask programmable ROM



Structural differences between NAND Flash and NOR Flash

The biggest difference between NAND Flash and NOR Flash devices is the way that the memory cells are arranged. Both NAND and NOR Flash cells use 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 eliminates 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.

NAND and NOR Flash Structure



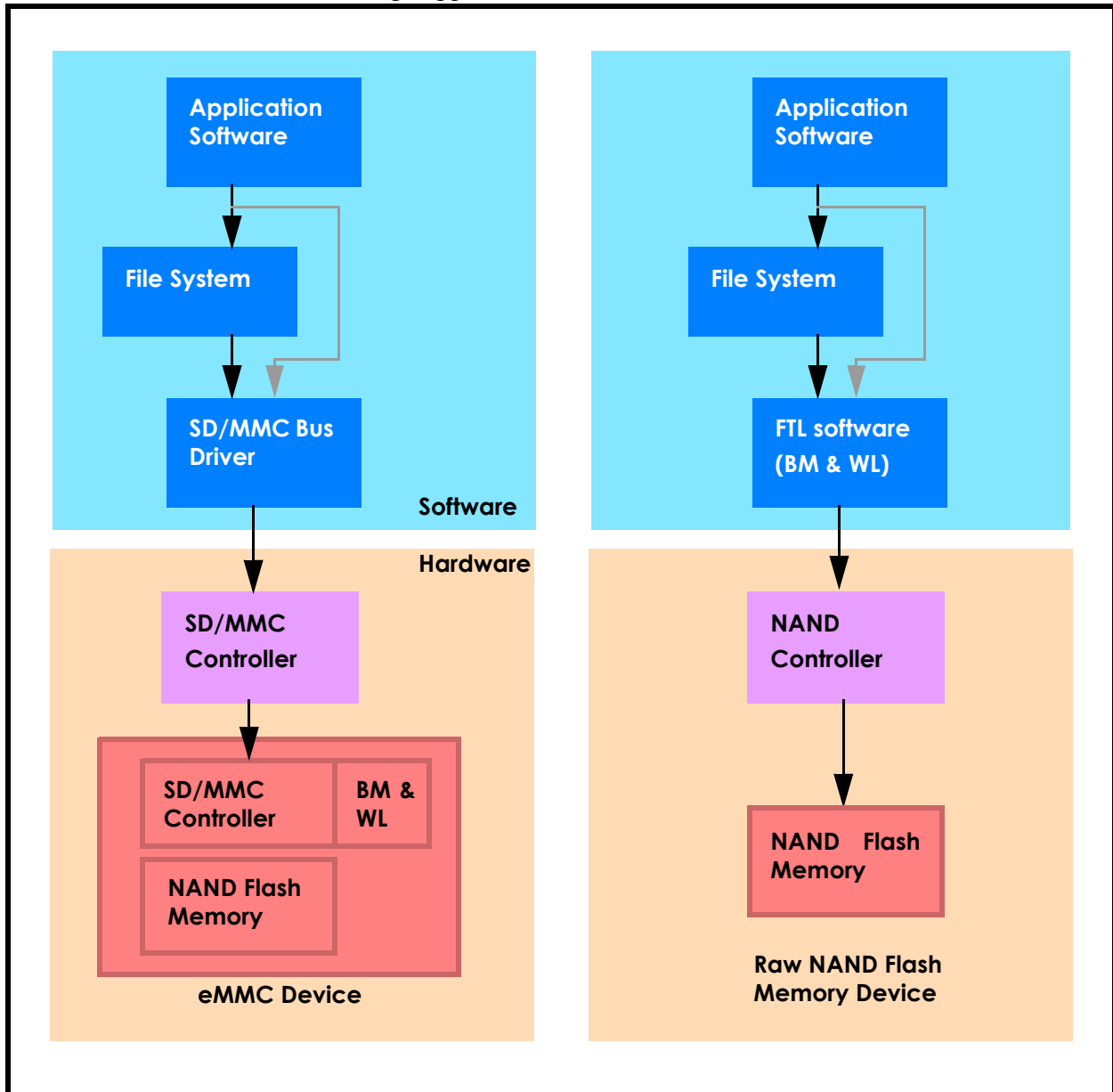
NAND Flash vs. SD memory vs. MMC

NAND Flash is the fundamental nonvolatile storage technology and raw NAND Flash device typically comes with the serial interface such as ONFI (more about that later) or similar interface. NAND Flash device can be found as discrete memory chips, or packaged with a controller to form SD memory card, MultiMedia Card (MMC) or solid state disk (SSD). SD and MMC card are widely used in consumer applications such as digital camera. SSD is designed to replace hard disk drive in computing systems and features SATA or similar interface. For embedded application where memory storage other than hard disk-like device is required, designer has a choice of using raw NAND Flash device or SD or MMC memory for nonvolatile storage. Many NAND Flash manufacturers offer single chip device that integrate both the MMC controller function with



NAND Flash memory array. This integrated solution is called the eMMC standard (controlled by JEDEC) and marketed under different names by different manufacturers.

If raw NAND Flash chip is used, the designer is responsible for all NAND Flash management tasks such as wear leveling, ECC, and bad block management. If eMMC device is used, these management tasks are off-loaded to the eMMC device. The following diagram compares the system architecture of these two design approaches.



Using eMMC device results in simpler hardware and software design but raw NAND Flash offers more flexibility, higher performance and lower cost.



SLC vs. MLC

NAND Flash memory uses one of the two memory technologies: Single Level Cell (SLC) or Multi Level Cell (MLC). As the name implies, 1 bit of information is stored in SLC and 2 or more bits of information can be stored in MLC. MLC offers higher memory density which results in lower cost per bit of storage. However, because multiple levels of charge storage is written into each memory cell, noise immunity and data integrity of MLC is not as good as SLC. SLC also offers higher performance with faster page open time and shorter programming time. Compare to MLC, SLC memory can be re-written (write endurance) many times more than MLC devices. SLC can offer write endurance of over 100,000 cycles but MLC are typically rated at 5,000 to 10,000 cycles.

To combat soft and hard errors in NAND Flash memory cells, error correction code (ECC) is required in both SLC and MLC devices. Because of lower data integrity, MLC devices requires more sophisticated ECC than SLC device. The added complexity of ECC increases the gate count of the controller for MLC device.

Other than the differences in ECC requirement and write endurance cycle, SLC and MLC functions the same and typically employs the same external interface. The same higher level design such as application software, device driver and the controller hardware are independent of SLC and MLC devices.

NAND Flash device interface

Depending on the type of NAND Flash devices, external interface to the NAND Flash device can be synchronous (synchronized to the clock signal) or asynchronous (through the use of read and write pulses). In both methods, NAND Flash device makes use of 8 data signals and a few command signals for address, command, and data communication. Commands and address information are shifted into the device through the data pins during the command and address phases.

NAND Flash data are organized into pages in the NAND Flash device. A two step process is required in order to read data from NAND Flash. First, the page must be opened (as a result of the read command) for read. Page open time range from 20 to 50 us, translated into thousands of cycles for typical operating frequency. After a page is opened, all the data from the page is transferred internally to shift registers inside the NAND Flash, ready to be transferred to the external interface. Data can then be shifted out sequentially from the NAND device one byte at a time. Data transfer typically takes 20 to 100 ns per byte for asynchronous device. Synchronous device are much faster with 5 to 25ns transfer time per byte.

Data are read sequentially from within a page. To support random access, NAND Flash device allows user to change the read address within the page once the page is opened. However, the ECC requirement and the extra time needed for shifting the new address make random access very inefficient and rarely used.

To store data into NAND flash, a similar (but reversed) two step process is required. Data is first shifted into the shift registers of the NAND Flash device. After all the data of the page has shifted in, the user issues a program command to direct all the data from the data registers to be pro-



grammed into the specific page. Typical programming time is a few hundred microseconds per page for SLC and a few miniseconds for MLC. Variations such as two plane programming, cache programming and random programming are supported by many NAND Flash devices to reduce programming time and provide flexibility.

Before new data can be written to a page, the page should be erased which sets all data bits to "1". During programming, either "1" or "0" can be written into each cell. After a "0" has been written to a cell, it cannot be changed back to "1" again with another write. Only the erase command can change the cell back to "1". In other words, the data stored in each cell is always the logical "AND" of the new (write data) and the existing data.

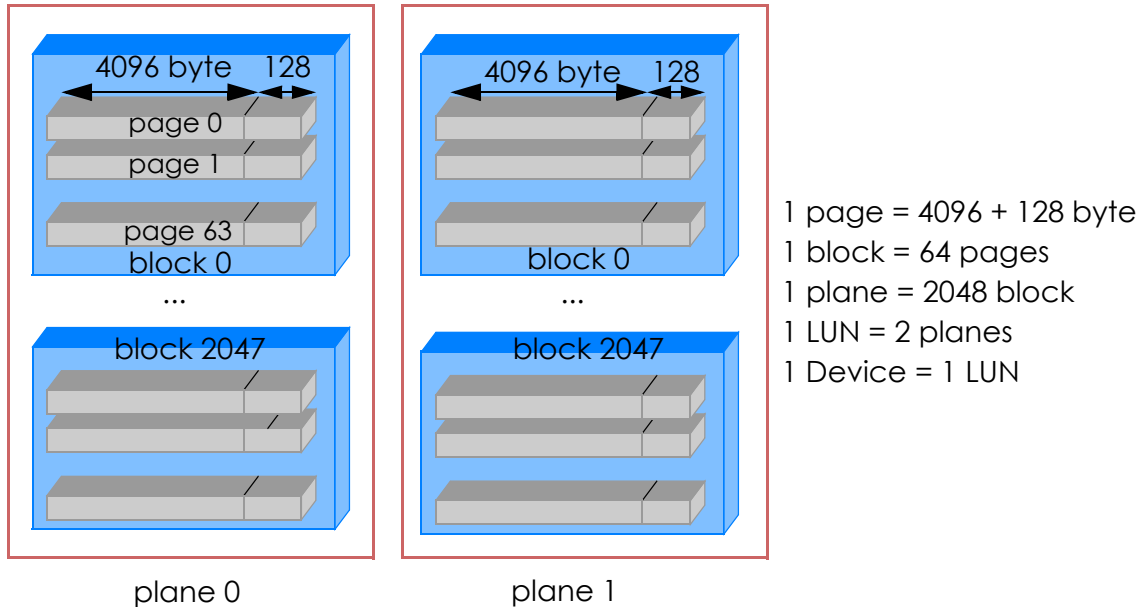
NAND Flash data organization

The basic unit of operation for NAND Flash device is a page of data. A page is typically 512 bytes to 16 kbytes, depending on the device. Each page also contains additional bits called the spare data or spare column. The spare data size may be as little as 16 bytes per page for page size of 512 bytes, or up to a thousand bytes per page for page size of 16 kbytes. The spare data is part of the page and is accessed at the same way as the main data page. The spare data can be used for marking in case a page or a block is bad and for storing ECC parity bits.

Multiple pages are grouped together to form a block and multiple blocks are grouped together to form a plane. A Logical Unit (LUN) contains one or two planes. If the LUN contains 2 planes, interleaving read or write between the two plane is supported. A NAND flash die contains one or multiple LUN. Typical NAND Flash device may contain one or multiple NAND Flash dies.



Some commands such as read and program affects only one page while other commands such as erase affect the whole block. Commands such as reset affects the entire LUN. The following diagram shows a typical organization of a 8Gbit NAND device.



ECC implementation

Data stored in NAND Flash device need to be protected by ECC in order to achieve the write endurance cycle as guaranteed by the manufacturer. The ECC requirement ranges from 1-bit correction per 512 bytes of data to 40-bit correction per 1 kbyte of data. As memory density increases, the charge held by each cell decreases, and ECC requirement increases.

When data is written to each page, the ECC for the page is computed and stored into the spare column of the page. When data is read from the page, both the data page and spare column are read. ECC is then computed and error data is corrected. As a result, the entire page of data has to be read or written in order to compute ECC. Data transfer on NAND Flash is page oriented and random access within the page or partial page access is not commonly used with NAND Flash.

Many types of ECC, such as Hamming code, Reed Solomon, BCH, and LDPC can be used to provide the required ECC protection. The most commonly used ECC scheme is BCH code, which is named after its inventors, Bose, Chaudhuri and Hocquenghem. Discovered in 1959, it is a form of cyclic redundant code that is widely used in many applications in addition to NAND Flash. For a given data size, the BCH code parity bits increases linearly in proportion to the number of error protection requirement. As a result, the spare column size in NAND Flash device also increases as the number of ECC bit protection increases. BCH code is typically implemented in hardware or a combination of hardware and software. A pure hardware implementation requires more gates and combination of hardware and software may provide the proper balance.



What does NAND Flash controller do?

Due to the serial cell structure used in NAND Flash devices, data from the memory block are read serially. The disadvantage, as compared to NOR Flash or SRAM design, is that data cannot be randomly accessed. However, once a page of memory is opened for read, data can be shifted out from the memory much faster than in NOR Flash. 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 SRAM.

NAND Flash device requires ECC protection for stored data. Typically the ECC generation and detection circuits are implemented in the NAND Flash controller. For devices that requires 8 or more bits of protection, the total logic gate count of the controller may be dominated by the ECC circuit.

DMA function is also found in many controller design. Due to the long page open and program time of the device, using DMA controller to access NAND Flash is a common technique to free up the CPU to handle other important system tasks.

These are the major reasons that make interfacing with NAND Flash memory more complicated than interfacing with typical SRAM or NOR Flash devices. Special controller designed for NAND Flash interface is required to communicate with NAND Flash devices. The controller serves as a bridge between the CPU and the NAND Flash device. A well designed controller delivers the maximum bandwidth of the NAND Flash device while a poorly designed controller not only reduces system performance but also puts a lot of burden on the software design. Designer and system architect must understand the importance of the NAND Flash controller and its impact on overall system performance. Care must be taken to select the suitable controller for each application.

Synchronous and Asynchronous NAND

Traditional NAND Flash memory uses asynchronous interface to transfer data externally. The interface method does not use clock signal. Similar to asynchronous SRAM device, asynchronous NAND Flash uses the rising edge of the read and write signals to shift out and shift in data. Even though simple to use, asynchronous interface is slow and the full bandwidth is very difficult to realize. The fastest asynchronous NAND Flash interface has a transfer time of 25ns. Typically, there is a requirement of 12ns read pulse low time and 12ns of read pulse high time. However, nearly all controller logic design are synchronous to clock input and pulse width are generated as multiple of clock cycles instead of nanosecond. If the controller operates at 100Mz (10 ns cycle time), it impossible to generate a pulse width of 12ns without using expensive analog circuit. Digital circuit with this clock rate can only generate pulse width which is multiple of the cycle time (10ns). In this case, it would generate pulse with 20ns high time and 20ns low time, resulting in 40ns cycle time and 37% reduction from the maximum bandwidth. At 25ns fastest access time, the maximum bandwidth of asynchronous NAND Flash interface is 40Mbit/sec per pin.

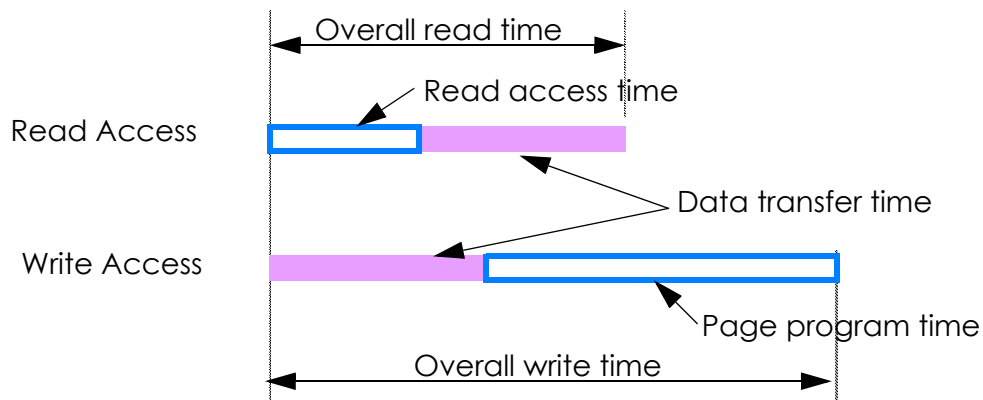


Synchronous NAND Flash utilizes the clock signal to transfer data. Double Data Rate (DDR) interface is also used to transfer 2 bits of data per clock cycle. Synchronous NAND Flash operating at 100Mhz is capable of transferring 200Mbit/sec per pin.

Several manufacturers also offer asynchronous NAND Flash with DDR mode to bridge the gap in bandwidth performance. In DDR asynchronous device, data is transferred both at the falling and rising edge of the read and write signals, effectively doubling the data rate over traditional asynchronous device but still fall short of the synchronous device.

Data bandwidth and interleave access

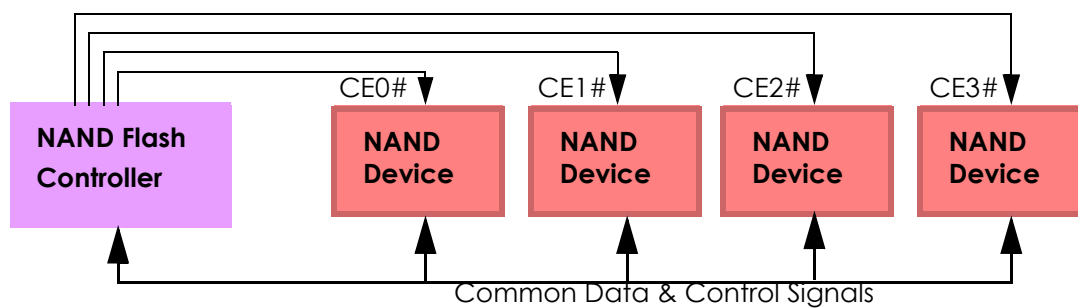
Data bandwidth of a NAND Flash memory system is dominated by two factors: (1) data transfer time, and (2) page access time. The discussion about synchronous and asynchronous NAND Flash in previous section highlights only the data transfer time. Data transfer time determines the performance of the NAND device when it is actively transferring data. Page access time determine how fast the device can start transferring data once a command is received. It plays as important a role in determining the overall performance as data transfer time. For read access, the page access time is the time between the read command and the time read data can be shifted out. Read access time is about 20us to 50us for most devices. For write access, the access time is the page program time. It is defined as the time between a program command and the time new data of the next page can be shifted in. It is several hundred microseconds for SLC and several milliseconds for MLC. Page program time is much longer than data transfer time.



To reduce the overall access time and to improve performance, interleaving technique is used. Most NAND Flash device that features two planes supports interleave read and interleave write. In interleave read, the NAND Flash device allows a page in the second plane to be opened while data of the a page in the first plane is being transferred. In interleave write, the NAND Flash device allows two pages from opposite planes to be programmed at the same time. With this method, there is only one program time overhead for two pages of data, effective reduces the program time by half.



In addition to interleave write supported internally by the NAND device, external interleave can also be employed to further improve performance. In this scheme, multiple NAND Flash devices are connected to the same data and control signals of the controller. Each device connection differs only in the chip enable (CE#) signal. When one NAND Flash device is in the program state, the controller can select a second device and transfer data to the second device. The same overlapping access can be extended to the third and fourth devices sharing the same data pins. The programming time of one chip can overlap with the data transfer and programming time of another chip, allow the data transfer from the controller to be continuous with all the programming time hidden.



What is bad block?

Like most 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 as bad block. During production testing, 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 have only limited lifetime. Typically, SLC NAND Flash cells can be erased or programmed only about 100,000 times before they fail. MLC cells may support only 5,000 to 10,000 erase/program cycle. If a particular page of data stored in NAND Flash is updated often, cells within that page can become useless after a very short time, rendering the entire system to fail. Wear-leveling is a technique that remaps the same logic address to different physical pages



each time it is used. The goal is to use the entire NAND Flash devices evenly to maximize the life span of the device and the system. Wear leveling and bad block management are typically done by software called Flash Transaction Layer (FTL). Many software vendors supply FTL software. The operating system and application software does not access the NAND Flash directly. Access is made through the FTL software.

What is RAM shadowing?

Due to 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 it is necessary.

Using NAND Flash as boot ROM

When using NAND Flash as the boot device, some special design considerations are needed. Since NAND Flash data cannot be read until a page is opened, it cannot be used for CPU code fetch during boot up time. Special hardware must be built into the NAND Flash controller to open a page for read immediately after reset. A special boot up code must be written to a designated page so that the CPU can fetch all the initial instructions within this page. The initial code is usually the boot loader which contain instructions for the CPU to access the remaining pages from the NAND Flash or to shadow the NAND Flash to SDRAM/SRAM.

DMA controller can also be used to transfer the boot code from NAND Flash to random access memory such as SDRAM or SRAM. After the transfer is completed, the CPU can start execution from the SDRAM or SRAM. This method requires a DMA controller which direct access to NAND Flash controller.

The ONFI standard

Formed in 2006, Open NAND Flash Interface (ONFI) Working Group is an consortium of NAND Flash technology companies who seek to create an industrial standard for NAND Flash device interface. The ONFI standards encompasses device pinout, device function, command set, timing, electrical specifications and others. The ONFI consortium includes many major NAND Flash manufacturers with the exception of Samsung and Toshiba.

Historically, different NAND Flash manufacturers produced NAND Flash devices that were very similar in pinout and functionality to each other. Even without a common, industry-wide standard, NAND Flash devices from different manufacturers can often be used interchangeably. As new features are added by each manufacturer, however, the need for a common standard becomes apparent. When the ONFI standard was first published, it formalize the *de facto* standard then in use. Significant changes to NAND Flash interface such as synchronous and DDR interface was



introduced in later revisions of the standard. The following table list the major changes in different ONFI revisions:

ONFI Revisions	Important Features
ONFI 1.0	Asynchronous interface only, maximum 50Mbps data rate.
ONFI 2.2	Add synchronous interface, maximum data rate is 200Mbps
ONFI 2.3	Built-in ECC.
ONFI 3.0	Maximum synchronous transfer rate is 400Mbps, asynchronous DDR transfer rate is 200Mbps, differential signal, on-die termination,

In order to keep up with the progress of technology, ONFI is designed to be very flexible so it will not be out grown by technology changes. For example, instead of limited the page size of a NAND Flash device to a few commonly used sizes, the standard allows the page size to be any number of bytes which is power of 2. While this and other flexibility in the standard ensure its longevity, the vast range of parameters limits the interoperability of different devices. For example, it is not very practical to design a controller (and ECC circuit) to cover page sizes ranging from 512 bytes to, say, 2 megabytes. Designer must keep in mind that the ONFI standard does not guarantee that an ONFI compatible controller design will be able to operate with all ONFI compatible NAND Flash devices. A controller and hardware system design should be targeted to a limited set of NAND Flash devices with a certain size and organization range instead of trying to cover all possible combinations.

Eureka Technology Inc.
4962 El Camino Real
Los Altos, CA 94022, USA

Tel: 1 650 960 3800
Fax: 1 650 960 3805
email: info@eurekatech.com
<http://www.eurekatech.com>