

# „Böse“ Blöcke im Griff

## Programmierung und Bad-Block-Management von NAND-Flashes

**Flash-Speicherbausteine in NAND-Technik waren 2003 mit einer Wachstumsrate von über 40 % die am schnellsten wachsende Speicherchip-Sparte. Hohe Schreib- und Leseraten sind unbestritten die Vorteile dieser Chips. Trotzdem scheuen sich Entwickler, NAND-Flashes wegen des komplizierten Bad-Block-Managements einzusetzen. Standardmethoden können weiterhelfen.**

Von Edwin Musch

Im Bereich von 128 Mbit bis 1 Gbit und größer sind NAND-Flash-Speicherbausteine mittlerweile eine ernst zu nehmende Alternative zur NOR-Flash-Technologie. Zwar werden NOR-Bausteine weiterhin verwendet, der Einsatz bleibt aber aufgrund langsamer Schreib- und Löschraten sowie geringer Speicherkapazität auf einfache Applikationen beschränkt. NAND-Flashes dagegen sind ideal geeignet für Applikationen, die große Speichermengen und Schnelligkeit erfordern. Namhafte Halbleiterhersteller setzen auf NAND-Technologie beispielsweise in den Branchen Mobile

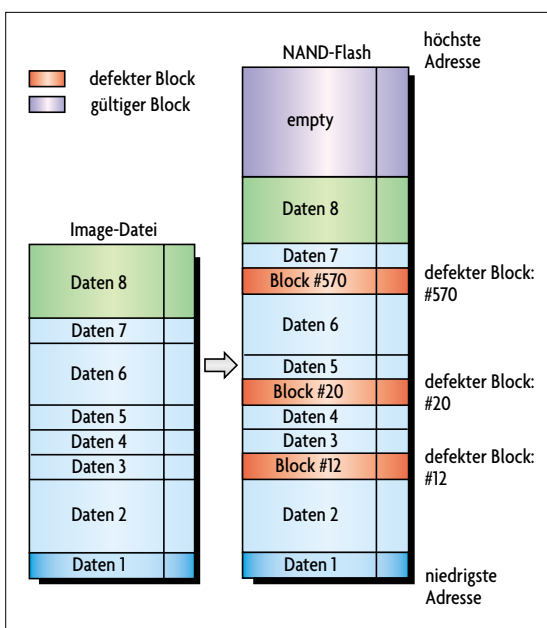
daraus resultierende Bad-Block-Handling lassen die NAND-Technologie kompliziert und schwer zu handhaben erscheinen – aber nur auf den ersten Blick, denn die Probleme sind lösbar. In enger Zusammenarbeit mit Halbleiterherstellern stellt Data I/O sichere und effiziente Standardlösungen für das Programmieren und Integrieren von NAND-Flashes zur Verfügung, die einfach und ohne Risiko genutzt werden können.

### Keine Angst vor Bad-Block-Handling

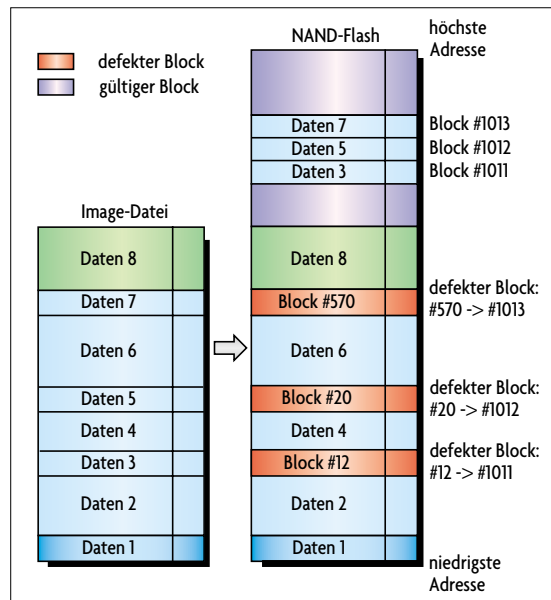
Aus Kosten- und Materialgründen sind NAND-Flash-Bausteine in Löscheinheiten, so genannte Blöcke gegliedert.

Es ist davon auszugehen, dass beim Beschreiben zufällig verteilte, defekte Blöcke – „Bad Blocks“ – auftreten können, die über einen entsprechenden Mechanismus, ähnlich dem in der Disk-Drive-Technologie, korrigiert werden müssen, wenn sie identifiziert wurden.

Um Informationen in einen Block des Speicherbausteins zu schreiben, muss gewährleistet sein, dass der Block unbeschrieben ist bzw. vorher gelöscht wurde. Beim Beschreiben eines NAND-Speicherblocks



**Bild 1. „Skip Bad Block“-Verfahren. Bei dieser Art von Bad-Block-Management werden die Daten einfach in den nächsten fehlerfreien Block geschrieben.**



**Bild 2. „Reserved Block Area“-Verfahren. Am Ende des Flash-Bausteins befindet sich ein ungenutzter, reservierter Bereich, der als Reservoir zum Ersatz defekter Speicherblöcke dient.**

Telekommunikation (Navigationssysteme), Informationstechnologie (PC-Cards) und Unterhaltungselektronik (MP3-Player, Digitalkameras).

In der Herstellung werden NAND-Flashes immer günstiger, sind zudem sehr langlebig, und ihre Blockstruktur erleichtert und beschleunigt das Schreiben und Löschen von Daten. Schnelle Schreib- und Leseraten kombiniert mit hohen Zelldichten und -kapazitäten sind die offenkundigen Vorteile dieser Bausteine.

Trotzdem scheut sich die Industrie davor, NAND-Flash-Bausteine in ihre Designs zu integrieren. Probleme mit Bit-Defekten beim Beschreiben und das

können Bit-Defekte auftreten, die den Block unbenutzbar machen. Diese Blöcke müssen bei der Programmierung identifiziert und entsprechend des Bad-Block-Handling-Verfahrens behandelt werden. Hierbei wird ein schlechter Block markiert, damit er später nicht mehr verwendet wird. Die Daten werden stattdessen in einen anderen Block geschrieben. Wo sich dieser Ersatzblock befindet und wie die Stelle wieder gefunden wird, definiert das jeweilige Bad-Block-Handling (BBH). Das bedeutet, dass das Programmiergerät den Baustein in derselben Art und Weise behandeln muss, wie es auch später in der Applikation erfolgt.

Eigene, flexibel auf die Applikation zugeschnittene BBH-Verfahren können definiert werden, sind aber sehr aufwendig und oftmals nicht ausfallsicher. Daher ist es ratsam, auf bereits getestete BBH-Verfahren zurückzugreifen. Diese sind mit Unterstützung des Bausteinherstellers schneller verfügbar. Data I/O empfiehlt zusammen mit NAND-Flash-Bausteinanbietern, bereits vorhandene und getestete BBH-Methoden zu verwenden. Diese sind sicher, und Kunden profitieren von aktuellen und vor allem schnellsten Programmierlösungen. Im Folgenden sind die wichtigsten Verfahren aufgeführt.

● „Skip Bad Block“-Verfahren

Dieses Verfahren (Bild 1) identifiziert Bad Blocks, markiert sie als schlecht und ersetzt sie durch den nachfolgenden Block im Speicher. Fünf Varianten, die nach diesem Prinzip Bad Blocks verwalten, werden derzeit von Data I/O-Programmiersystemen unterstützt: 4 Block Skip, Mode2/Mode3, Ignore All, MSM6100 und MSM6100 EFS. Die beiden letzten sind geeignet für den Chipsatz MSM6100 von Qualcomm mit 12 byte Error Code Correction (ECC) und Embedded-File-System-Unterstützung (EFS).

● „Reserved Block Area“-Verfahren (RBA)

Bei dieser Methode (Bild 2) wird am Ende des Halbleiters ein Reservoir mit Ersatzblöcken definiert. Daten aus Bad Blocks im Nutzbereich werden nach Identifikation hierher kopiert. Nach diesem Verfahren arbeiten das GBBM (Global Bad Block Management) von Samsung, Windows CE sowie das Unistore.

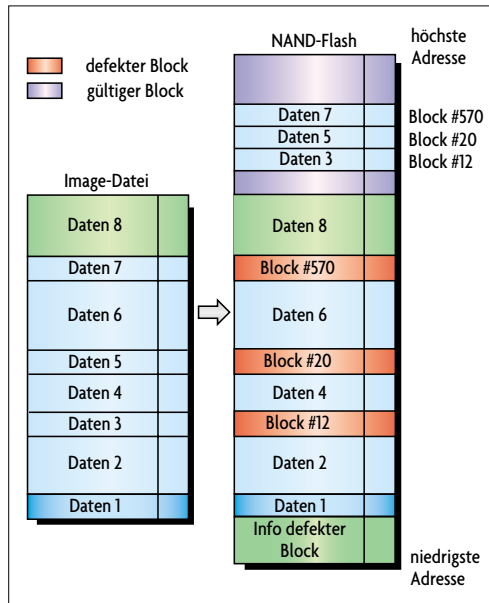


**Dipl.-Ing. (FH) Edwin Musch**

aus dem oberschwäbischen Rot an der Rot studierte Elektrotechnik an der FH Kempten/Allgäu. Seit seinem Abschluss ist er bei Data I/O Corporation/GmbH tätig und arbeitet an den Standorten Wangen und Redmond/USA. Heute ist er in der europäischen Zentrale von Data I/O in Gräfelfing bei München als FAE beschäftigt.  
 ▶ E-Mail: e.musch@data-io.de

● „File System Map“-Verfahren

„File System Map“ (Bild 3) ist eine Abwandlung der RBA-Methode. Es benutzt eine im Image integrierte Übersicht, um Bad-Block-Info-



**Bild 3. Das „File System Map“-Verfahren funktioniert ähnlich wie das RBA-Verfahren aus Bild 2, nutzt zur Verwaltung der Bad-Block-Informationen aber eine Dateisystemtabelle am Anfang des Flash-Bausteins.**

mationen zu platzieren. Am Ende des Speichers werden Blöcke reserviert, die als Ersatz für Bad Blocks im User-Bereich dienen. Die Information über das Bad-Block-Handling befindet sich in den ersten beiden Blöcken.

▶ **Programmierung kritisch bei Serienproduktion**

Die Frage, wie ein Halbleiter programmiert wird, spielt eine entscheidende Rolle bei der Wahl des Bausteins für neue Produktdesigns. Während im Prototypenstadium Halbleiter überwiegend noch mit seriellen, vom Bausteinhersteller zur Verfügung gestellten Methoden programmiert werden können, sind diese bei der Serienfertigung meist unbrauchbar. Darüber hinaus fehlen in den Spezifikationen oft wichtige Informationen zur Programmierung des entsprechenden Bausteins. Gerne werden die kürzestmöglichen Programmierzeiten angegeben. Das ist aber meist nicht die ganze Wahrheit: Maximale Geschwindigkeiten werden erfahrungsgemäß nur mit parallelen Programmiermethoden und nicht mit serieller Onboard-Programmierung erreicht.

Folgendes Beispiel zeigt, was passieren kann, wenn wichtige Informationen fehlen. Ein Ingenieur der Entwicklungsabteilung entwirft einen Media-Router, wählt dazu einen NAND-Flash-

Speicherbaustein und programmiert gemäß den Empfehlungen des Bausteinherstellers mittels eines Mikrocontrollers mit Hilfe von seriellem JTAG-Equipment. Dann tritt kurz vor Serienproduktion ein Problem auf. Die Fertigung kann eine Programmierzeit von sechs Minuten pro Bauteil nicht akzeptieren; sie ist mit Kapazität und Budget unvereinbar. Die Produktion wird zunächst gestoppt. Statt dessen muss ein externer Programmierdienstleister beauftragt werden, der zusätzlich Kosten verursacht. Außerdem vergrößern die vorprogrammierten Bausteine den Lagerbestand.

Ebenfalls problematisch ist es, wenn sich die Pro-

grammiertechnologien von Entwicklungs- und Fertigungsabteilung unterscheiden, d.h. nicht einheitliche Programmiersysteme verwendet werden und keine Schnittstellen vorhanden sind. Spezielle Features eines Bausteins können dann nicht richtig für die Massenfertigung genutzt werden. In diesem Fall verschiebt sich der Launch des Produktes, da erst eine entsprechende Programmierlösung für die Produktion entwickelt werden muss.

Um so wichtiger ist es daher, alle Informationen rund um die Programmierung für Entwicklung und Produktion bereitzustellen. Das „NAND Flash Device Support“-Portal unter [www.dataio.com/nand](http://www.dataio.com/nand) dient vorab als ausführliche Informationsquelle für Entwickler und Hersteller, die das Potential von NAND-Flash-Speicherbausteinen voll ausschöpfen, Kosten reduzieren und Produktionsengpässe vermeiden wollen. Hier finden sich White Papers führender Halbleiterhersteller, Application Notes und eine Liste mit Distributoren von NAND-Flash-Speicherbausteinen; außerdem eine vollständige Übersicht der NAND-Flashes, die derzeit von den Programmierlösungen von Data I/O unterstützt werden. Durchgängige Programmiersysteme garantieren, dass die Anforderungen für die jeweilige Applikation direkt über den Algorithmus abgebildet werden. *jk*