swissbit®

Application Note

AN2103en

Performance tests

[1]This work is licensed under the Creative Commons License "Attribution 4.0 International". To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/

Contents

1 Abstract

This paper examines the speed of writing and reading NAND flash storage media in relation to the type of access pattern and the influence of the previously written pattern.

This will provide some guidance for selecting the most suitable methods of testing user-specific requirements for the storage medium. For a better understanding of this paper, the basic functions of the garbage collector should be known.

2 Framework

Unlike hard drives, NAND flash storage media often cannot meet millisecond real-time requirements when writing. This is primarily the result of the higher-priority garbage collector, whose load is difficult to predict. Therefore, hard real-time needs always require a correspondingly large cache in the operating system.

While hard drives have fixed mapping between the physical addresses of the memory and logical addresses used by the operating system, this connection does not exist with NAND Flash storage media. If multiple writes to the same logical address occur, the data is written to a different physical address every time. The relationship between logical and physical addresses is managed by the firmware in tables that are also stored in the NAND flash.

The type of data written to the storage medium during a test does not affect speed. Controllers with built-in data compression were used in the early days of NAND flash, but they are no longer common. Accordingly, the high data-processing requirement for the generation of pseudo-random data, which could adversely affect speed measurement, can be dispensed of.

3 Write speed

Figure 1 shows the write speed in relation to the type of access and the existing data distribution on the storage medium. Here, the data volume of the nominal capacity is written to the storage medium seven times and the access method is changed for each cycle. The illustration shows a standard storage medium with "page-based mapping" without DRAM cache and without pSLC cache. The seven sections are explained as follows:

Section ①: 4 KiB Random Write

At the beginning of section 1 the storage medium is empty. It has never been written to, is completely trimmed or has been completely deleted ("secure-erased"). Accordingly, the physical addresses are not assigned to any logical addresses – the mapping tables contain no entries. As the storage medium begins to be filled with random write accesses (4 KiB Random-Write) each logical address is written exactly once. Due to the small amount of data per write access and the high administrative overhead when tracking the allocation tables, the speed is severely limited. The speed stays constant over the entire time.

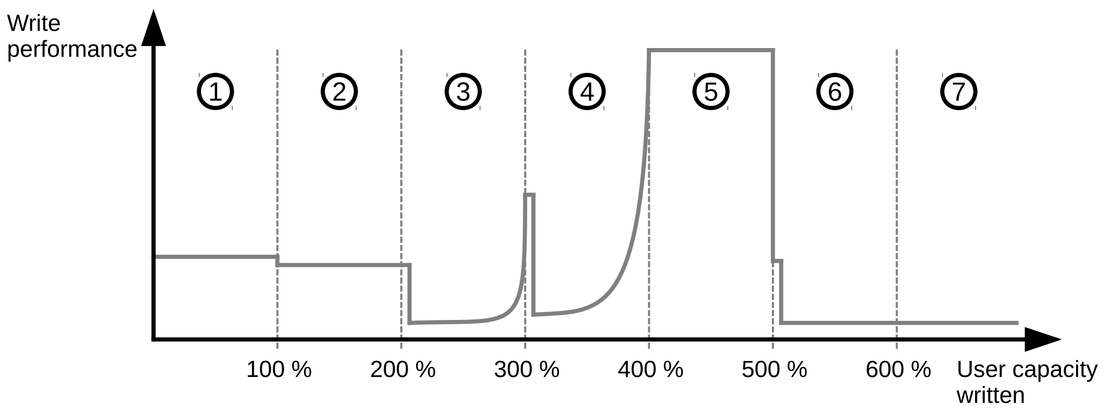

Figure 1: Write speed

Section ②: 4 KiB Random-Write, same address sequence

In section 2, the same write accesses from section 1 are repeated: the sequence of addresses written is identical.

A slight reduction in write speed can be observed. This is due to the mapping tables being completely filled and the old entry having to be invalidated for each newly written address. Because of the identical address sequence, there is no pressure on the garbage collector, because with each newly filled flash block another block is automatically released again. When a block is completely filled, a second block exists with the same logical addresses, which are now obsolete, whereby the second block can be deleted.

Within a datasheet the speed of this section is often shown as "sustained random write". However, this can be misleading by suggesting this is the minimum achievable write speed. The minimum write speed is only reached at maximum load on the garbage collector. This is covered in the following sections.

Careful consideration must be given to the minimal speed requirements of the respective application.

Section ③: 4 KiB Random-Write, new address sequence

After filling the storage medium twice with the same sequence of random write accesses, the address sequence is changed at the start of section 3. Even after writing a small amount of data, the speed drops significantly: "over-provisioning" has been used up and the previously inactive garbage collector, now works under full load. By changing the address sequence, blocks are no longer automatically freed, and the garbage collector must constantly copy data to gain free blocks for new data.

Since each address is still written exactly once in each section, the write speed increases exponentially shortly before the end of the section, as – with the shrinking, as yet missing amount of addresses – with each newly filled block, blocks are freed-up again automatically. When the last block of this section is written to, all blocks from the over-provisioning are automatically freed-up again.

Section ④: 128 KiB Sequential-Write

At the beginning of section 4, a random address sequence is now switched to a sequential address sequence. In addition, the amount of data per access is increased from 4 KiB to 128 KiB. This increase immediately results in higher throughput through greater efficiency of transfer between host and storage medium. After over-provisioning is used up, the write speed drops to almost the same value as in section 3 because all the old data was written with random addresses and now the load on the garbage collector is just as high.

Since the garbage collector sorts the addresses as far as possible when moving data, the speed increase occurs earlier with decreasing address quantity and when blocks automatically become free again. Consequently, write-speed increases at a faster rate than in section 3 because of the greater access size of 128 KiB.

Section ⑤: 128 KiB Sequential-Write, second run

In the previous section, the storage medium was filled sequentially. The addresses of the data in the flash blocks show a strictly monotonous increase in each block. All blocks from the over-provisioning are freed up again, since again each address has been written exactly once. In section 5, the storage medium is filled sequentially just as in section 4. Since the address sequence is the same as in the previous section, with each flash block written, another block is freed up automatically. The large amount of data per host write access makes the transfer between host and storage medium very efficient. The garbage collector is load-free, and owing to the sequential write accesses, the tracking of the allocation tables also generates little additional load. Here, the maximum write speed of the storage medium is reached. Only an empty storage medium (after trim or secure erase) could still achieve a slightly higher speed.

Sections ⑥ and ⑦: 4 KiB Random-Write, complete random address

At the beginning of section 6, random address sequences are chosen again. In contrast to sections 1–3, where each address has been written exactly once, the entire address space is now open for each new address. Accordingly, in both sections, on the one hand, not all logical addresses are necessarily rewritten, and on the other hand, several addresses are written several times. After filling the initial over-provisioning, the garbage collector is under full load. Since a flash block is never automatically freed, the garbage collector remains permanently under full load. This operating state is now the true sustained random write, which achieves the minimum write speed. However, this is unlikely to reflect a practical application, especially as the write amplification factor is so high that the storage medium would very quickly reach its specified number of write and erase cycles.

At the beginning of section 6, random address sequences are once again chosen. In contrast to sections 1–3, where each address has been written exactly once, the entire address space is now open for each new address. Accordingly, in both sections, on the one hand, not all logical addresses are necessarily rewritten, and on the other hand, several addresses are written several times. After filling the initial overprovisioning, the garbage collector is under full load. Since a flash block is never automatically freed, the garbage collector permanently remains under full load. This operating state is now the true sustained-random-write, which achieves the minimum write speed. However, this is unlikely to reflect a practical application, especially as the write amplification factor is so high that the storage medium would very quickly reach its specified number of write and erase cycles.

4 Pseudo-SLC-cache

To increase write speed, the “pseudo SLC cache”, commonly known from consumer markets, was also introduced for industrial storage solutions. In this case, part of the NAND capacity is configured as SLC memory in which only one bit per cell is stored. Accordingly, this memory can be written and read very fast. Since it is not dedicated, real SLC memory, it is called pseudo SLC (pSLC). Such a cache can be used for all memory types that store several bits per flash cell (MLC, TLC, QLC). For the following examples, TLC (three bits per cell) is assumed.

Figure 2 shows the typical write speed of a storage medium with a pseudo SLC cache.

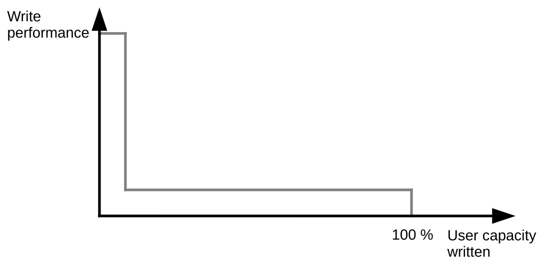

Figure 2: Sequential-Write with pSLC-Cache

When the fast pSLC-cache is full, the speed drops significantly, as further write access to the storage medium requires to free up the pSLC by moving older data from the cache to the TLC memory.

The use of pSLC-cache offers a speed advantage, especially when the storage medium is not busy with read or write accesses between the writing of larger amounts of data. These idle times are used by the storage medium to move data from the cache to the TLC area.

A pSLC cache offers even more advantages:

- It can improve power-fail behavior because a power cut during a write process to the pSLC cache cannot corrupt older data since it stores only one bit per cell. If the power fail occurs while moving data to the TLC memory, the data is still available in the cache.
- It improves data security when “bad blocks” occur, especially with 3D NAND and it reduces the complexity and RAM requirements of the controller.

When qualifying a NAND flash memory medium, a pSLC cache should be considered, as non-stop writing of large amounts of data can lead to a massive drop in write performance. A suitable test is the sequential filling of the storage medium to 100 % without interruption as shown in Figure 2.

Dynamic-pSLC-cache

Dynamic pSLC cache has also found its way into industrial storage solutions. In contrast to the static pSLC cache in Figure 2, up to 100 % of the NAND flash is used dynamically as a pSLC cache, depending on how full the storage medium is. The following must be noted:

- The write speed of the storage medium depends not only on the amount of data written without interruption but also on the fill-level of the memory. This makes write speed difficult to predict.
- Dynamically changing the configuration of flash blocks as pSLC or TLC memory is discouraged by NAND flash manufacturers for reliability reasons, especially in the industrial temperature range. A maximum of one change from pSLC to MLC is allowed; but this must be done within 1 % of the specified pSLC cycles.
- All manufacturers of NAND dynamic storage media change to static cache after a few program and erase cycles. Before that, the storage medium achieves best values with short speed tests that do not fill the entire capacity. After a short period of use, the medium slows down permanently.

Figure 3 shows the behavior of a dynamic cache TLC storage device at the beginning of life (solid line) and after 10 % of the specified cycles (dashed line) at the latest.

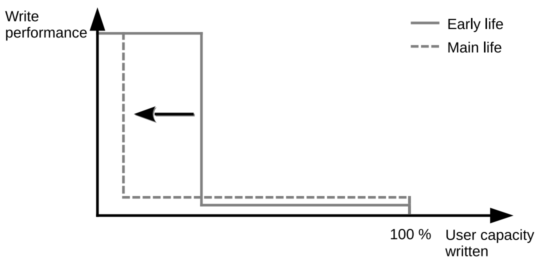

Figure 3: Sudden decrease in write speed with dynamic pSLC-Cache

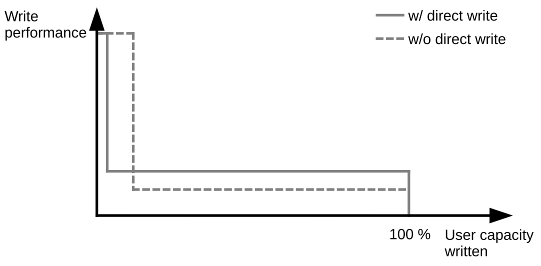

Figure 4: Switch to direct-write when detecting sequential write

5 Direct-Write

"direct write" is described as the bypass for a pSLC cache. This can be implemented in different cases:

- If there is continuous write access to the storage medium, and this happens mostly sequentially, it can be assumed that the pSLC cache would be running out of space in a short time. In this case some industrial storage devices bypass the pSLC cache and write directly to the TLC memory. This reduces wearout of the pSLC cache and increases write speed for very large amounts of data since the data does not have to be written twice. A disadvantage, however, is reduced safety against power failures. Since large amounts of data typically only occur during the production of a system when the image with the operating system is installed, this disadvantage is usually acceptable. In contrast, there is the advantage of faster installation. Figure 4 shows the difference in sequential writing with and without direct write.

- In combination with a dynamic pSLC cache, direct write is used with the aim of delaying the dissolving and reduction of an enlarged pSLC cache (during a speed test) as long as possible. Figure 5 shows one such case: The storage medium is filled sequentially and has a dynamic pSLC cache with a dynamic portion that occupies about 3/4 of the physical NAND memory at the beginning of the test, which equals 1/4 of the nominal capacity at TLC NAND. When at the end of section 1 the pSLC cache is completely full, the storage medium in section 2 changes to the slower direct write. After subtracting the pSLC cache, 1/4 of the nominal capacity remains as TLC memory for direct write. When this is also full, the physical memory is exhausted:

  - 3/4 of the physical TLC capacity is used as pSLC memory but can only store 1/4 of the nominal capacity.

  - 1/4 of the physical capacity is still used as TLC memory and can therefore also store 1/4 of the nominal capacity.

  Now the storage medium in section 3 must convert the dynamic part of the pSLC cache into TLC memory in order to get free physical memory again. The write speed is correspondingly low.

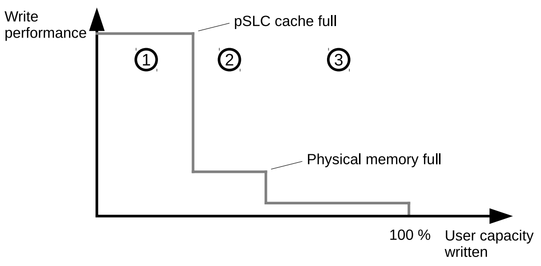

Figure 5: Direct-Write with dynamic pSLC-Cache

6 Read Performance

Read performance is not dependent on as many factors as write speed. It depends only on the type of access used to write and how it is read. Figure 6 illustrates this:

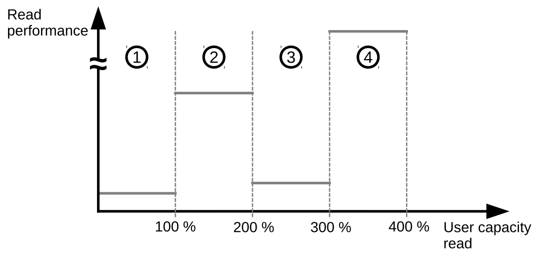

Figure 6: Read Performance

The data in section 1 was randomly written and read again at random, but in a different order. Here, the lowest read performance is achieved because the search effort in the allocation tables is very high and the efficiency of the data transfer between storage medium and host is low because of the packet size of 4 KiB.

In section 2, the randomly written data is sequentially read. The read performance is much higher, since 128 KiB are now transferred to the host per access, which increases efficiency. The search effort in the allocation tables is still very high.

Before section 3 the storage medium was written sequentially. The read accesses in this section were random. The read performance is correspondingly low due to the package size of 4 KiB. However, sequential write makes searching in mapping tables easier and faster than in section 1.

In section 4, the storage medium reaches the maximum read performance. The sequentially written data is now also read sequentially in 128 KiB packets. The search in the allocation tables is very simple, the transfer to the host very efficient and internal access to the NAND flash can be done using "read ahead".

For a storage medium with pSLC cache, a higher read performance would occur for some of the memory addresses in the four sections – namely, if these addresses are in the pSLC cache.

7 Summary

To compare speed specifications in data sheets of NAND flash memory media, the prevailing test conditions must be identical. If measurements were not repeatedly taken or did not cover the entire logical address space, the specified speed can be significantly higher than the real achievable speed. When qualifying a NAND flash memory medium for performance or time-critical applications, it is advisable to simulate maximum load demand and access patterns of the application as closely as possible in a test program and to test this over a longer period.

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