swissbit®

Application Note

AN2101de

Garbage Collection

[1] Dieses Werk steht unter der Creative-Commons-Lizenz „Namensnennung 4.0 International“. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie http://creativecommons.org/licenses/by/4.0/

swissbit®

Inhaltsverzeichnis

1 Einleitung

Im Gegensatz zu Festplatten oder NOR-Flash gibt es bei NAND-Flash keine feste Zuordnung von logischen Speicheradressen zu physikalischen Speicheradressen. Die Zuordnung erfolgt über Mapping-Tabellen, die von der Firmware des Speichermediums verwaltet werden. Der Flash-Speicher selbst besteht aus mehreren tausend Blöcken und diese wiederum aus „Pages". Ein Block ist die kleinste Einheit, die in einem Vorgang gelöscht werden kann; eine Page ist die kleinste Einheit, die in einem Vorgang programmiert werden kann. Diese Aufteilung führt dazu, dass Speicherbereiche mit veralteten Daten nicht einfach für neue Daten verwendet werden können, wenn sich im selben Block noch gültige Daten befinden. Den Aufräum-Vorgang, um ganze Blöcke freigeben zu können, nennt man „Garbage-Collection".

2 Beispiel

Wird das Speichermedium beschrieben, so werden die Pages der freien Blöcke sequentiell gefüllt. Dabei spielt es keine Rolle, auf welche logischen Adressen (LBA) die Schreibzugriffe erfolgen. Der Zusammenhang zwischen LBA und Block- bzw. Page-Nummer wird in den Mapping-Tabellen protokolliert, die ebenfalls in Flash-Blöcken abgelegt werden.

Die Abbildung 1 zeigt ein Beispiel. Zur Vereinfachung wird angenommen, dass der gesamte Flash nur aus vier Blöcken mit je drei Pages besteht, und jede Page nur eine LBA enthält. Weiterhin sind die internen Verwaltungsdaten (Mapping-Tables) nicht gezeigt. In diesem Beispiel wurde das Speichermedium bereits teilweise mit einem Boot-Image gefüllt, so dass vier Pages bereits belegt sind.

Der Host schreibt nun auf folgende LBA: 7, 4, 7, 4, 7. Das Resultat zeigt Abbildung 2. Der erste und zweite Eintrag von LBA 7 und der erste Eintrag von LBA 4 sind nun veraltet (durchgestrichen), da sie bereits durch neuere Daten ersetzt wurden.

Da sich aber nur ganze Blöcke löschen lassen, können diese Pages nicht direkt wieder verwendet werden, d. h. sie können nicht sofort neu beschrieben werden. Jetzt ist nur noch ein einziger freier Block verfügbar, wodurch nun zwingend der Garbage-Collector laufen muss, bevor das Speichermedium weitere Daten annehmen kann. Dies zeigt Abbildung 3: Die gültigen Pages von Block 1 und Block 2 werden vom Garbage-Collector nach Block 3 kopiert.

Dadurch werden die Blöcke 1 und 2 wieder frei und anschließend in Abbildung 4 gelöscht. Nun stehen wieder zwei freie Blöcke zur Verfügung.

3 Funktionsweise

Der Garbage-Collector läuft meistens im Hintergrund. Wenn die Anzahl der freien Blöcke einen Schwellwert unterschreitet, wird er aktiv, sobald keine Lese- und Schreibzugriffe mehr erfolgen. Er kann auch jederzeit wieder unterbrochen werden. Ist das Speichermedium jedoch unter Dauerlast, so wird der Garbage-Collector zwingend aktiviert, wenn die Anzahl der freien Blöcke einen kritischen Wert erreicht wie im gezeigten Beispiel in Abschnitt 2. Durch die frühzeitige Aktivierung des Garbage-Collectors im Hintergrund wird erreicht, dass meistens genug freie Blöcke zur Verfügung stehen, um auch größere Datenmengen speichern zu können, ohne die Übertragungsgeschwindigkeit wegen dem laufenden Garbage-Collector zu reduzieren.

Ist der Garbage-Collector aktiv, so sucht er nach den Blöcken mit der höchsten „Garbage-Collection-Efficiency". Diese gibt an, wie flash-

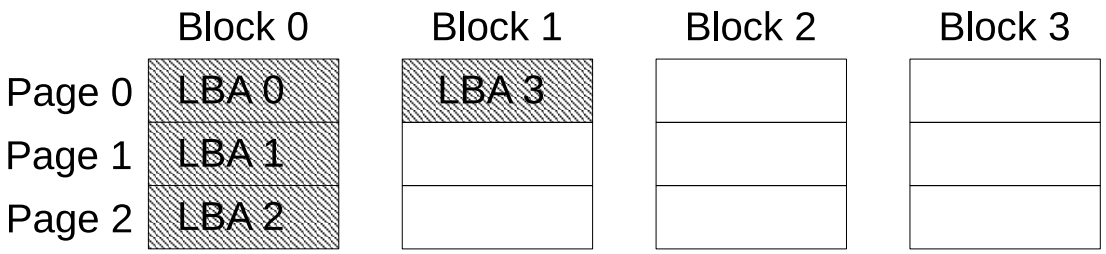

Abbildung 1: Vier Pages beschrieben

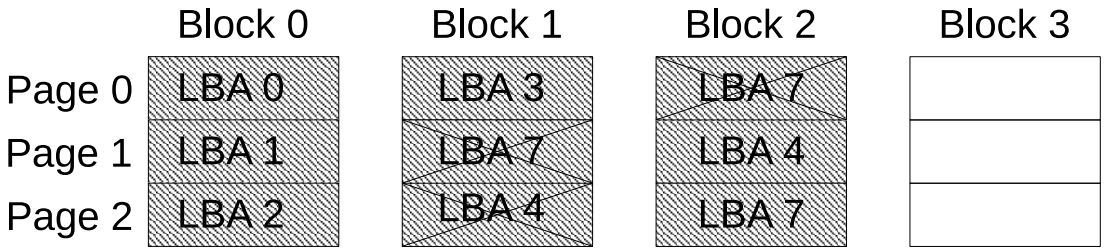

Abbildung 2: Nach Schreibzugriffen auf die LBA 7, 4, 7, 4, 7

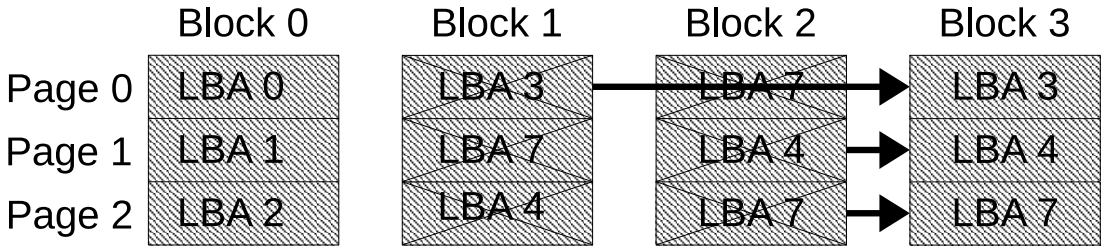

Abbildung 3: Umkopieren der noch gültigen LBA in einen neuen Block

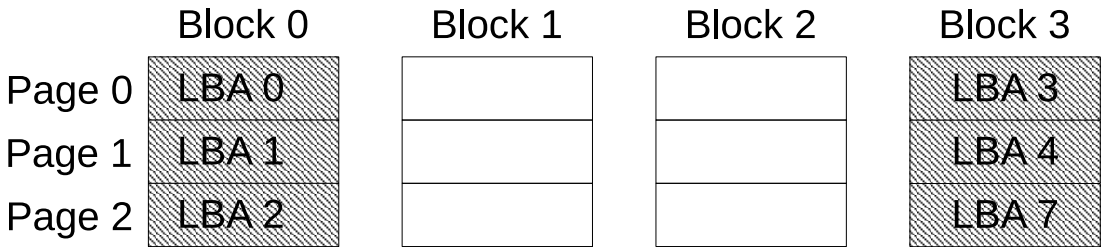

Abbildung 4: Löschen der nun freien Blöcke

freundlich veraltete Daten aus diesem Block freigegeben werden können:

$$E_{gc,block} = \frac{Anzahl\ veralteter\ Pages\ pro\ Block}{Anzahl\ aller\ Pages\ pro\ Block}$$

Ist $E_{gc,block} = 1$, dann müssen keine Pages kopiert werden und der Block kann direkt gelöscht werden. Ist $E_{gc,block} < 1$, so müssen vor dem Löschen des Blocks noch Pages umkopiert werden. Dadurch kommt es zu einer stärke-

ren Abnutzung des Flashes (siehe auch: Write-Amplification-Factor). Es werden solange Blöcke mit absteigender $E_{gc,block}$ freigegeben, bis wieder ausreichend freie Blöcke zur Verfügung stehen.

4 Overprovisioning

Bei der Einstellung des Schwellwertes für freie Blöcke muss dabei immer ein Kompromiss zwischen hoher Schreibgeschwindigkeit und stärkerer Abnutzung eingegangen werden. Diese Problematik lässt sich durch das Vergrößern des sogenannten „Overprovisioning" entschärfen. Hierbei wird die sichtbare Kapazität des Speichermediums reduziert; es verringert sich somit das Verhältnis von sichtbarer Kapazität zu physikalischer Kapazität. Dadurch erhöht sich die durchschnittliche $E_{gc,block}$, und die Abnutzung sinkt.

Abbildung 5 zeigt die Aufteilung des physikalisch verfügbaren Speichers. Dabei handelt es sich um eine rein quantitative Aufteilung – die drei Gruppen („User data", „Management data" und „Overprovisioning") sind keinen festen Speicherbereichen zugeordnet. Der physikalische Speicher bildet einen Pool, aus dem jede Gruppe Blöcke beziehen kann.

Der größte Teil steht für Nutzdaten zur Verfügung, ein kleiner Teil wird für die internen Verwaltungsdaten (z. B. Mapping-Tabellen) benötigt, und ein vom Hersteller des Speichermediums konfigurierbarer Anteil bildet das Overprovisioning. Das Overprovisioning kann nicht beliebig klein sein, da ein Minimum an freiem Speicher dem Garbage-Collector zur Verfügung stehen muss. Von einem großen Overprovisioning profitieren besonders Anwendungen, die nicht sequentiell auf das Medium schreiben sondern hauptsächlich zufällige Schreibzugriffe ausführen.

Weiterhin werden aus diesem Speicher auch Reserveblöcke bezogen, falls während der Lebensdauer Flash-Blöcke wegen erhöhter Bitfehleranzahl ersetzt werden müssen („Grown bad blocks").

5 Trim

Wurde jede logische Adresse mindestens einmal beschrieben, z. B. weil das Speichermedium komplett gefüllt wurde, so ist in den Mapping-Tabellen jeder physikalischen Speicheradresse eine logische Adresse zugeordnet. Die Anzahl der nun dem Garbage-Collector noch zur Verfügung stehenden freien und veralteten Pages wird durch die Größe des Overprovisioning bestimmt. Ist das Overprovisioning zugunsten einer hohen Nutz-Kapazität gering, fällt nun $E_{gc,block}$ entsprechend ungünstig aus und der Write-Amplification-Factor steigt. Wenn in diesem Zustand Dateien im Dateisystem wieder gelöscht werden, ändert sich die Situation im Flash-Speicher nicht, da das Speichermedium keine Kenntnis von den gelöschten Dateien hat und die zu den gelöschten Dateien gehörenden Pages immer noch in den Mapping-Tabellen mit den alten logischen Adressen der Dateien verknüpft sind. Hier hilft das „Trim" moderner Betriebssysteme: Nach dem Löschen einer Datei wird das Trim-Kommando an das Speichermedium geschickt. Mit diesem Kommando werden sämtliche logische Speicheradressen übertragen, die die gelöschte Datei belegte. Die entsprechenden Einträge in den Mapping-Tabellen werden nun als veraltet gekennzeichnet. Die durchschnittliche $E_{gc,block}$ wird verbessert, und der Garbage-Collector wird – je nach Größe und Anzahl der gelöschten Dateien – Blöcke mit $E_{gc,block} = 1$ finden.

Trim wird von allen modernen Betriebssystemen unterstützt und ist typischerweise aktiviert. In Microsoft Windows wird Trim ab Windows 7 unterstützt, bei GNU/Linux wurde es ab Kernel 2.6.28 für verschiedene Dateisysteme implementiert und ist seit Kernel 3.0 für alle gängigen Dateisysteme verfügbar. Für GNU/Linux stehen dabei zwei Varianten zur Verfügung:

- Bei **batched-discard** wird periodisch das Kommando „fstrim" ausgeführt, das alle ungenutzten Bereiche an das Speichermedium meldet.
- Bei **online-discard** wird dem Speichermedium sofort mitgeteilt, wenn Bereiche frei

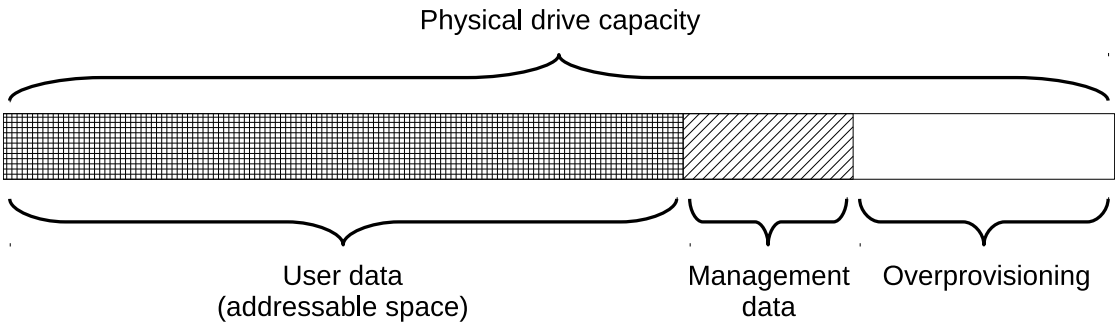

Abbildung 5: Overprovisioning (nicht maßstabsgetreu)

werden. Dies wird mit der Mount-Option „discard“ aktiviert.

Trim ist Teil der ATA-Spezifikation und ist daher nur für Speichermedien mit SATA-Schnittstelle verfügbar. Speichermedien mit PCIe-Schnittstelle sowie eMMC verfügen über einen vergleichbares Kommando.

Kontaktieren Sie uns

Hauptsitz	**Swissbit AG** Industriestraße 4 9552 Bronschhofen Schweiz	Tel. +41 71 913 03 03 sales@swissbit.com
Deutschland (Berlin)	**Swissbit Germany AG** Bitterfelder Straße 22 12681 Berlin Deutschland **Standort Wolfener Straße** Swissbit Germany AG Wolfener Straße 36 12681 Berlin Deutschland	Tel. +49 30 936 954 0 sales@swissbit.com
Deutschland (München)	**Swissbit Germany AG** Leuchtenbergring 3 81677 Mümchen Deutschland	Tel. +49 30 936 954 400 sales@swissbit.com
Nord- und Südamerika	**Swissbit NA Inc.** 238 Littelton Rd #202b Westford, MA 01886 USA	Tel. +1 978-490-3252 sales@swissbitna.com
Japan	**Swissbit Japan Co., Ltd.** 4-20-7 Asagaya-kita, Suginami-ku, Tokio, 166-0001 Japan	Tel. +81 3 5356 3511 sales@swissbit.co.jp
Taiwan	**Swissbit Taiwan** 4F-1, No 104, Sec. 2, Dunhua S. Rd. Da'an Dist, Taipei City 106 Taiwan (R.O.C.)	Tel. +886 2 27010788 salesasia@swissbit.com
China	**Swissbit China**	Tel. +86 15000 393 870 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.