swissbit®

Application Note

AN2101en

Garbage Collection

Contents

1 Abstract

Unlike hard drives or NOR flash, NAND flash does not have a fixed mapping of logical memory addresses to physical memory addresses. The assignment takes place via mapping tables, which are managed by the firmware of the storage medium. The flash memory itself consists of several thousand blocks, and these in turn are made up of "pages". A block is the smallest unit that can be deleted in one operation. A page is the smallest unit that can be programmed in one operation. This split means that outdated pages cannot easily be used for new data if there is still valid data in the same block. The clear up process of being able to release entire blocks is called "garbage collection".

2 Example

If the storage medium is written, the pages of the free blocks are filled sequentially. It does not matter which logical addresses (LBA) the write accesses are made to. The relationship between LBAs and block or page number is logged in the mapping tables, which are also stored in flash blocks.

The figure 1 shows an example. For simplicity's sake, it is assumed that the whole flash consists of only four blocks of three pages each, and each page contains only one LBA. Furthermore, the internal management data (mapping tables) are not shown. In this example, the storage medium has already been partially filled with a boot image, so that five pages are already occupied.

The host now writes to the following LBA: 7, 4, 7, 4, 7. The result is shown in figure 2. The first and second entries of LBA 7 and the first entry of LBA 4 are now obsolete (crossed out), as they have already been replaced by newer data.

However, since only whole blocks can be deleted, these pages can not be released and programmed again immediately. Now only a single free block is available, which now requires the garbage collector to run before the storage medium can accept further data. This is shown in figure 3: The valid pages of block 1 and block 2 are copied to block 3 by the garbage collector.

This releases blocks 1 and 2 and then deletes them in figure 4. Now there are again two free blocks available.

3 Operation

The garbage collector usually runs in the background. If the number of free blocks falls below a threshold value, it becomes active as soon as there are no more read and write accesses. He can also be interrupted again at any time. However, if the storage medium is under continuous load, the garbage collector will be activated if the number of free blocks reaches a critical value, as in the example shown in section 2. Early activation of the garbage collector in the background ensures that there are usually enough free blocks available to store even larger amounts of data without reducing the transfer speed due to the running garbage collector.

If the garbage collector is active, it searches for the blocks with the highest garbage collection efficiency. This indicates how flash-friendly stale data can be released from this block:

$$E_{gc,block} = \frac{Number\ of\ outdated\ pages\ per\ block}{Number\ of\ all\ pages\ per\ block}$$

If $E_{gc,block} = 1$, then no pages need to be copied and the block can be deleted directly.

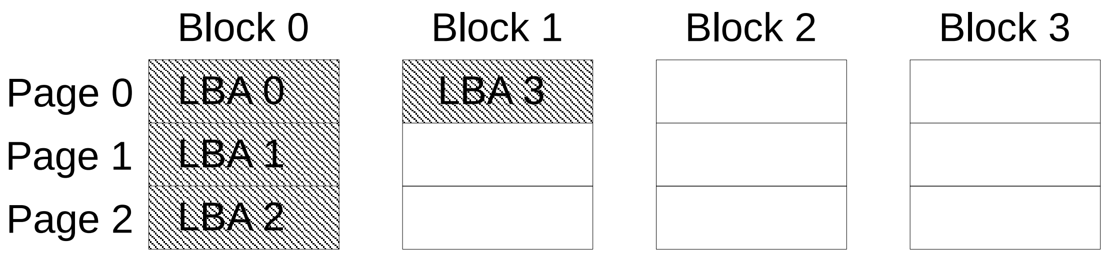

Figure 1: Four pages written

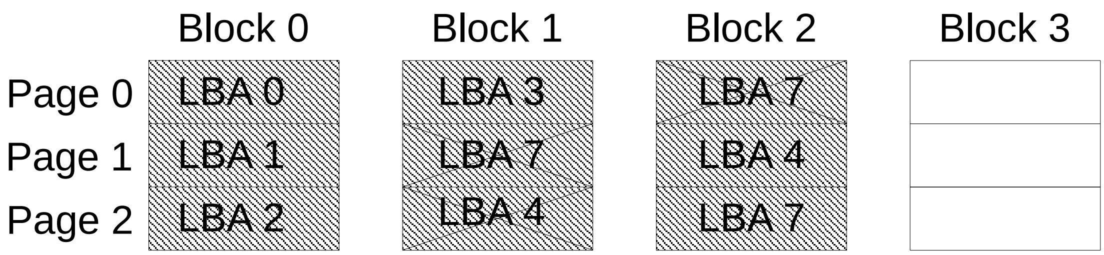

Figure 2: After write access to the LBA 7, 4, 7, 4, 7

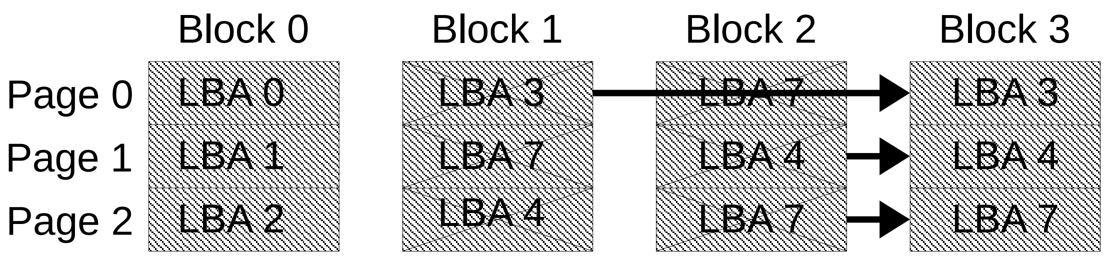

Figure 3: Copying of the still valid LBAs into a new block

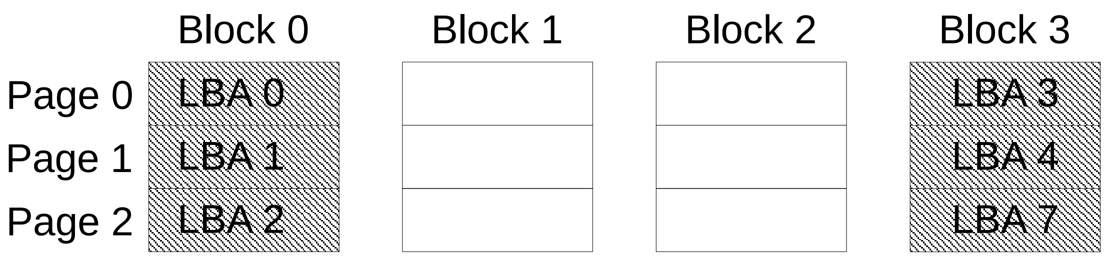

Figure 4: Deletion of the now free blocks

If $E_{gc,block} < 1$, pages must be copied before the block is deleted. This leads to a greater wear-out of the flash (see also: Write Amplification Factor). Blocks with descending $E_{gc,block}$ are released until enough free blocks are available again.

4 Over-provisioning

When setting the threshold for free blocks, there must always be a compromise between high write speed and greater wear-out. This problem can be alleviated by increasing the so-called "over-provisioning". This reduces the visible capacity of the storage medium and the ratio of visible capacity to physical capacity. It results in increasing of the average $E_{gc,block}$ and decreasing wear-out.

Figure 5 shows the usage of physically available memory. This is a purely quantitative distribution – the three groups ("user data", "management data" and "over-provisioning") are not allocated to fixed memory areas. The physical memory forms a pool from which each group can obtain blocks.

Most of it is available for payload, a small part is needed for the internal management data (eg. mapping tables), and a share configurable by the manufacturer of the storage medium is over-provisioning. The over-provisioning can not be arbitrarily small, since a minimum of free memory must be available to the garbage collector. Applications that do not sequentially write to the medium but mainly execute random write access benefit from a large overprovisioning.

Furthermore, reserve blocks are also obtained from this memory if, during the lifetime, flash blocks have to be replaced because of increased bit error numbers ("grown bad blocks").

5 Trim

If each logical address has been written at least once, e.g. because the storage medium has been completely filled, then in the mapping tables every physical memory address has a logical address assigned to it. The number of free and obsolete pages now available to the garbage collector is determined by the size of the over-provisioning. If the over-provisioning is low in favor of a high user capacity, then $E_{gc,block}$ is correspondingly low and the write amplification factor increases.

If files are deleted in the file system in this state, the situation in the flash memory does not change since the storage medium has no knowledge of the deleted files, and the pages belonging to the deleted files are still linked in the mapping tables with the old logical addresses of the files. This is where the "Trim" of modern operating systems helps: After deleting a file, the Trim command is sent to the storage medium. This command transfers all logical memory addresses that were occupied by the deleted file. The corresponding entries in the mapping tables are now marked as obsolete. The average $E_{gc,block}$ will be improved and the garbage collector will find blocks with $E_{gc,block} = 1$, depending on the size and number of files deleted.

Trim is supported by all modern operating systems and is typically enabled by default. It is supported in Microsoft Windows since Windows 7, in GNU/Linux it has been implemented beginning with kernel 2.6.28 for various file systems, and is available for all popular file systems since kernel 3.0. There are two variants available for GNU/Linux:

- At **batched-discard** the command "fstrim" is executed periodically, which reports all unused areas to the storage medium.
- At **online-discard** the storage medium is immediately informed when areas become free. This is activated with the mount option "discard".

Trim is part of the ATA specification and is therefore only available for storage media with a SATA interface. Storage media with PCIe interface as well as eMMC have a comparable command.

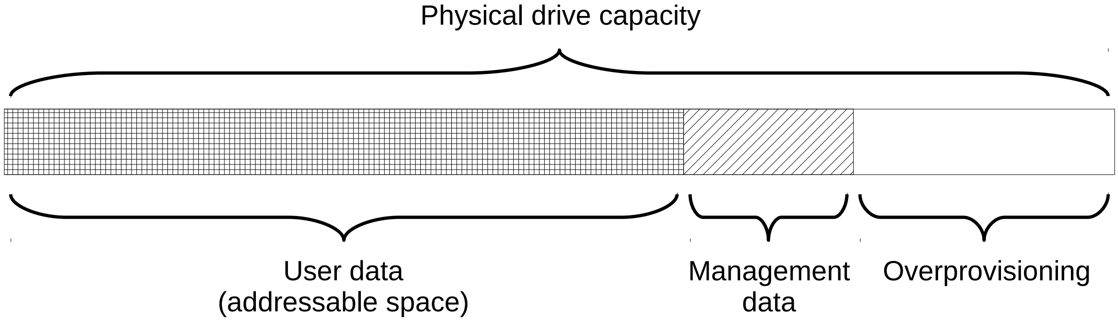

Figure 5: Over-provisioning (not to scale)

CONTACT US

Headquarters	**Swissbit AG** Industriestrasse 4 9552 Bronschhofen Switzerland	Tel. +41 71 913 03 03 sales@swissbit.com
Germany (Berlin)	**Swissbit Germany AG** Bitterfelder Strasse 22 12681 Berlin Germany	Tel. +49 30 936 954 0 sales@swissbit.com
Germany (Munich)	**Swissbit Germany AG** Leuchtenbergring 3 81677 Munich Germany	Tel. +49 30 936 954 400 sales@swissbit.com
North and South America	**Swissbit NA Inc.** 238 Littleton Road, Suite 202B Westford, MA 01886 USA	Tel. +1 978-490-3252 salesna@swissbit.com
Japan	**Swissbit Japan Co., Ltd.** CONCIERIA Tower West 2F 6-20-7 Nishishinjuku Shinjuku City, Tokyo 160-0023 Japan	Tel. +81 3 6258 0521 sales-japan@swissbit.com
Taiwan	**Swissbit Taiwan** 3F., No. 501, Sec.2, Tiding Blvd. Neihu District, Taipei City 114 Taiwan, R.O.C.	Tel. +886 912 059 197 salesasia@swissbit.com
China	**Swissbit China**	Tel. +886 958 922 333 salesasia@swissbit.com

Disclaimer:

The information in this document is subject to change without notice. Swissbit AG ("SWISSBIT") assumes no responsibility for any errors or omissions that may appear in this document, and disclaims responsibility for any consequences resulting from the use of the information set forth herein. SWISSBIT makes no commitments to update or to keep current information contained in this document. The products listed in this document are not suitable for use in applications such as, but not limited to, aircraft control systems, aerospace equipment, submarine cables, nuclear reactor control systems and life support systems. Moreover, SWISSBIT does not recommend or approve the use of any of its products in life support devices or systems or in any application where failure could result in injury or death. If a customer wishes to use SWISSBIT products in applications not intended by SWISSBIT, said customer must contact an authorized SWISSBIT representative to determine SWISSBIT willingness to support a given application. The information set forth in this document does not convey any license under the copyrights, patent rights, trademarks or other intellectual property rights claimed and owned by SWISSBIT.

ALL PRODUCTS SOLD BY SWISSBIT ARE COVERED BY THE PROVISIONS APPEARING IN SWISSBIT'S TERMS AND CONDITIONS OF SALE ONLY, INCLUDING THE LIMITATIONS OF LIABILITY, WARRANTY AND INFRINGEMENT PROVISIONS. SWISSBIT MAKES NO WARRANTIES OF ANY KIND, EXPRESS, STATUTORY, IMPLIED OR OTHERWISE, REGARDING INFORMATION SET FORTH HEREIN OR REGARDING THE FREEDOM OF THE DESCRIBED PRODUCTS FROM INTELLECTUAL PROPERTY INFRINGEMENT, AND EXPRESSLY DISCLAIMS ANY SUCH WARRANTIES INCLUDING WITHOUT LIMITATION ANY EXPRESS, STATUTORY OR IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.