CMP United Business Media
ELECTRONIC ENGINEERING
EETIMES europe
For the creators of technology
www.eetimes.eu

DESIGN CURRENTS

Controllers in a flash – solid state discovered

By Axel Mehnert

Flash memory has significantly replaced various historic storage media. As examples, some 5 years back people still used film in non-digital cameras, computers were offering floppy disk drives, mobile CD players played your favorite tunes, and mobile phones were incapable of taking pictures, browsing the web or synchronizing with a PC.

In recent years, consumers have become accustomed using a large variety of NAND flash memory devices for data storage applications. Users of mobile phones, digital cameras, and other gadgets are familiar with the terms SD card, USB-stick, CF cards and the like. Such kind of small and handy memory systems have become commodities among consumers.

Requirements for such consumer applications are normally determined by

- Cost
- Compatibility to a plethora of host devices
- Fast transfer rates for reading and writing of data
- Storage Capacity
- Conformity to standards

Memory cards or USB sticks that fulfill these requirements can meanwhile be found in almost any supermarket or grocery store around the world. As the above features are not too difficult to obtain, it has become a challenge to list the variety of vendors who manufacture or sell such memory systems.

Have you ever experienced the sharp sound of your hard disk when it crashes? If so, you may specifically like features of a solid state disk (SSD). Flash based solutions offer many advantages such as being faster, more power efficient, more rugged than rotating media, and being more easily integrated together with other chips in system design and production flows. Tremendous technology advances have decreased cost and reduced prices significantly.

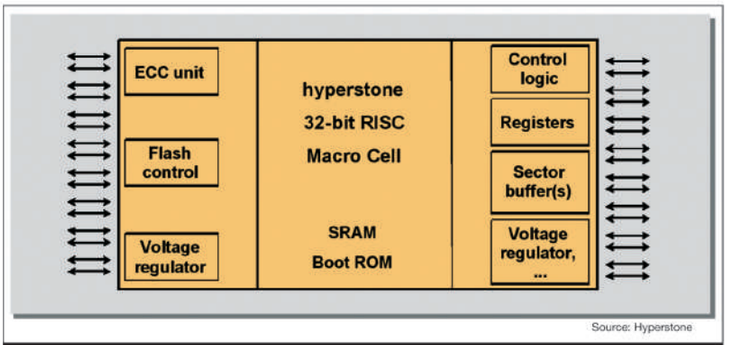

Fig 1: Generic block diagram solid state memory solution

These advances however are more and more demanding for very intelligent control functions.

Requirements for highly reliable flash memory storage systems or in SSD are more demanding than in e.g. consumer grade SD cards:

- Decent mechanism to detect and correct errors inherent to flash memories
- Efficient algorithms for wear leveling
- Methods to maximize and estimate life time of systems
- Methods to monitor failure rates
- Ability to implement customer-specific features
- Sufficient transfer rates for read and write operations

New technology shrinks, interfaces and cell technologies will continue to improve performance and capacity of flashes and drive cost per megabyte down. Furthermore, new packaging and manufacturing processes will continue to drive overall system cost down and capacity up as well. As a result, controllers and firmware feature will become more important as an enabling and competitive factor of applications.

While ensuring highest performance, reliability, enabling highest capacity, guaranteeing data integrity and maximum life-time are most important features determined by controllers, cost of controllers together with firmware represent only a small fraction of the overall systems' cost.

All solid state data flashes, whether based on NAND, ORNAND, or AG-AND technologies, are cost efficient because it is accepted that cells have errors right from production and wear down during cells' life.

In the case of single-level cells (SLC) each cell stores /reflects one bit. For multi-level cells (MLC) two bits of information can be represented per cell with 8 bits per cell already being considered.

The smallest logical/administrative unit is a sector. Each sector contains 512 bytes plus an overhead area of 16 bytes.

A block has a defined number of sectors, e.g. currently between 16 and 512. There are 1 to 8 thousand blocks per chip. Erasing is done at the block level. As blocks are the "management" or "administrative" unit, blocks will wear out as cells break down after a number of erase cycles. When defective, blocks will be considered bad blocks. Erase cycles or lifetime of flash is typically specified as a minimum of 100,000 cycles for SLC and 10,000 cycles for MLC. In application, a flash's life can be much higher by making use of wear-leveling algorithms.

Sectors are grouped into pages within blocks. Blocks could include 32 pages of 512 bytes or more recently 64 pages of 2,048 bytes. Page sizes are in the range of 512 or 4,096 bytes. Writing or programming is done at the page level. Also, programming requires pages to be pre-erased. Once pre-erased, programming or writing can be done one page at a time or by addressing a sector within a page that is pre-erased or "empty". Depending on flash memory type, pages can be accessed up to 8 times, when writing a sector.

Flash memories from different manufactures vary greatly with respect to bus width, number of blocks, block size, page size, number of dies, and programming capabilities, including caches. These characteristics pose some interesting challenges to the controller and its firmware:

- How to re-write to areas already containing data?
- How to maximize product's life time with only a limited amount of erase cycles available?
- How to ensure data transfer integrity?

Flash controllers of all kinds usually consists of an interface to the flash memory, a processor and a host interface – see figure 1.

Hyperstone's controllers are based on a 32-bit reduced instruction set

Axel Mehnert (amehnert@hyperstone.com) is director of marketing & customer support at Hyperstone GmbH (Konstanz, Germany).

computer (RISC) central processing unit (CPU) together with dedicated hardware blocks including error correcting code (ECC) unit, buffers, flash and host interface control logic – see figure 2.

As an example of a more complex controller, the hyperstone F3 offers two channels, 'enhanced direct flash access' each with 4-bit error correcting code (ECC) unit capable of correcting 4 symbols in a 512 bytes sector. Up to 16 flash memory chips (eight chip-selects per channel) can be connected directly. The Hyperstone 32-bit RISC microprocessor can be scaled running at clock frequencies from 10 MHz to 60 MHz using a trimmable internal oscillator. The F3 offers 16 kbytes internal boot ROM, 20 kbytes internal SRAM, four 512 byte sector buffers and 256 byte Personal Computer Memory Card Association (PCMCIA) attribute memory. Performance with SLC flash memories of 40 Mbytes/s reading and over 30 Mbytes/s writing is achievable. With automatic power-down mode during wait periods for host data or flash memory operation completion and automatic sleep mode during host inactivity periods, an Icc of less than 100 μA can be achieved.

The F3 is specifically designed for applications such as high-speed CF cards, disk-on-module, or solid state disks. It supports automatic sensing of PCMCIA or True-IDE host interface mode.

Compliant to PCMCIA 2.1, PC Card Advance Technology Attachment (ATA), and CF 3.0/4.0, memory mapped or I/O operations, fast ATA host-to-buffer transfer rates, programmed input/output (PIO) mode 6, MDMA mode 4, and ultraDMA mode 5 in true-Integrated Drive Electronics (IDE) mode. While offering parallel-ATA (PATA), serial-ATA (SATA) can also be realized using an additional PATA to SATA bridge chip.

The firmware, stored in flash memory, is application and host interface specific. All tasks with respect to flash and data management and data transfers between flash and host are implemented either in hardware or in software. Hyperstone flash controllers boot-up using firmware that is stored within the flash memory of the product. Other solutions might store firmware in the ROM of the controller. Therefore, based on identical product hardware, manufacturers are able to provide different products or feature sets. Also, firmware could be updated in the field or immediately before delivery. The firmware is copied into the flash in a so called pre-formatting process after the storage product has been assembled.

Several algorithms and concepts are used to address the questions initially posed re-writing to areas, maximizing flash life time, and ensuring data transfer integrity.

Blocks and Block Mapping: Logical block addresses (LBA) including the related static sector address or information are mapped corresponding to physical block addresses (PBA). A table is maintained by the controller or firmware translating requests for particular LBA to its corresponding PBA. Logical blocks are distributed across all available flash chips and "good" blocks e.g. logical block 0 might correspond to a physical block at chip 1 block 2, and logical block 1 might correspond to a physical block on chip 3, block 3.

It all starts with the anchor block used to store vital boot information such as location of firmware (boot sector), the bad block table, and may contain certain boot relevant feature variable settings or configurations. Management blocks are used for mapping, wear leveling, logbook etc. The so-called commit block holds permanent data plus information to find the management blocks. A pool of buffer blocks is maintained, along with a table holding their status and mapping to its "Twin" user block. User blocks are addressable blocks for user data. And finally, defect blocks are unusable or bad blocks and a pool of spare blocks is used to replace blocks that become bad during card lifetime.

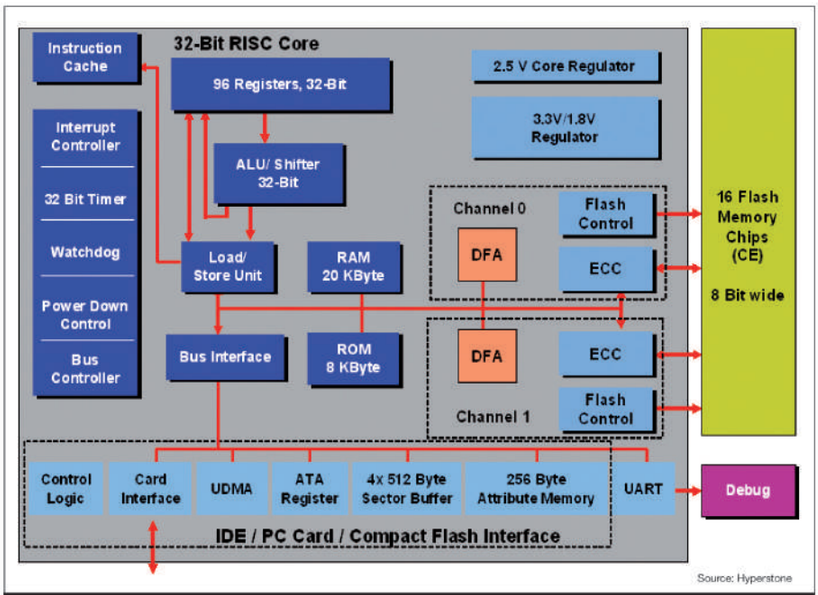

Fig 2: Hyperstone F3 flash memory controller for CF cards or solid state disks

Bad Block Management: During pre-formatting flash memories are tested and bad blocks marked by the manufacturer are mapped out. Bad blocks are entered into a bad block table.

A pool of spare blocks is defined and used for dynamic bad block replacement, where blocks from the pool are used to replace buffer blocks that produce errors when erased or programmed. This is called "bad block re-mapping. The size of the spare block pool, while usually in the range of 2 percent, can be customized and has an impact on a product lifetime.

Wear Leveling: Hyperstone's firmware provides patented algorithms balancing the use of all blocks guarantying a maximum lifetime of products. Wear Leveling is triggered at buffer block erase time. All blocks are classified into so-called wear level classes. Whenever a block is erased, a counter is increased. If the erase counter reaches a defined threshold, the block's wear level class is increased. By comparing wear level classes of blocks to be used, the load is balanced throughout all blocks. Depending on products' overall capacity and the data amount per write cycle the effect might be significant. By trying to ensure that as few blocks as possible reach their end-of-life before others, and together with defining an adequate size of the spare block pool, the products life can be maximized and data integrity be ensured.

A complementary available software tool, 'read wear level count', enables system developers to read block usage statistics, showing how many blocks have been erased certain number of times. Listing the wear level classes and the related number of blocks for each class can support qualification and testing efforts of products based on Hyperstone controllers.

Error Code Correction: This is done on a bit/byte level within sectors. Several algorithms could be used; most common today is Reed-Solomon code. Depending on the cell structure and quality of the flash, this task is of increasing importance.

Example concepts are e.g. a 10 bit symbols, parity and syndrome calculation in hardware, correction in software with hardware (instruction set) assisted Galois field arithmetic where four symbols can be corrected. Alternatively, a more flexible binary Bose, Ray-Chaudhuri, Hocquenghem (BCH) code with parity and syndrome calculation as well as correction could all be done in hardware. For this concept, six bits are correctable for a 16 byte overhead area per 512 data bytes, or 14 bits correctable for a 32 byte (30 byte or larger) overhead area. As algorithms can either be implemented in software or in hardware there is a trade-off between flexibility to support many different flash memory cell architectures and performance of the ECC.

Power Loss Protection: Hyperstone has developed a patented concept in order to ensure data integrity when transferring or writing data. By using certain buffer blocks, information is written in a way minimizing the delta between an old and a new state. The data system is coherent at all times.

On power down, the controller is reset and the flash immediately write protected. A log of the most recent Flash transactions is kept, where entries are made just before any programming to the flash. Should the last entry of the log be corrupted, the controller recovers the last valid entry. This minimizes data loss due to power failures and data corruption at the physical layer is prevented completely. Only if the power loss happens at the very same time when data is written to the flash, this data might get lost. In no case however will the overall data system be corrupted.

Several features can be realized aside of standard features. Cards can be defined as read-only, content can be protected, and writing to a card can be limited or defined in number, e.g. write-once, -twice, x times. Event triggered write-protection is possible as is. making cards workable only in certain devices. Proprietary hardware format can make use of standards such as SD/MMC, ATA, or IDE together with firmware comprising a disk-on-board. Requirement for qualification, tight quality control, or simply longer life cycles compared to consumer products might easily be addressed, and result in specific products or in-sourcing.

The cost of an overall flash based storage system is mainly driven by the flash memory component. Decreasing cost of flashes, results in more complex tasks for the systems' flash controller. A fine-tuned balance between hardware and software is necessary for these controllers to provide a robust and performing system.

As controllers and firmware built the heart of flash storage systems, features and performance are a vital factor in enabling and competitiveness of applications. Ensuring highest performance, reliability, enabling highest capacity, guaranteeing data integrity and maximum life-time are some features significantly determined by the controller. ■

Online:
Flash to surpass DRAM capacity in '08.
www.eetimes.eu/202200329