NAND Flash Bad Block Management Research Based On FPGA

Guohui Wu[1, a], Yongjie Hu[1,b] , Jian Wu[1,c]

[1]Nanchang Hangkong University, Nanchang 330063, China

[a]43151481@qq.com, [b]13697083633@163.com, [c]78313993@qq.com

Keywords: NAND Flash, bad block management, mapping table

Abstract. For the requirement of the stability in large-capacity data storage, this paper proposes a management method of bad block based on setting up block address mapping table in the interior of FPGA. While operating the NAND Flash , we can take advantage of the mapping tables to shield inherent bad blocks and new produced blocks .The research proved that the method can avoid the operation of bad blocks efficiently and improve the reliability of data shortage greatly.

Introduction

With the rapid development of semiconductor engineering, the technology of data storage with a high speed and large capacity has gained much achievement [1]. Owing to the characteristic of non-volatile, high density, large capacity, faster speed of storing ,more erasing times and so on, NAND Flash has gradually become the preferred storage medium in all mass storage device. Nevertheless, we can not guarantee its memory areas keeping quality during the lifecycle on account of the limit of NAND Flash technology. When leaving the factory, NAND Flash will contain a certain amount of bad block .At the same time; it will also produce some new ones in the process of using. Once considered as the bad block, it cannot be permitted to erase or write the operation. In brief, the article has presented a management method of bad block directing at large-capacity NAND Flash, which can be solved very well by means of setting up the address mapping table amt.

The physical structure and basic operation of NANDFlash

Take an example of Samsung company's chip[2] , the storage structure of chip is shown in Figure 1.It is divided into 32768 blocks, each one consisting 128 pages. What is more, the capacity of each page is （4K+128）Byte, 4K Byte regarded as data storage region for saving data and 128 Byte thought as backup area for storing ECC check code or label information of bad blocks. The smallest unit of read-writing in NANDFlash is a page and scratching is a block with the addition of the nonvolatile feature, as a result , every time we must erase the operation before writing the it.

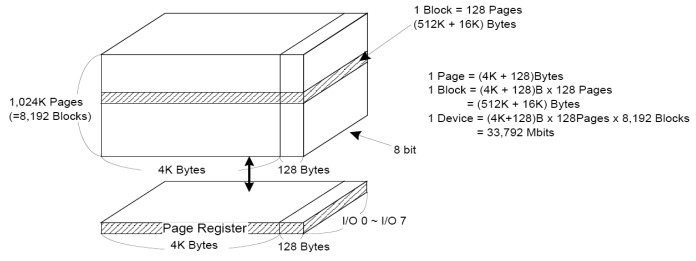

Fig.1 The NANDFlash physical structure

The bad management of NANDFlash

In either case, these blocks are independent of each other and the good blocks can be affected nothing but jumping over the bad ones[3]. There are some major work in the bad blocks management .To begin with, we should detect the block mark of every block in the chip and build up the block mapping table. Next, When writing and wiping the operation in the chip ,we can provide the location of good blocks

through the modules produced by block address of the mapping table[4]. Once new bad blocks generated during the process of using, we can continue to manipulate the NANDFlash and sign the block and update the mapping table by the standby block. Finally, The new mapping table is supposed to be preserve in the segment No.0 after finishing the operation.

Building up the block mapping table

When NANDFlash leave the factory, the manufactures will check it up and take found the inherent bad blocks on the specific location of 4096 bytes on the page that the first byte of spare area rather than the district which is called "ffh". We can discover the inherent bad blocks by testing the bytes and establish the block mapping table afterwards. The block diagram is shown in the figure 2. A case study of K9MDG08U5M, it contains 32768 blocks and we plan to divide them into 16 groups, each one having 2048 pieces and setting up a block mapping table[5].

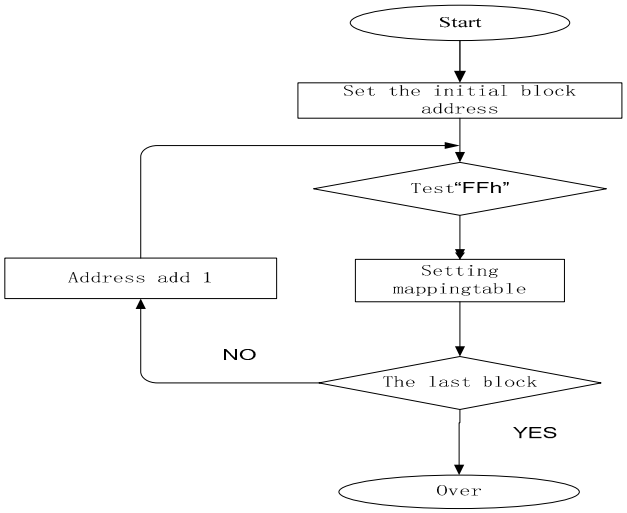

Fig. 2 The bad block of NANDFlash test flow chart

First of all, we transfer an RAM ,whose specification is 2048*16bit , in the interior of FPGA for recording the information of an group of mapping table. The positions ranking from 0 to 12 are designed for memorizing the physical address, namely the substantial address of the blocks. It is worth mentioning that the fifth is used for recording the quality of blocks. Given the block is good, we place it on the zone or otherwise make it one .And the third and fourth are used for inspecting whether the blocks are used .If they has been used, we should mark them "11" or otherwise label "00".we can establish the group mapping table by detecting the block symbols and remnant is also set up by the same method.

The producing of modules by blocks address

The function of generating modules for the block address is handling the flash memory with providing the standby block address and completing the mapping table renewal by marking the block when the new bad block appears, which can make good use of parallel property and improve the speed of sorting[6] .

When storing the data on the chip,to begin with , we read a group of mapping tables from the NANDFlash's zero block into the FPGA's RAM . Then we can read the message of mapping tables from RAM and extract the block address of good blocks for storing operation. We should find the next good block address as the spare block address by reading the information orderly in the RAM while carrying on the NANDFlash. Provided that spare address has been employed, we have to search for the next one. The failure during the process of operating demonstrates that the operation has become the new bad block ,in the which we'd better take the fifth position of mapping table information to the place of zone block and store the message of updated mapping table into the RAM

so as to accomplish the mapping table renewal .At the time of updating the mapping table, we continue to manipulate the NANDFlash by the spare block address . While finding out that the block has been used in the process of operation, we carry on the operation through the spare block and eliminate the used blocks. If clearing away successfully, we take the block's thirteen and fourth position of mapping table information to the place of zone block and store the former address of RAM .It shows that the block has become new bad blocks on the condition that the elimination is broken down. Hence, we should take the fifth position of mapping table to the place of zone one and store the updated mapping table information.

The shortage of mapping table

The key of the bad block management is the block mapping table, especially learning to save the mapping tables precisely. The vendor must insure the quality of former two blocks of chips when they leave the factory, which helps us store the mapping table to the flash former two blocks. Take example of K9MDG08U5M, we divide the all blocks into 16 groups and each one correspond one mapping table. After completing the building of mapping table, we should make the 16 mapping tables store to the pages from one to sixteen in the flash zone block [7]. When a group of mapping tables has been updated, we place the new mapping table of group to the corresponding location of the first one block. For instance, the initial mapping table has been stored in the zone block of the first page; we should take the updated mapping table to the first block of the first page [8]. When the all operation has been completed, we can clear away the content of the zone block and copy the undated mapping table in the zone one. In the meantime, we are supposed to eliminate the content of the first block for using next time in case of repetition of renewal [9].

The research simulation

The simulation of building the mapping table can be demonstrated in the figure 3. The bad_block and well_block help distinguish the quality of blocks. Given that the block is good, the well_block is keeping the high level for about one hour otherwise is the bad_block. The clock is the writing clock of RAM, the wraddress is the writing address of RAM, wren is the writing function of RAM, block_addr is the address of inspection block and data is the writing data consisting block address and block attribute. When the block locating on the place of "001h" has been detected well, we pull up the wren signal and write the message of data "0001h" into the "000h" address of RAM in the rising edge of clock. On the contrary, when the block locating on the place of "002h" has been detected badly, we pull up the wren signal and write the message of data "8002h" into the "001h" address of RAM in the rising edge of clock. By the same method, we can set up a group of block mapping tables.

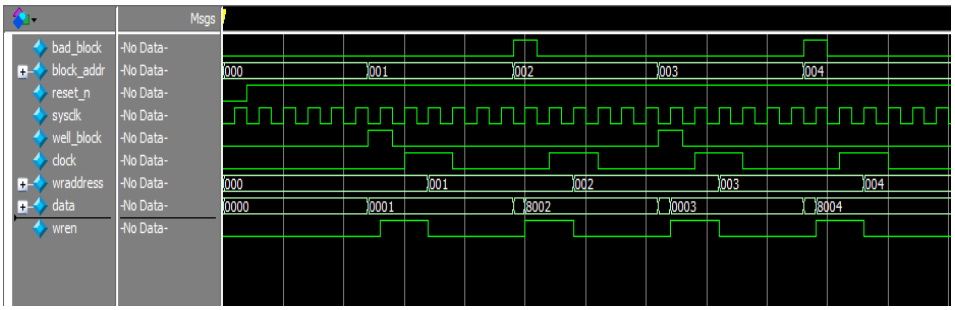

Fig.3 Functional simulation of building mapping tables

In order to supply the efficient block address of data storage, we harness the block to produce the modules. The function simulation can be shown in the figure 4. First of all, we should a series of mapping tables from the mapping table storage region of NANDFlash to the place of RAM, the internal of FPGA. Then we read the mapping table information ram_data from RAM. At the same time, we ought to estimate the numerical value of the ram_data's highest order. If it's located in the zone block, we should assign ram_data [12:0] to the block_addr for the address of block operation. When the operation of block_success returning a clock of high level is succeed, the writing address of

RAM add one and carry on reading the next block mapping table information. While manipulating one block, it reveals the operation went under and the new bad blocks have been produced in the case that the block_failure inputs a clock of high level. At the moment, the writing address of RAM is supposed to add one and we should search for the next one. In the meantime, we should update mapping tables and assign the highest order of ram_data to the data for the writing messages of RAM. The RAM writing address is considered as the original RAM address and the wren is made. We can complete the renewal of mapping tables in the rising edge of wrclk by writing the data into RAM. In the process of updating, the block_copy signal produce a clock of high level to the next level module so as to the data written in the bad blocks is copied to the efficient block.

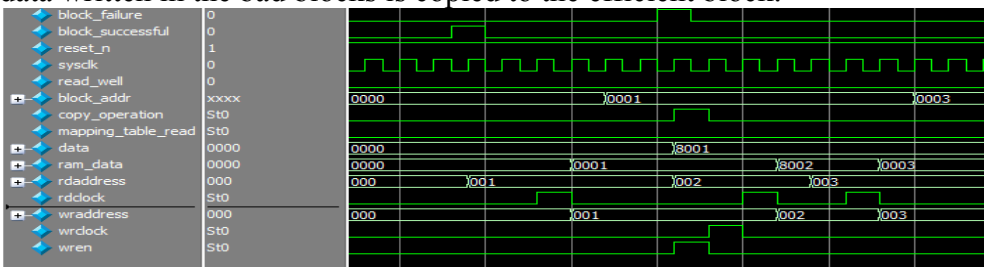

Fig. 4 the function simulation of updating mapping tables

Conclusion

Pass the bad blocks management, we can not only take effective shield on the bad blocks to avoid the operation of read-writing and scratching but also improve the read-writing speed and the accuracy of data storage by transfer the block address in the process of bad blocks management.

Acknowledgements

This work was financially supported by the National Natural Science Foundation of China(No. 61164015), the Natural Science Foundation of Jiangxi Province (No.20142BAB207021), Science and technology landing program of Jiangxi Province Education Department (No.DB201304126) and Science and technology support program of Jiangxi Province Education Department(No.DB201404071).

References

[1] Fishell, Wallace G. High-speed Solid State Recorder for Airborne Recce [J]. SPIE: 2829, Airborne Reconnaissance, 2010.

[2] Electronics K9MDG08U5M Advanced Flash Memory Datasheet[Z].2012.

[3] Kadowaki T,Yamakawa Y,Nakamura H,Kimura Y,Niwano M,Masuoka F. A NewAriticurture for High-Density High-Performance SGT nor Flash Memory. IEEE Transactions on Circuits and Systems.2010, 55(6):551-555.

[4] Park C,Seo J,Seo D,Kim S,Kim B.Cost-efficient memory architecture design of NAND flash memory embedded systems. Proceddings of 21st International Conference on Computer Design, 2013:474-480.

[5] Sang Lyul Min, Eyee Hyun Nam. Current trends in flash memory technology. Asia and South Pacific Conference on Design Automation.2010:45-46.

[6] Shengyong Zhang, Shijie Gao, Zhiyong Wu.etc. Flash NAND bad block processing method basedon FPGA[J].Computer Engineering.2013,36 (6):239-243.

[7] Yonghao Han, Shaoyun Wang. Design and implementation of a Flash NAND dynamic bad block Management algorithm[J].Information research,2014,37 (3):23-26.

[8] Xiao Luo, Hao Liu. Flash NAND bad block processing method based on FAT file system[J], Electronic device,2012,31 (2):716-719.

[9] Wei Liu. Design and verification of Flash NAND controller [D].Master Thesis.University of national defense science and technology, 2010.