# SSR (Simple Sector Remapper) the fault tolerant FTL algorithm for NAND flash memory

Gui-Young Lee[1], Bumsoo Kim[1], Shin-han Kim[1], Byungsoo Jung[1]
[1] Software Center, SamSung Electronics Co. Ltd,
Apkujon Bldg., 6th Fl., ShinSa Dong, Kangnad-ku, Seoul, Korea,
Tel. +82-2-3416-0439, Fax.: +82-2-3416-0400
e-mail: gui0@samsung.com

**Abstract:** In this paper, we introduce new FTL(Flash Translation Layer) driver algorithm that tolerate the power off errors. FTL driver is the software that provide the block device interface to the upper layer software such as file systems or application programs that using the flash memory as a block device interfaced storage. Usually, the flash memory is used as the storage devices of the mobile system due to its low power consumption and small form factor. In mobile system, the state of the power supplement is not stable, because it using the small sized battery that has limited capacity. So, a sudden power off failure can be occurred when we read or write the data on the flash memory. During the write operation, power off failure may introduce the incomplete write operation. Incomplete write operation denotes the inconsistency of the data in flash memory. To provide the stable storage facility with flash memory in mobile system, FTL should provide the fault tolerance against the power off failure. SSR (Simple Sector Remapper) is a fault tolerant FTL driver that provides block device interface and also provides tolerance against power off errors.

## 1. Introduction

In this paper, we introduce a new Flash Translation Layer (FTL) driver algorithm that tolerates power-off errors. FTL driver is the software that provides the block device interface to the upper layer software such as file systems or application programs that use the flash memory as a storage device.

Usually, the flash memory is used as the storage devices of mobile systems due to its low power consumption and a small form factor. In mobile systems, the state of the power is not stable. So, a sudden power-off can be occurred during the read or write operations are held on the flash memory. To provide the stable storage facility with flash memory in mobile systems, FTL should provide the fault tolerance against the power-off failure.

In this paper, we provide SSR (Simple Sector Remapper) the new FTL for NAND flash memory. SSR provides block device interface and tolerance against power-off errors.

## 2. Related Work

Flash Translation Layer (FTL) driver is the software that provides the block device interface to the upper layer software such as file systems or application programs that use the flash memory as a storage device. M-System s TrueFFS [3] and Datalight s FlashFX [4] is a typical FTL software. The software s component, which supports re-mapping algorithm, is called as FTL in TrueFFS and as VBF in FlashFX. M-System s FTL is adopted as the standard FTL of a PC card. AMD s DMS [1] is also a mapping driver such as FTL and VBF. However, it stores all mapping information on memory so it is not appropriate to use in systems with a small memory.

To access a flash memory by a block unit, a flash memory is divided into smaller blocks with a fixed size. Numbers are assigned to each divided block in order and these numbers are called as a Physical Sector Number (PSN). When the data on a block is changed, it is impossible to overwrite the old data with the new data directly in flash memory. To change the old data, there are two different methods. First method is to erase the old data first and then write the new data. However, the erase operation can be performed only by erase block unit. Therefore, other blocks, which belong to the same erase block unit, will be erased together even though the data on those blocks is not changed. Second method is to write new data on a free block and then sets an erase flag on a block, which contains the old data. When new data is written on a free block as described above, the Logical Sector Number (LSN) is not changed and only the PSN is changed. LSN is the address, which is used by programmers or application programs. When LSN remains same and PSN is changed, it is also necessary to update the mapping information between LSN and PSN. There are three major mapping algorithms for LSN and PSN mapping. First is a fixed mapping algorithm. Second is a random mapping algorithm. Third is a hybrid mapping algorithm.

## 3. SSR

SSR is using a hybrid mapping algorithm. LSN (Logical Sector Number) is assigned to each sectors and the assignment method is similar with hash function, which is used for key value assignment to each bucket (Search (Hash(LSN))=PSN). In actual implementation of SSR, sector allocation table or hash function can be used. There are also several search algorithms to find required LSN inside the bucket.

In SSR, using an erase unit instead of a block to perform an erase operation. An erase unit is consisted of several blocks from each flash memory. N blocks consist of one erase unit and the user can define the number of blocks that consists the erase unit.

Physical Unit Number (PUN) and Logical Unit Number (LUN) are assigned to each erase unit. PUN is decided by the physical order of each sector at NAND flash memory, and LUN indicates logical order of each erase unit. During flash memory initialization, memory with the mapping information between PUN and LUN is loaded to the main memory, and as the data is erase or modified, this mapping table is updated according to mapping changes between PUN and LUN.

An erase unit can be divided into smaller sector and each sector is used as the basic unit of NAND flash memory operation. The size of each sector is same with the page size in NAND flash memory. At each erase unit, first sector contains the configuration information of the corresponding erase unit (such as the size of an erase unit) and control information (LUN, wear level). The first sector of each erase unit is called as an Erase Unit Header (EUH).

At each erase unit, the spare arrays of the sectors are used for BAM (Block Allocation Map). BAM has the LSN and status information about each sector. LSN is the address of a sector, which is used by programmers or application programs. BAM shows how LSN and PSN (Physical Sector Number) are mapped to each corresponding address.

To access one sector with LSN, it is necessary to find out which LUN it is located at. LUN can be found by using LSN with hash function or hash table. After LUN is found, LUN-PUN mapping table is used to find out PUN. After PUN is determined, BAM of this PUN is accessed to find out PSN.

More details of the mapping algorithm of SSR are provided in the full paper [7].

## 4. Power Failure Recovery

In SSR, the whole data cannot be used, if mapping information is lost by a power-off failure. It is necessary to have a consistent ECC to recover mapping information.

Fig 3.1 shows NAND flash memory s sector data structure for SSR. Generally, flash memory s sector is divided into data and spare area. Data area is used to store the data and spare area is used to store sector s logical address, ECC value, and FTL control information.

In spare area, ls indicates LSN, cnt is a flag for sector status, ecc_ls is ecc value for sector number, and ecc_dat is an ecc value for the data.

Cnt field stores 4 bit flag values. 4 bit flag values at cnt field indicate each sector s current status. The meaning of 4 bit flag values are explained below:

- 1111: free status
- 0000: Deleted status
- 1110, 1100, 1000: Valid status

1111 means that this sector is an available free sector. When cn field has 0000, it indicates that this sector s data is invalid and will be erased later. Other 4 bit values are used to indicate current status. If more than two sectors have a same LSN, 4 bit values have a priority as shown below and the sector with a higher priority flag values is valid.

$$1110 > 1100 > 1000 > 1110$$

Based on above information, sector write and modification and ECC processes are shown at Fig 3.3.

In Fig 3.3, (a) column s (1) row indicates a free sector. To write the data on this free block, following steps are needed.

Step 1: Write cnt field and ecc_data ((a) s (2) in Fig 3.3)
Step 2: Write the data, lpn, and ecc_lpn ((a s (3))

If power-off error happens during or right after step 1, ECC checks the data area and ecc_data to find out whether data write is processed or not, and sets cn field in spare area as deleted status during recovery process. If power-off error happens during step 2, ECC checks lpn and ecc_lpn to find out whether current process has been finished or not. If the current process was not finished, ECC changes cn field to deleted status.

To modify the old data, it writes new data at an available free sector. These modification steps are shown at (4), (5), (6) in Fig 3.3. In each row, (a) column indicates a old sector and (b) column indicates a new sector. Steps at each row is explained below:

- (4) row: Subtract one from cn field and write this new cnt field value and ecc_data at a new sector.
- (5) row: Write new sector s data, lpn, and ecc_lpn.
- (6) row: Set old sector s cnt field as 0000.

If power-off error occurs during above processes, ECC compares cnt field values. The sector associated with unfinished process has a smallest cnt field value and this sector s cnt field status is set to deleted status.

## 5. Conclusion

To use flash memory more efficiently, hardware oriented device driver is necessary. Flash memory driver needs to consider flash memory s unique erase properties. In mobile systems, the state of the power is not stable. A sudden power-off can be occurred when we read or write the data on the flash memory. In mobile systems, FTL should provide the fault tolerance ECC against the power-off failure.

This paper introduces SSR algorithm, which is designed to provide ECC for power-off failure and to manage the memory more efficiently with a better mapping algorithm.

## References


[1] Detailed Design Document - AMD AM29DL DMS, Advanced Micro Devices, June 1988.

[2] Samsung Flash Memory KM28DU160-T OS Independent Device Driver API user's guide, Samsung Electronics, 1999.

[3] Raz Dan and Jeff Williams, A TrueFFS and FLite - Technical Overview of M-Systems' Flash File System, M-Systems, March 1997.

[4] Flash FX 4.0, Datalight, Inc., 1999.

[5] Flash File System Selection Guide, Intel Corporation, December 1997.

[6] KM28DU160-T/B user's guide Revision 0.5, Samsung Electroncis, 1998.

[7] SSR: Simple Sector Remapper version 1.0 technical report, Samsung Electroncis, 1999.


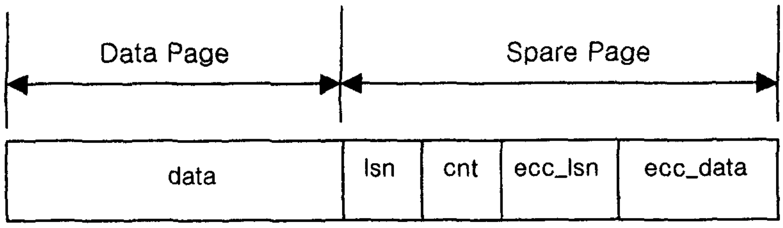


**Figure 3.1 Sector data structure**

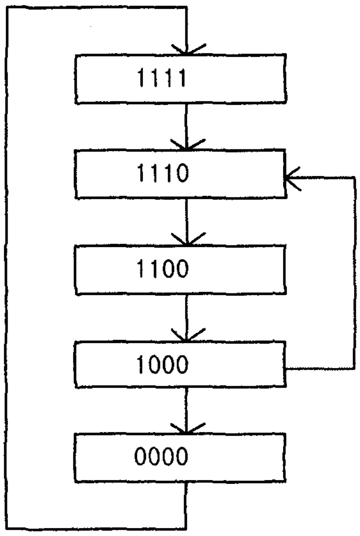


**Figure 3.2 cnt field's wrap counter**

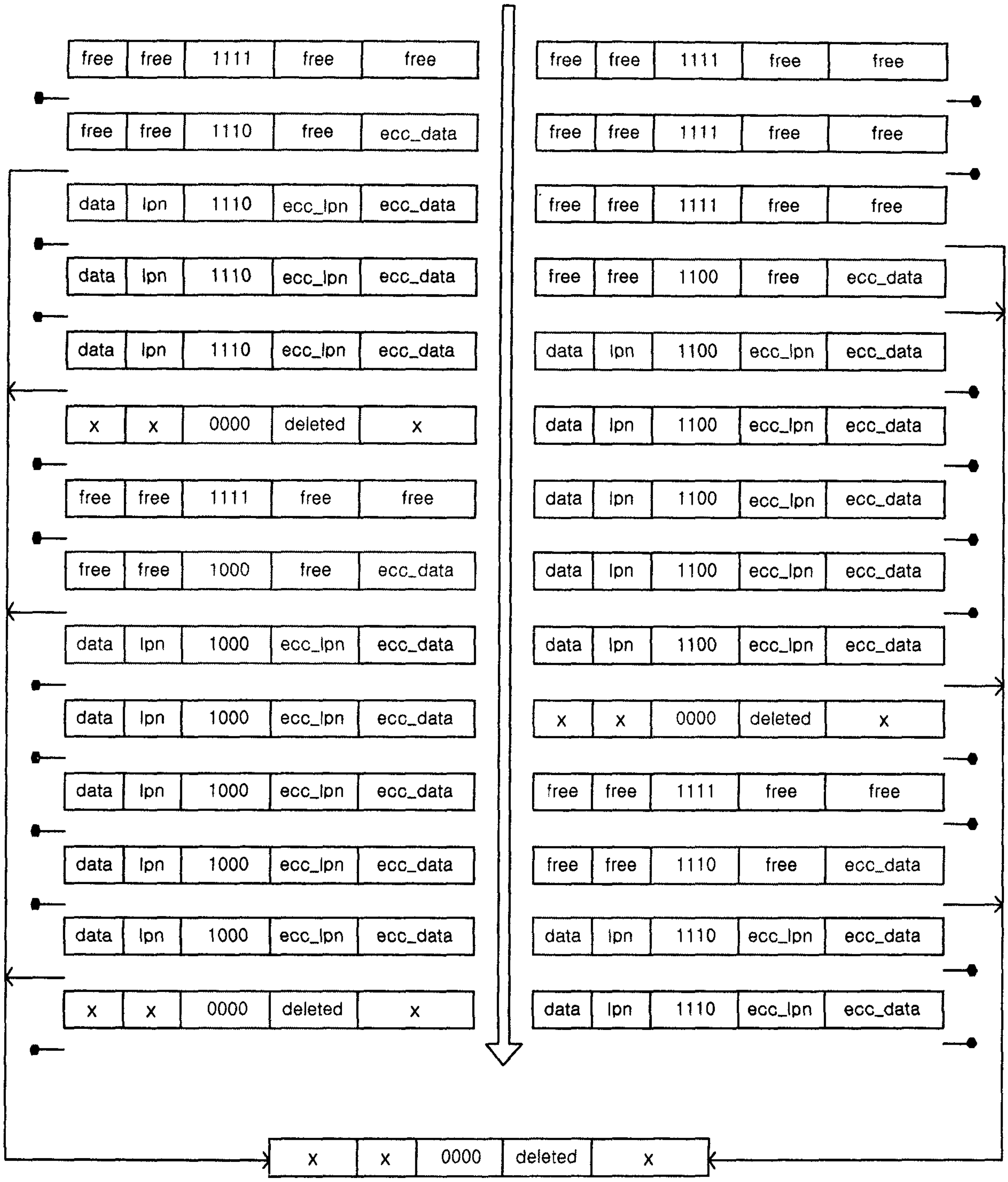


**Figure 3.3 Sector Record Control**