TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 12, No. 9, September 2014, pp. 6940 ~ 6945
DOI: 10.11591/telkomnika.v12i9.4616

Schemes for Extending the Lifetime of SSD

Kai Bu*[1], Hui Xu[2], Qiyou Xie[3], Wei Yi[4]
School of Electronic Science and Engineering, National University of Defense Technology,
Changsha 410073, People's Republic of China,
*Corresponding author, e-mail: booquai@163.com[1], Xuhui@nudt-esss.com[2], yiwei409@163.com[3]
xqy_gratitude@163.com[4]

Abstract

The limited use lifetime is a significant drawback of the solid state disk (SSD). When the P/E cycles exceed the nominal endurance limit, the entire SSD is rendered non-functional as worn out. As the low endurance, the MLC Flash based SSD is not suitable used in lifetime-aware applications and the total-byte-written of conventional MLC SSD is much less than SLC SSD. In this paper, we proposed an Adaptively Level Adjusting Scheme to use the MLC Flash dynamically to storage different amount of data levels through the entire lifetime. The result shows that the MLC SSD adopting this method could be totally written 4.8X more data than conventional MLC SSD and 16.5% more than SLC SSD. Another simple method named Post Life Slavaging Scheme is also proposed to benefit from the post life of a MLC SSD. It can extend the real P/E endurance by 2X when the user capacity reduced to 92% of original cacapcity.

Keywords*: SSD, NAND flash, endurance, P/E cycle, use lifetime.*

1. Introduction

The solid state disk (SSD) based on NAND Flash memory has been widely used in consumer and industry applications due to its high performance. Comparing with HDD, the SSD is more advanced at read and write performance, power consumption. But for the NAND Flash is an endurance-limited non-volatile memory technology, that the memory cell would degrade after each erase and program operation, the SSD could only be used reliably before the memory cell is worn out. The use life of SSD is generally decided by the endurance of the Flash chip and is still the topic being mostly worried about. Generally, due to the Flash physics feature constraints, the flash endurance could be measured with the factor of P/E cycles. Most SSD manufactures use a fixed number of P/E cycles, e.g., 10K P/E cycles for MLC and 100K P/E cycles for SLC, provided by flash manufacturers as a primary factor to manage the lifetime of SSD [1].

The MLC Flash based SSD has been used more widely in consumer market than SLC based SSD as it's much cheaper at the capacity cost. However, because the nominal endurance of MLC Flash is much less than that of SLC Flash, it's not suitable to be adopted in lifetime aware applications, such as enterprise market, although it's much cheaper than SLC SSD. As the market demand for cheaper SSD rather than expensive SLC SSD, many works have focused on how to adopt MLC SSD to replace SLC SSD by utilizing powerful ECC algorithm and wear-leveling algorithm. The most important is to extend the lifetime of the MLC SSD. In this paper, based on the endurance concept of flash cell, we proposed an Adaptively Level Adjusting Scheme and split the entire lifetime process into three stages. The MLC Flash cell is dynamically used to storage different amount of data levels at each stage. It could be used to replace SLC SSD with MLC Flash in the lifetime-aware applications with lower cost.

2. NAND Flash Endurance Analysis

The NAND Flash memory stores data by keeping charges in the floating gate which is between the control gate and substrate layer in a MOS transistor. The floating gate is isolated from substrate layer by the SiO2 tunnel oxide layer, which is a semiconductor. The NAND Flash write operation contains program state and erase state. Both program and erase operations depend on the Fowler-Nordheim (FN) effect. Much previous works [2, 5] has showed

Received November 3, 2013; Revised May 25, 2014; Accepted June 8, 2014

that during FN current injection, injected electrons would be trapped both within the bulk and at the interface of the oxide [2, 3].

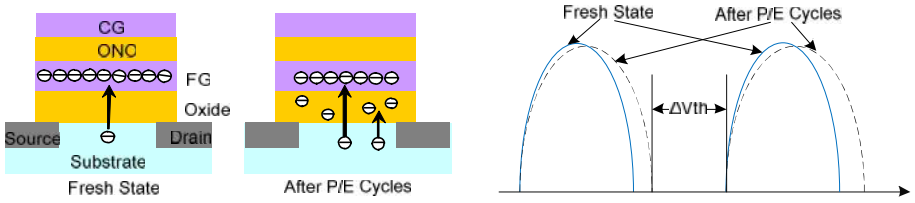


Figure 1. The Charge Trap and Threshold Voltage Shift due to P/E Cycles

The charge trapping will increase as the NAND Flash memory P/E cycling, which will directly result in the potential accumulated in the oxide layer as showed in Figure 1. The trap electrons and potential model has been proposed in the paper [3, 4] as Nt=A× Na + B × Nb. We use Nt to indicate the number of trap electrons both at the interface and in the oxide bulk. As described in the model function, we use δVth= (Nt ×q)/Cox to indicate the threshold voltage shift of floating gate induced by charge trap after P/E cycles. Here q is the electron charge and Cox is the capacitance of the oxide.

As shown in Figure 1, there must be an enough wide voltage margin to clearly distinguish between different voltage levels. When the charges trapped in the oxide result in a threshold voltage increase over ΔVth, the lower level would shift cross the higher level and it will not be possible to reliably read from or write to the memory cell. And this situation causes a write error or read error. Once the endurance limit is reached, a page is considered to have failed and is no longer usable.

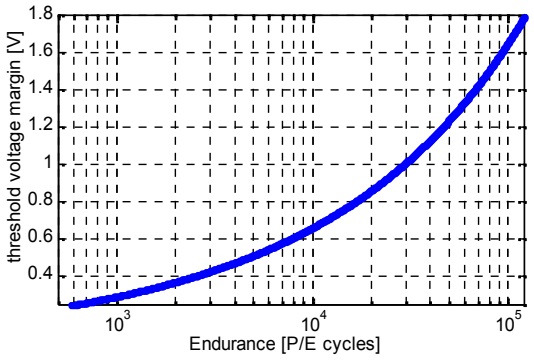


Figure 2. Endurance vs. Voltage Threshold Margin

The programmed level threshold voltage is not taken into account as its effect on the recovery. When the threshold voltage of the cell exceed the margin, the cell is judged as worn out. We use δVth = ΔVth to express endurance limit of Flash.

The endurance distribution of cell under different margins is showed in Figure 2. We calculate the endurance under the margins from 0.65V to 1.7V [4], which simulates the threshold voltage margins of MLC and SLC. The endurance spread from 10K to 100K when the threshold voltage margin extends from 0.65V to 1.7V. The endurance of MLC Flash is much less than that of SLC Flash for the narrower threshold margin.

3. Adaptively Level Adjusting Scheme

The basic physical feature and concept of MLC Flash are the same as SLC Flash while it could stores 4 levels each cell which is the double of the SLC. But the nominal endurance of

MLC Flash is much less than that of SLC Flash for the narrower threshold voltage margin. If the threshold margin of MLC could be extended, the lifetime would also be enlarged. A MLC based SSD is mostly used for consumer application because its low cost while the SLC is mostly used for enterprise application for longer lifetime. Compared with SLC technology, the MLC technology could support 2X capacity with penalty of 90% lifetime degradation. In this paper, we are interesting in leveraging the capacity advantages of MLC technology to build a SSD that could replace a SLC based SSD in the lifetime aware applications. We proposed an Adaptively Level Adjust Scheme (ALAS) to extend the use life time of MLC Flash. The method contains three stages as showed in Figure 3.

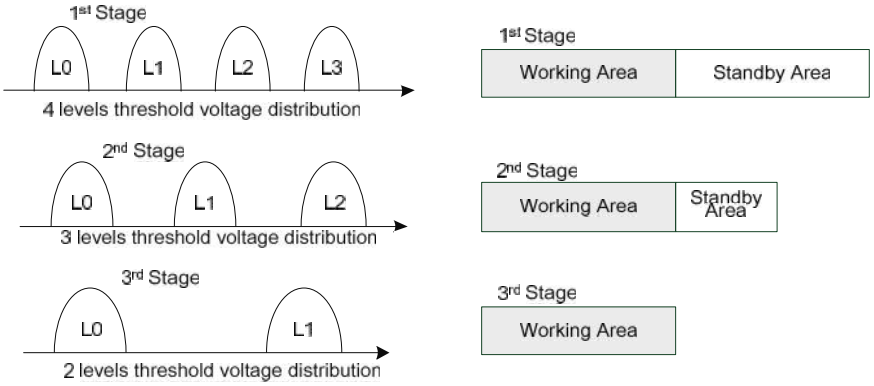
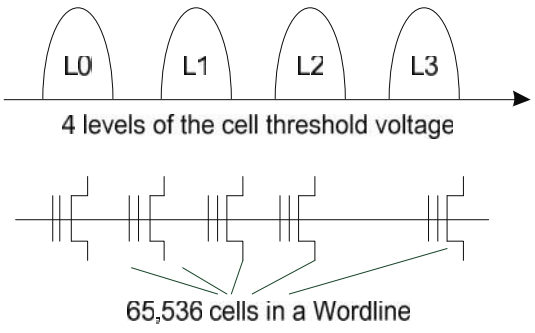


Figure 3. Adaptively Level Adjust Scheme

For a SSD, we organize the SSD with logical pages of 4KB. Logical pages are made up by cells belonging to the same Wordline (WL). The number of pages per Wordline is related to the storage capabilities of the memory cell. The MLC Flash is operated as 4-level in each cell. In Figure 4, for a MLC device with a Wordline of 65,536 cells, which could stores 128Kbit data, there are four pages as each cell stores one Least Significant Bit (LSB) and one Most Significant Bit (MSB) [6].



2 bit data /cell		Level in cell
MSB	LSB	
1	1	L0
1	0	L1
0	0	L2
0	1	L3

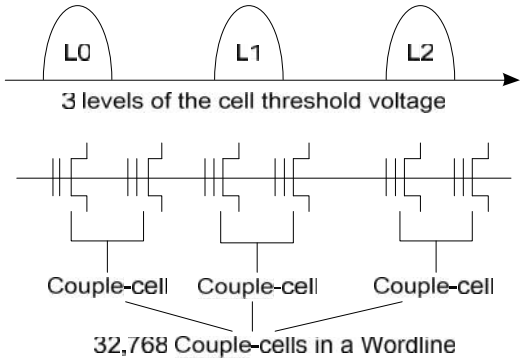
Figure 4. 4-level MLC bit Mapping

During the first stage, we set the user capacity to only half of total raw capacity. It means that there is only 64GB user capacity for a SSD which consists of sixteen 8GB MLC Flash memories. All of the pages are dynamically split into two areas dynamically, half in Working Area and half in Standby Area. Through the logical-physical address mapping in the Flash Translation Layer, although the pages within the Standby Area are not visible to users, these pages take part in the access when they are swapped with the pages within the Working Area. All pages would be accessed relatively balanced with the function of weal-leveling [1].

When the SSD P/E cycle almost reaches the nominal number, we adjust the data storage levels from 4 to 3 of each cell. Thus, for a MLC device with a Wordline of 65,536 cells, it could stores 96Kbit data as each two cells (we say it a Couple-cell). As Figure 5 shows, there are three pages in each Wordline: MSB page, HSB page and LSB page. Compared with the 4-level MLC Flash, the 3-level devices could offer ¾ pages. As it only need to store 3 levels per cell, the threshold voltage margin could be relaxed. So the cell could wear more P/E cycles than the 4-level cell.

Similar to the logical page organization in the first stage of the SSD, there are also Working Area and Standby Area. 1/3 of all the pages are not visible to users for they are within the Standby Area. And 2/3 of all pages take part in the Working Area. As a fact of result, the user capacity is still 64GB. The invisible 32GB Standby area also take part in the access as swapping pages.

As the 3-level Flash based SSD working, the cells degradation more and the threshold voltage distribution spreads wider. When the cells almost wear-out, we adjust the data levels from 3 to 2 per cell. At this time, the MLC Flash has been degrade to a SLC-type Flash. The cell threshold voltage window is set to as conventional SLC Flash, and read margin would be extended much more compared with MLC Flash. The entire endurance could reach the level of SLC Flash memory. For there are only two levels per cell, which means 1bit/cell, the raw entire available capacity is only 64GB. Within the 2-level Flash based SSD, all pages is located at the Working Area.



3 bit data /Couple-cell			Levels in Couple-cell
MSB	HSB	LSB	
0	0	0	L0L0
0	0	1	L0L1
0	1	0	L0L2
0	1	1	L1L0
1	0	0	L1L1
1	0	1	L1L2
1	1	0	L2L0
1	1	1	L2L1

Figure 5. 3-level MLC bit Mapping

Therefore, through the entire lifetime, the user capacity is always 64GB without capacity penalty by means of the Standby Area capacity decreasing gradually.

4. Lifetime Analysis of SSD Adopting ALAS

The threshold voltage margin is different from each other according to the levels of each stage. As the amount of levels decreasing, the threshold voltage margin increases. From the first stage to the last one, we choose 0.65V, 1.1V and 1.7V as the threshold voltage margins [4]. So the endurance for each stage is 10K, 23K and 100K P/E cycles separately according to the simulation results from Figure 2. It is reasonable to expect that the MLC memory with adaptively level adjusting scheme can sustain the same amount of P/E cycles in conventional SLC memory.

The endurance is the key factor related to the SSD use lifetime. But, in a real application, we usually use total-byte-written (TBW) to evaluate the lifetime of a SSD. The TBW means the total amount of data could be written into the SSD before it's worn out and it's expressed as TBW = Capacity × Endurance. With the same workload, the SSD that has more TBW could work for longer time before it's worn out. For the SSD using the adaptively level adjusting scheme, the TBW is the summation of the total written data during three stage as TBW =TBW1+TBW2+TBW3. The results in Figure 6 show the TBW of each stage and the Figure 7 shows the results of the TBW of different SSD technology. The 64GB MLC SSD using

the adaptively level adjust scheme implies 16.5% more write operations than conventional SLC SSD and almost 5.8X the TBW of 128GB MLC SSD.

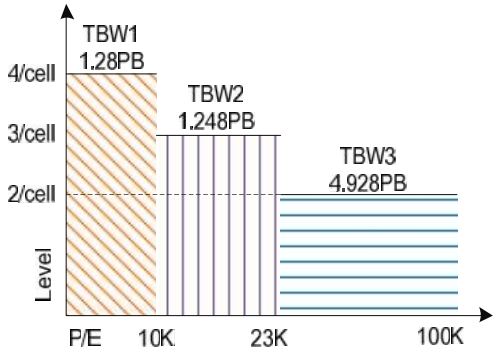


Figure 6. ALAS Process and TBW of each Stage

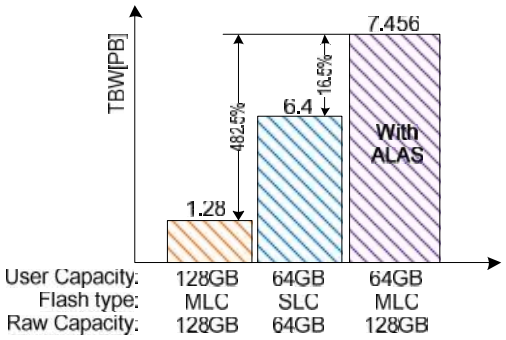


Figure 7. TBW of Different SSD Technology

5. Post Life Salvage Scheme

Accoding to [10], for an MLC Flash-based SSD it shows that even the average P/E cycles of each block has been to 100K times, no more than 1% blocks are out of function. That is to say, more than 99% blocks are still available. So, the SSD is not dead entirely, it could be used in some ways by benefit from the residual available blocks. We proposed another scheme named Post Life Slavaging Scheme which consists of two ways.

First, we use the SSD as a read-only disk for archiving data. Flash memory is a non-volatile memory technology, and the retention time is more than 10 years. When a SSD worn-out, only a number of blocks became badblock and could not be write and read, the other part could still be read and write. When a SSD is initializing, there are blocks reserved for badblock replacement. As the SSD controller detects a bad block, the controller would migrate data into the reserved good blocks. Such that when a SSD is worn-out, the data stored in it could be still accessed. And we could use the SSD as a read-only disk. For the Flash has a read disturbance, which will induce read error after thousands read operations. According to the study [9], the read error would accumulate over the ECC ability when the read operation repeated over 10^5 times. In order to ensure the reliability of the read-only disk, we use a data auto-refresh method. We set the endurance threshold N times lower than 10K, this means the SSD have N P/E cycles reserved even when the SSD is detected as worn-out. And we set an auto-refresh threshold as 10^5. A read count register is used to store the read count of each block. When the block is read over the threshold, the block would be programmed with the data store in it again as an auto-refresh operation. And if the data store in the block has retained more than specified retention time (10 years), the auto-refresh operation will also be performed. If we set N to 10, a 128GB SSD could be read for 10^6 times entirely and the data retention could be up to 100 years totally.

Second, we extend the use life by reducing available capacity and offer more reserved blocks. As the Flash P/E cycling, the amount of worn out cells will increase gradually and the available capacity is becoming less. As usual, we must limited the endurance to support enough usable capacity for the real application [8]. The affordable badblock capacity portion is no more than 2% when Flash chip reach the specified endurance. 2% reserved blocks are enough for badblock replacement. And when the SSD is worn-out, we chose another 2% block as reserved blocks, and the available capacity partition is set to 96%. Before the SSD worn-out, we could get more 3K P/E cycles and another 2% blocks become badblock when the SSD worn out. By this mean, we could extend the amount of reserved blocks and get more P/E cycles. And the actual acceptable ratio depends on the application where the ratio could even be extend to 8% or more in a SSD design. So we will get the acceptable endurance over a range of badblock rates.

A 128GB SSD based on MLC Flash Chip with 256KB/block and 4KB/page organization was tested. All blocks in the SSD wear down in a balanced speed with the effect of wear-leveling algorithm. We use the life extension method and test the achievable P/E cycles. The SSD could be still read and wrote even it was considered as worn-out. And the real achievable P/E endurance is much more than 10K times. If we set the maximum affordable badblocks to 8%, the SSD could reach 20.1K P/E cycles when the SSD worn-out. The life extension approximates to the simulation results shown in Figure 8.

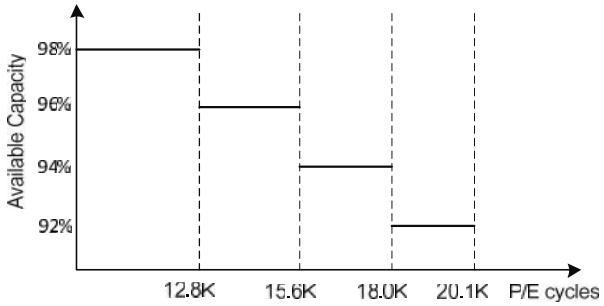


Figure 8. Extend the Use Life by Reducing Available Capacity

6. Conclusion

We introduced and calculated the Flash endurance based on the charge trap concept. As the MLC Flash having more levels than SLC Flash, its threshold margin is narrower and its endurance is much less. In the endurance simulation result, we could find out that the endurance will be extended while the voltage threshold margin expanding. Therefore, we proposed an Adaptively Level Adjusting Scheme to use the MLC Flash in the SSD. We took a 128GB MLC SSD as example, although the user capacity is only 64GB that is half of the raw capacity, the total bytes written through the entire lifetime is much bigger than the conventional 128GB MLC SSD. And it's even bigger than the TBW of 64GB SLC SSD. This method could also be used in the SSD based on TLC and QLC to replace the SLC SSD in the lifetime aware applications with lower cost. When the SSD is worn out, it entries the post life phase, so we propose another simple method named Post Life Slavaging Scheme to benefit from the post life of a MLC SSD. It consists of two methods to salvage the post life of the SSD. One is to use the SSD as a read only disk with auto-refresh algorithm. The other one is to extend the use life by reducing available capacity and offer more reserved blocks. We could extend the real P/E endurance by 2X when the user capacity reduced to 92%.

References

[1] Simona Boboila, Peter Desnoyers. *Write Endurance in Flash Drives: Measurements and Analysis.* FAST2010. San Jose. 2010: 115-128.
[2] G Ghidini. Charge-related phenomena and reliability of non-volatile memories. *Microelectronics Reliability*. 2012; 52(9): 1876-1882.
[3] H Yang et al. *Reliability issues and models of sub-90nm NAND Flash memory cells.* International conference on Solid-State and Integrated Circuit Technology, Shanghai, China. 2006.
[4] Vidyabhushan Mohan, Taniya Siddiquaal. *How I learned to stop worrying and love flash endurance. 2nd Workshop on Hot Topics in Storage and File Systems.* Berkeley. 2010: 3-3.
[5] Grupp LM, Caulfield AM al. *Characterizing flash memory: Anomalies, observations, and applications*. Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture. New York. 2009: 24-33.
[6] L Crippa, R Micheloni. Inside NAND Flash Memories. New Jersey: Springer. 2010: 261-270.
[7] Koji Sakui, Kang-Deog Suh. *Nonvolatile Memory Technologies with Emphasis On Flash*. New Jersey: John Wiley. 2008: 297-301.
[8] Chundong Wang, Weng-Fai Wong, *Extending the lifetime of NAND flash memory by salvaging bad blocks*. DATE 2012. Dresden. 2012: 260-263.
[9] Jian Justin Chen et al. Flash Memory Reliability, Nonvolatile Memory Technologies with Emphasis on Flash: A Comprehensive Guide to Understanding and Using Flash Memory Devices. Malden: Wiley Interscience. 2010
[10] Kai Bu, Hu Xui, et al. Salvage the Post Life of Solid State Disk. ICMIA. Guilin. 2013; 336: 1510-1513.