AN1819
APPLICATION NOTE

Bad Block Management in Single Level Cell NAND Flash Memories

This Application Note explains how to recognize factory generated Bad Blocks, and to manage Bad Blocks that develop during the lifetime of the NAND Flash device.

INTRODUCTION

Bad Blocks are blocks that contain one or more invalid bits whose reliability is not guaranteed. Bad Blocks may be present when the device is shipped, or may develop during the lifetime of the device.

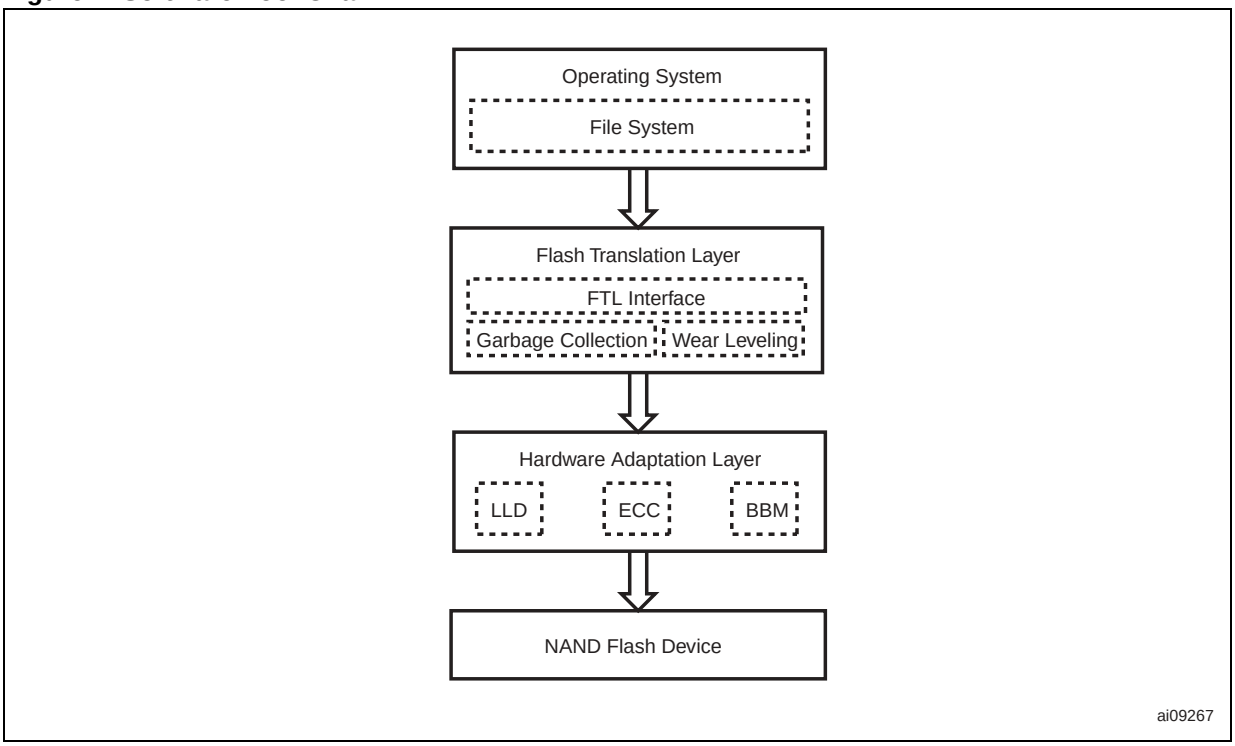
Devices with Bad Blocks have the same quality level and the same AC and DC characteristics as devices where all the blocks are valid. A Bad Block does not affect the performance of valid blocks because it is isolated from the bit line and common source line by a select transistor.

Bad Block Management, Block Replacement and the Error Correction Code software are necessary to manage the error bits in NAND Flash devices. ST provides this software in modules in the Hardware Adaptation Layer (HAL) which manages the hardware functions of the NAND Flash (see Figure 1. and refer to Application notes *AN1820, AN1823*).

This Application Note covers ST Small Page (NANDxxx-A) and Large Page NAND Flash memories (NANDxxx-B). Refer to the datasheets for the full list of root part numbers and for further information on the devices (see REFERENCES section).

TABLE OF CONTENTS

Figure 1. Software Tool Chain



RECOGNIZING BAD BLOCKS

The devices are supplied with all the locations inside valid blocks erased (FFh). The Bad Block Information is written prior to shipping.

For Small Page (528 Byte/256 Word Page) NANDxxx-A devices, any block where the 6th Byte/ 1st Word in the spare area of the 1st page does not contain FFh is a Bad Block.

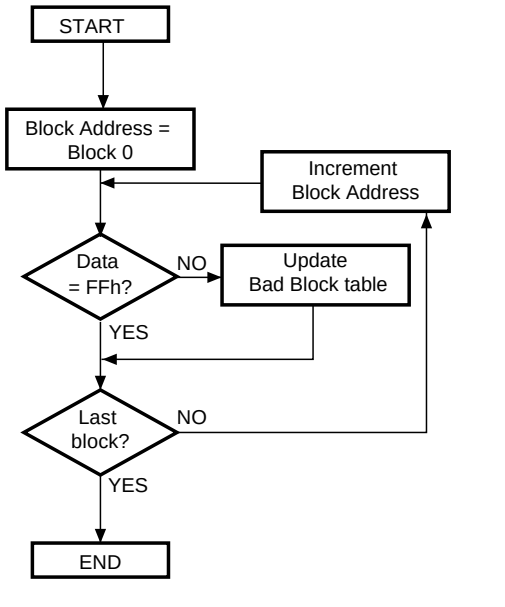
For Large Page (2112 Byte/1056 Word Page) NANDxxx-B devices, any block, where the 1st and 6th Bytes, or 1st Word, in the spare area of the 1st page, does not contain FFh, is a Bad Block.

The Bad Block Information must be read before any erase is attempted, because the Bad Block information is erasable and cannot be recovered once erased. It is highly recommended not to erase the original Bad Block information. To allow the system to recognize the Bad Blocks based on the original information, it is recommended to implement the Bad Block Management algorithm shown in Figure 2.

The Bad Block Table is created by reading all the spare areas in the NAND Flash memory. The Bad Block recognition methods that build the Bad Block table without using the original Bad Block information provided in the spare areas of the memory are not equally effective. The invalid blocks are detected at the factory during the testing process which involves severe environmental conditions and program/erase cycles as well as proprietary test modes. The failures that affect invalid blocks may not all be recognized if methods different from those implemented in the factory are used.

Once created, the Bad Block table is saved to a good block so that on rebooting the NAND Flash memory, the Bad Block Table is loaded into RAM. The blocks contained in the Bad Block Table are not addressable. So, if the Flash Translation Layer addresses one of the Bad Blocks, the Bad Block Management software redirects it to a good block.

Figure 2. Bad Block Management Flowchart



BLOCK REPLACEMENT

During the lifetime of the NAND device additional Bad Blocks may develop. The NAND devices have a Status Register that indicates whether an operation is successful or not. Additional Bad Blocks are identified when attempts to program or erase give errors in the Status Register.

As the failure of a page program operation does not affect the data in other pages in the same block, the block can be replaced by re-programming the current data and copying the rest of the replaced block to an available valid block.

Blocks can be marked as bad and new blocks allocated using two general methods.

Skip Block Method

In the Skip Block method the algorithm creates the Bad Block Table and when the target address corresponds to a Bad Block address, the data is stored in the next good block, skipping the Bad Block.

When a Bad Block is generated during the lifetime of the NAND Flash device, it's data is also stored in the next good block. In this case, the information that indicates which good block corresponds to each developed Bad Block, also has to be stored in the NAND Flash device.

If the Flash Translation Layer writes the logical sector using the Linked List method (see *AN1820*) it will not mark the whole block as bad, but only skip the bad page. With this method the Flash Translation Layer writes the logical sector to a new physical block, marking the old page invalid (and not the entire block).

Reserve Block Method

In the Reserve Block method, the Bad Block Table is created in the same manner as described in Figure 2.. In this method Bad Blocks are not skipped but replaced by good blocks by “re-directing” the Flash Translation Layer to a known good block. For that purpose, the Bad Block Management software creates two areas in the NAND Flash: the User Addressable Block Area and the Reserved Block Area as shown in Figure 3..

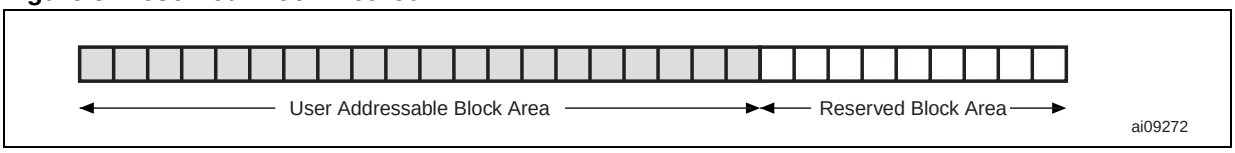
The Flash Translation Layer can use the **User Addressable Block Area** to store data whereas the **Reserved Block Area** is only used for Bad Block replacement and to save the Bad Block Table that also keeps track of the re-mapped “developed Bad Blocks”.

To define these two areas, it is necessary to determine the start address and the size of the Reserved Area. The size may either be given by the user, or imposed by the Bad Block Management software (as stated in the *NANDxxx-A, 528 Byte/ 264 Word Page* and *NANDxxx-B, 2112 Byte/1056 Word page datasheets*, for ST NAND Flash devices, the Reserved Area size is equal to 2%). For example in the 512Mb, 528 Byte page NAND Flash device, the User Addressable Block Area is 4016 blocks and the Reserved Block Area is from block 4017 to block 4096.

Each time the Flash Translation Layer Writes a logical sector, it calculates the physical address of the block to which it will write. Then, before the Flash Translation Layer starts writing, the Bad Block Management software checks whether the block is bad or not. If it is bad, it returns the address of the good block to which the sector is re-mapped. If the block becomes bad during the NAND Flash lifetime, the Bad Block Management software re-maps the Bad Block, and copies the data it contains to the block that will replace it.

The Bad Block management is completely transparent to the Flash Translation Layer. For the FTL, it is as if the data are written to the same address.

Figure 3. Reserved Block Method



CONCLUSION

To detect factory produced Bad Blocks, and manage any Bad Blocks that may develop over the lifetime of a NAND Flash device, ST recommends to use the Bad Block Management module implemented in the Hardware Adaptation Layer software provided by ST. By using this software, the data contained in a Bad Block are not lost, but are recovered and saved in a good block.

REFERENCES

- NAND128-A, NAND256-A, NAND512-A, NAND01G-A, 528 Byte/ 264 Word Page datasheet
- NAND512-B, NAND01G-B, NAND02G-B, NAND04G-B, NAND08G-B, 2112Byte/1056 Word Page datasheet
- AN1820 How to Use the FTL and HAL Software Modules to Manage Data in Single Level Cell NAND Flash Memories
- AN1821 Garbage Collection in Single Level Cell NAND Flash Memories
- AN1822 Wear Leveling in Single Level Cell NAND Flash Memories
- AN1823 Error Correction Code in Single Level Cell NAND Flash Memories

REVISION HISTORY

Table 1. Document Revision History

Date	Version	Revision Details
19-May-2004	1.0	First issue.
13-Dec-2004	2.0	The application note also applies to the NANDxxx-B Family of Single Level Cell NAND Flash memories (see Table 1., Product List). RECOGNIZING BAD BLOCKS and BLOCK REPLACEMENT sections revised.
16-Jan-2006	3	Product list removed.

If you have any questions or suggestions concerning the matters raised in this document, please refer to the MPG request support web page:

http://www.st.com/askmemory