



(12)

Veröffentlichung

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2019/045951**
in der deutschen Übersetzung (Art. III § 8 Abs. 2
IntPatÜG)
(21) Deutsches Aktenzeichen: **11 2018 004 795.4**
(86) PCT-Aktenzeichen: **PCT/US2018/045223**
(86) PCT-Anmeldetag: **03.08.2018**
(87) PCT-Veröffentlichungstag: **07.03.2019**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **18.06.2020**

(51) Int Cl.: **G11C 16/14 (2006.01)**
G11C 16/34 (2006.01)
G11C 29/00 (2006.01)
G11C 29/44 (2006.01)

(30) Unionspriorität:
62/551,314 29.08.2017 US
15/850,779 21.12.2017 US

(71) Anmelder:
**Cypress Semiconductor Corporation, San Jose,
Calif., US**

(74) Vertreter:
Murgitroyd & Company, 80636 München, DE

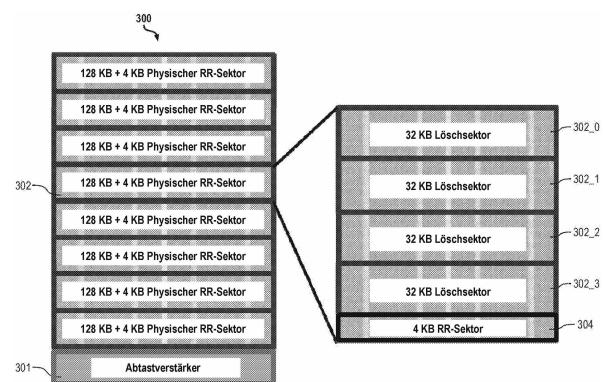
(72) Erfinder:
**Danon, Kobi, Tel-Aviv, IL; Betser, Yoram, Mazkeret
Banya, IL; Kotlincki, Uri, Ramat Gan, IL; Feldman,
Arieh, Yokne'am Illit, IL**

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **ZEILENREDUNDANZ MIT VERTEILERSEKTOREN**

(57) Zusammenfassung: Hierin werden Techniken für Flashspeicher mit Zeilenredundanz beschrieben. In einem Ausführungsbeispiel beinhaltet eine Halbleitervorrichtung einen eingebetteten Flashspeicher. Der eingebettete Flashspeicher umfasst eine Speicherbank, die mehrere physische Sektoren umfasst, wobei jeder physische Sektor eine Vielzahl von Löschsektoren beinhaltet. In der Speicherbank werden mehrere Abschnitte eines zusätzlichen Löschsektors jeweils auf die mehreren physischen Sektoren verteilt. Die mehreren Abschnitte des zusätzlichen Löschsektors sind als Zeilenredundanzsektor für die Speicherbank konfiguriert.



Beschreibung**PRIORITÄT**

[0001] Diese Patentanmeldung ist eine internationale Patentanmeldung der am 21. Dezember 2017 angemeldeten US-amerikanischen nichtvorläufigen Patentanmeldung Nr. 15/850,779, die den Vorteil der am 29. August 2017 angemeldeten US-amerikanischen vorläufigen Patentanmeldung Nr. 62/551,314 beansprucht, die alle in ihrer Gesamtheit durch Verweis hierin aufgenommen werden.

TECHNISCHES GEBIET

[0002] Diese Offenlegung bezieht sich im Allgemeinen auf nichtflüchtige Speichervorrichtungen und insbesondere auf die Bereitstellung von Hardware-Zeilenredundanz im Flashspeicher.

HINTERGRUND

[0003] Nichtflüchtige Speicher, wie z. B. Flashspeicher, werden häufig zur Speicherung von Daten und Anweisungen in Computervorrichtungen verwendet. Ein Flashspeicher umfasst typischerweise eine oder mehrere Speicherbänke aus physischen Sektoren, die Speicherzellen aufweisen, welche in Zeilen und Spalten eingerichtet sind. Einige Halbleitervorrichtungen, wie z. B. Mikrocontroller, können einen Flashspeicher umfassen, der darauf auf demselben Halbleiterchip eingebettet ist.

[0004] Es ist wünschenswert, eingebettete Flashspeicher mit Zeilenredundanz für die Korrektur von Herstellungsfehlern nach der Herstellung bereitzustellen, aber eine effiziente Implementierung einer solchen Zeilenredundanz ist nicht einfach. In einigen Anwendungen entspricht beispielsweise die Menge der erforderlichen Zeilenredundanz der Größe eines Viertels eines physischen Sektors. Das Hinzufügen einer solch kleinen Menge an Flashspeicher zu jeder Speicherbank erfordert jedoch die Verwendung eines isolierten physischen Sektors mit einer kleinen, nicht standardmäßigen Größe, was jedoch einen großen Overhead in Bezug auf die Die-Fläche mit sich bringt. Dies liegt daran, dass X-Decodierer und Y-Decodierer mit normaler Größe erforderlich sind, um Speichervorgänge auf dem kleinen, nicht standardmäßigen physischen Sektor durchzuführen, und solche normalen X-Decodierer und Y-Decodierer nehmen eine Die-Fläche von der gleichen Größe ein wie die Decodierer, die für den Betrieb der normalen physischen Sektoren benötigt werden.

Figurenliste

Fig. 1 ist ein Blockdiagramm, das ein beispielhaftes Die-Layout einer physischen x2-Sektorschtruktur gemäß einigen Ausführungsformen veranschaulicht.

Fig. 2 ist ein Blockdiagramm, das eine beispielhafte Speicherbank eines Flashspeichers ohne Zeilenredundanz veranschaulicht.

Fig. 3 ist ein Blockdiagramm, das eine beispielhafte Speicherbank eines Flashspeichers mit verteilter Zeilenredundanz gemäß einigen Ausführungsformen veranschaulicht.

Fig. 4 ist ein Blockdiagramm, das die Struktur eines beispielhaften physischen Sektors mit verteilter Zeilenredundanz gemäß einigen Ausführungsformen veranschaulicht.

Fig. 5 ist ein Blockdiagramm, das einen beispielhaften Mikrocontroller mit eingebettetem Flashspeicher gemäß einigen Ausführungsformen veranschaulicht.

DETAILLIERTE BESCHREIBUNG

[0005] Die folgende Beschreibung legt zahlreiche spezifische Details, wie etwa Beispiele für spezifische Systeme, Komponenten, Verfahren und so weiter dar, so dass verschiedene Ausführungsformen der hierin beschriebenen Techniken für die verteilte Zeilenredundanz im Flashspeicher gut verstanden werden können. Für den Fachmann wird es jedoch ersichtlich sein, dass mindestens einige Ausführungsformen ohne diese spezifischen Details praktiziert werden können. In anderen Beispielen werden bekannte Komponenten oder Verfahren nicht im Detail beschrieben oder werden in einem einfachen Blockdiagrammformat präsentiert, um eine unnötige Verschleierung der hierin beschriebenen Techniken zu vermeiden. Somit sind die im Folgenden dargelegten spezifischen Details lediglich beispielhaft. Bestimmte Implementierungen können von diesen beispielhaften Details abweichen und dennoch als innerhalb des Wesens und Schutzbereichs der vorliegenden Erfindung erachtet werden.

[0006] Verweise in der Beschreibung auf „eine Ausführungsform“, „ein Ausführungsbeispiel“, „einige Ausführungsformen“ und „verschiedene Ausführungsformen“ bedeuten, dass ein bestimmtes Merkmal, eine bestimmte Struktur oder Charakteristik, die im Zusammenhang mit der/den Ausführungsform(en) beschrieben werden, in mindestens einer Ausführungsform der Erfindung umfasst ist. Des Weiteren verweisen die Ausdrücke „eine Ausführungsform“, „ein Ausführungsbeispiel“, „einige Ausführungsformen“ und „verschiedene Ausführungsformen“ an verschiedenen Stellen in der Beschreibung nicht not-

wendigerweise alle auf dieselbe/n Ausführungsform (en).

[0007] Die Beschreibung umfasst Verweise auf die begleitenden Figuren, die einen Teil der detaillierten Beschreibung bilden und in denen sich ähnliche Bezugszeichen auf ähnliche Komponenten oder Vorgänge beziehen. Die Zeichnungen zeigen Veranschaulichungen gemäß beispielhaften Ausführungsformen. Diese Ausführungsformen, die hierin auch „Beispiele“ genannt werden können, werden ausreichend detailliert beschrieben, um es dem Fachmann zu ermöglichen, die Ausführungsformen des hierin beschriebenen, beanspruchten Gegenstands zu praktizieren. Die Ausführungsformen können kombiniert werden, andere Ausführungsformen können benutzt werden oder strukturelle, logische und elektrische Veränderungen können vorgenommen werden, ohne dass vom Schutzbereich und Wesen des beanspruchten Gegenstands abgewichen wird. Es versteht sich, dass die hierin beschriebenen Ausführungsformen den Schutzbereich des beschriebenen Gegenstands nicht beschränken sollen, sondern es dem Fachmann vielmehr ermöglichen sollen, einen solchen Gegenstand in die Praxis umzusetzen, herzustellen und/oder zu verwenden.

[0008] Ein Flashspeicher, wie z. B. ein eingebetteter Flash, muss typischerweise eine gewisse Menge an Zeilenredundanz-Speicherzellen aufweisen, die Speicherzellen mit Herstellungsfehlern, welche zeilenbezogen sind (z. B. Fehler in X-Richtung in einer Speicherzellenanordnung) ersetzen können. Die Menge der notwendigen Zeilenredundanz kann basierend auf verschiedenen Herstellungsparametern, wie z. B. Ertragsanalyse, Größe des physischen Sektors, usw., bestimmt werden. In einigen Anwendungen ist die Standardanforderung, mindestens einen Löschsektor (z. B. pro ~ 1 MB Speicherbank) für den Ersatz eines fehlerhaft hergestellten Löschsektors nach der Herstellung zu haben, wobei vier Löschsektoren einen physischen Sektor in der Speicherbank umfassen können. Wenn nach der Herstellung in einem Löschsektor während eines Sortierablaufvorgangs ein zeilenbezogener Fehler festgestellt wird, muss der fehlerhafte Löschsektor durch redundante Sektor-Selektions-Gate-Leitungen (SG-Leitungen) (auch als Wortleitungen oder WLs bezeichnet), Speicher-Gate-Leitungen (MG(Memory Gate)-Leitungen) und Source-Leitungen (SLs) für alle Flash-Vorgänge (z. B. Programmieren, Löschen, Verifizieren und Lesen) ersetzt werden. Ein Sortierablaufvorgang ist ein Vorgang nach der Herstellung, der von Prüfsonden und anderer Herstellungsausstattung auf einem Die durchgeführt werden kann, während dieser sich noch auf dem Wafer befindet, wenn der Die in einem Package angeordnet wird, oder zu irgendeinem anderen Zeitpunkt, bevor das Package an einen Kunden versandt wird. Die Fehler, die durch einen Sortierablaufvor-

gang detektiert werden können, können von der Herstellungstechnologie abhängen, wobei zeilenbezogene Fehler unter anderem MG-zu-MG-Kurzschlüsse, SG-zu-SG-Kurzschlüsse, MG-zu-SL-Kurzschlüsse, MG-zu-BL(Bitleitung)-Kurzschlüsse, MG-Stromkreisunterbrechungen, SG-Stromkreisunterbrechungen und SL-Stromkreisunterbrechungen sein können.

[0009] Die Herausforderung beim Hinzufügen von Zeilenredundanz besteht darin, den Die-Flächen-Mehraufwand zu minimieren. Beispielsweise können einige Anwendungen eine physische x2-Sektorstruktur verwenden, was bedeutet, dass ein physischer Sektor nicht allein hinzugefügt werden kann, sondern auch der angrenzende Sektor hinzugefügt werden muss. Der Grund dafür ist, dass Speicher-Lesevorgänge unter Verwendung lokaler differenzieller Abtastverstärker (Sense Amplifier, SA) durchgeführt werden, welche Referenzströme verwenden, die von auf dem angrenzenden physischen Sektor befindlichen Referenzzellen bereitgestellt werden. Beispielsweise kann eine solche x2-Struktur in Anwendungen, die Speicherzugriff auf jeweils einen physischen Sektor erfordern, bevorzugt werden, da derselbe Satz von SAs für zwei physische Sektoren verwendet werden kann, wodurch Die-Fläche eingespart wird, die ansonsten erforderlich wäre, wenn jeder physische Sektor seinen eigenen SA aufweist. **Fig. 1** ist ein Blockdiagramm, das ein Die-Layout einer solchen physischen x2-Sektorstruktur gemäß einigen Ausführungsformen veranschaulicht.

[0010] In **Fig. 1** umfasst die x2-Struktur **100** eine Kernfläche mit Flashspeicherzellen und Die-Flächen **101**, **103** und **105**, die im Wesentlichen ein „peripherer“ Die-Flächen-Overhead sind. Die Fläche **101** ist zwischen zwei angrenzenden physischen Sektoren **102a** und **102b** angeordnet, die jeweils Kernflächen mit in Zeilen und Spalten eingerichteten Flashspeicherzellen umfassen. Eine Flashspeicherzelle umfasst typischerweise ein Selektions-Gate, ein Speicher-Gate und einen oder mehrere Speichertransistoren, die über einem Kanal angeordnet sind, der in einem Substrat einer integrierten Schaltung (Integrated Circuit, IC) zwischen einem Source-Bereich und einem Drain-Bereich auf dem Substrat angeordnet ist. Die Fläche **101** beherbergt die lokalen SAs (die auf beiden physischen Sektoren **102a** und **102b** arbeiten) und die Beschaltung, die konfiguriert ist, um die oberen (sowohl ungeraden als auch geraden) SG-Leitungen auszuwählen. Die Fläche **103** beherbergt die Beschaltung, um die unteren/geraden SG-Leitungen auszuwählen, und die Fläche **105** beherbergt die Beschaltung, um die unteren/ungeraden SG-Leitungen auszuwählen. Die X2-Struktur **100** umfasst auch SG-Treiber **106**, SL-Treiber **108** und MG-Treiber **110**. Ein Treiber ist eine Schaltung aus elektronischen Elementen (z. B. Transistoren, Dioden, Logikgatter, usw.), die konfiguriert sind, bestimmte Spannungen

und/oder Ströme an daran gekoppelte andere elektronische Elemente oder Komponenten (z. B. Leiterbahnen, Speicherzellen, usw.) anzulegen. In **Fig. 1** sind die SG-Treiber **106** konfiguriert, um SG-Leitungen (auch als Wortleitungen bezeichnet) anzusteuern, die an die ausgewählten Selektions-Gates der Speicherzellen in den physischen Sektoren **102a** und **102b** gekoppelt sind. Die SL-Treiber **108** sind konfiguriert, um Source-Leitungen anzusteuern, die an die Source-Bereiche von Speicherzellen in den physischen Sektoren **102a** und **102b** gekoppelt sind. Die MG-Treiber **110** sind konfiguriert, MG-Leitungen anzusteuern, die an die Speicher-Gates von Speicherzellen in den physischen Sektoren **102a** und **102b** gekoppelt sind. Es sollte sich verstehen, dass auf dem Die der Abstand der Halbleiterelemente in den verschiedenen Treibern größer ist als der Abstand, mit dem die Flashspeicherzellen in den Kernflächen der physischen Sektoren **102a** und **102b** angeordnet sind.

[0011] Ein herkömmlicher Ansatz zur Implementierung von Zeilenredundanz ist die Hinzufügung eines speziell dafür vorgesehenen physischen Sektors zu der Speicheranordnung. Zum Beispiel muss bei Anwendungen, die eine physische x2-Sektorstruktur verwenden (wie z. B. in **Fig. 1** gezeigt), die geeignete Menge an WLs, die benötigt werden, um einen Löschsektor zu ersetzen, als ein separater (wenn auch kleiner) physischer Sektor hinzugefügt werden. Bei diesem Ansatz gibt es zwei Probleme. Zunächst muss die für eine physische x2-Sektorstruktur erforderliche periphere Beschaltung (z. B. die SAs und die Selektions-Elemente) getrennt von der peripheren Beschaltung, die für den Betrieb der Kern-Flashspeicherzellen verwendet wird, hinzugefügt werden. Zweitens beträgt die Anzahl von WLs in einem speziell dafür vorgesehenen zeilenredundanten Sektor nur $\frac{1}{4}$ derjenigen eines normalen physischen Sektors. Diese beiden Probleme stellen ein großes Layout-Problem für die X-Decodierer (z. B. mit den darin umfassten SG-, SL- und MG-Treibern) dar, da die Sektorhöhe des speziell dafür vorgesehenen Sektors nur $\frac{1}{4}$ eines normalen Sektors beträgt und die Breite nicht ausreicht, um die in **Fig. 1** gezeigten MG- und SL-Treiber einzupassen. Dies führt zu einer größeren Breite der Decodierer, was sich auf die Breite der gesamten Anordnung auswirkt. Während für einen normalen physischen Sektor die gemeinsame Nutzung von MG- und SL-Treibern umfangreich sein könnte (wodurch die Gesamtbreite des X-Decodierers reduziert wird), sind die Möglichkeiten der gemeinsamen Nutzung für den speziell dafür vorgesehenen zeilenredundanten Sektor begrenzt und führen zu einem breiteren X-Decodierer.

[0012] Um diese und andere Probleme zu lösen, werden hierin verschiedene Ausführungsformen von Techniken für verteilte Zeilenredundanz im Flashspeicher beschrieben. In diesen Ausführungsformen

besteht die Lösung der oben genannten Probleme darin, die Zeilenredundanz (Row Redundancy, RR)-WLs innerhalb von mehreren der normalen physischen Sektoren zu verteilen, so dass die Gesamtgröße der an die RR-WLs gekoppelten Speicherzellen gleich der Größe mindestens eines Löschsektors ist, der als Zeilenredundanzsektor verwendet würde. Wenn der Zeilenredundanzsektor tatsächlich konfiguriert wird, um einen fehlerhaften Löschsektor zu ersetzen, greifen die Speichervorgänge auf dieselben normalen physischen Sektoren zu, jedoch mit unterschiedlichen dekodierten Adresszielen, die auf die Speicherzellen verweisen würden, die an die RR-WLs in dem verteilten Zeilenredundanzsektor gekoppelt sind. Auf diese Weise verwenden die hierin beschriebenen verteilten Zeilenredundanztechniken denselben Satz von Dekodierungs- und Hilfsbeschaltung, der für die normalen physischen Sektoren verwendet wird. Beispielsweise teilen sich die X-Decodierer (z. B. mit ihren SG-, MG- und SL-Treibern), die ausgewählten Y-Decodierungstransistoren und die SAs, die für die normalen physischen Sektoren verwendet werden, dieselbe Die-Fläche wie die X-Decodierer, die verwendet werden, um die RR-WLs in dem verteilten Zeilenredundanzsektor zu wählen. Dies führt zu einer erheblichen Einsparung bei der Die-Fläche und zu einem robusten Redundanzschema.

[0013] In einem Ausführungsbeispiel beinhaltet eine Halbleitervorrichtung einen eingebetteten Flashspeicher, der eine Speicherbank mit mehreren physischen Sektoren umfasst. Jeder physische Sektor umfasst eine Vielzahl von Löschsektoren, wobei mehrere Abschnitte eines zusätzlichen Löschsektors jeweils auf die mehreren physischen Sektoren verteilt sind und die mehreren Abschnitte des zusätzlichen Löschsektors als Zeilenredundanzsektor für die Speicherbank konfiguriert sind. In einem Aspekt dieser Ausführungsform umfasst die Speicherbank einen fehlerhaften Löschsektor (der z. B. während eines Sortierablaufvorgangs auf dem eingebetteten Flashspeicher bestimmt werden kann), und die mehreren Abschnitte des zusätzlichen Löschsektors sind konfiguriert, um den fehlerhaften Löschsektor innerhalb der Speicherbank zu ersetzen. In einem beispielhaften Aspekt ist der eingebettete Flashspeicher konfiguriert, während eines Speichervorgangs auf der Speicherbank auf die mehreren Abschnitte des zusätzlichen Löschsektors statt auf den fehlerhaften Löschsektor zuzugreifen. Der Speichervorgang kann zum Beispiel ein Lesevorgang oder ein Programmiervorgang sein. Der Speichervorgang kann auch ein Löschvorgang sein, der seriell oder parallel auf den mehreren physischen Sektoren durchgeführt wird, um jeden der mehreren Abschnitte des zusätzlichen Löschsektors zu löschen. In einem Aspekt beinhaltet die Halbleitervorrichtung erste WLs, die an erste Speicherzellen in den mehreren physischen Sektoren gekoppelt sind, und zweite WLs, die zusätz-

lich zu den ersten WLs auf die mehreren physischen Sektoren verteilt sind, wobei die zweiten WLs an zweite Speicherzellen in den mehreren Abschnitten des zusätzlichen Löschsektors gekoppelt sind. In einem weiteren Aspekt beinhaltet die Speicherbank zusätzlich zu den mehreren physischen Sektoren einen oder mehrere physische Sektoren, die keinen Abschnitt des zusätzlichen Löschsektors umfassen.

[0014] In einem weiteren Ausführungsbeispiel beinhaltet ein System einen Mikrocontroller und einen in dem Mikrocontroller eingebetteten Flashspeicher. Der Flashspeicher beinhaltet eine Speicherbank, die mehrere physische Sektoren umfasst. Jeder physische Sektor umfasst eine Vielzahl von Löschsektoren, wobei mehrere Abschnitte eines zusätzlichen Löschsektors jeweils auf die mehreren physischen Sektoren verteilt sind und die mehreren Abschnitte des zusätzlichen Löschsektors als Zeilenredundanzsektor für die Speicherbank konfiguriert sind. In einem Aspekt dieser Ausführungsform umfasst die Speicherbank einen fehlerhaften Löschsektor (der z. B. während eines Sortierablaufvorgangs auf dem eingebetteten Flashspeicher bestimmt werden kann), und die mehreren Abschnitte des zusätzlichen Löschsektors sind konfiguriert, um den fehlerhaften Löschsektor innerhalb der Speicherbank zu ersetzen. In einem beispielhaften Aspekt ist der Flashspeicher konfiguriert, während eines Speichervorgangs auf der Speicherbank auf die mehreren Abschnitte des zusätzlichen Löschsektors statt auf den fehlerhaften Löschsektor zuzugreifen. Der Speichervorgang kann zum Beispiel ein Lesevorgang oder ein Programmervorgang sein. Der Speichervorgang kann auch ein Löschtorgang sein, der seriell oder parallel auf den mehreren physischen Sektoren durchgeführt wird, um jeden der mehreren Abschnitte des zusätzlichen Löschsektors zu löschen.

[0015] In einem weiteren Ausführungsbeispiel beinhaltet ein Verfahren zum Konfigurieren eines eingebetteten Flashspeichers mit Zeilenredundanz Folgendes: Bereitstellen von ersten WLs, die in mehreren physischen Sektoren einer Speicherbank in dem eingebetteten Flashspeicher angeordnet sind, wobei jeder physische Sektor eine Vielzahl von Löschsektoren beinhaltet; und Bereitstellen von zweiten WLs, die zusätzlich zu den ersten WLs in den mehreren physischen Sektoren angeordnet sind, wobei die zweiten WLs konfiguriert sind, um mehrere Abschnitte eines zusätzlichen Löschsektors zu bilden, die jeweils auf die mehreren physischen Sektoren verteilt sind, und wobei der zusätzliche Löschsektor als ein Zeilenredundanzsektor für die Speicherbank konfiguriert ist. In einem Aspekt dieser Ausführungsform beinhaltet das Verfahren ferner Folgendes: Bestimmen (z. B. während eines Sortierablaufvorgangs auf dem eingebetteten Flashspeicher), dass die Speicherbank einen fehlerhaften Löschsektor umfasst; und Konfigurieren des eingebetteten Flashspeichers, um den feh-

lerhaften Löschsektor mit den mehreren Abschnitten des zusätzlichen Löschsektors in der Speicherbank zu ersetzen. In einem beispielhaften Aspekt kann das Verfahren ferner das Konfigurieren des eingebetteten Flashspeichers für einen Speichervorgang (z. B. einen Lese- oder Programmervorgang), der auf die mehreren Abschnitte des zusätzlichen Löschsektors statt auf den fehlerhaften Löschsektor zugreifen soll, beinhalten. In einem weiteren beispielhaften Aspekt kann das Verfahren ferner das Konfigurieren des eingebetteten Flashspeichers für einen Löschtorgang, der seriell oder parallel auf den mehreren physischen Sektoren durchgeführt werden soll, um jeden der mehreren Abschnitte des zusätzlichen Löschsektors zu löschen, beinhalten.

[0016] Entsprechend den hierin beschriebenen Techniken besteht eine Alternative zur herkömmlichen Lösung darin, den/die notwendigen Zeilenredundanzsektor(en) auf die normalen physischen Sektoren zu verteilen, die für Benutzerdaten verwendet werden. Zum Beispiel veranschaulicht **Fig. 2** eine 1-MB-Speicherbank eines Flashspeichers ohne Zeilenredundanz. In **Fig. 2** umfasst die Speicherbank **200** Abtastverstärker **201**, die an acht physische Sektoren **202** gekoppelt sind, die jeweils 128 KB Flashspeicherzellen aufweisen. Es sollte sich verstehen, dass obwohl die Abtastverstärker **201** als unten an der Speicherbank **200** angeordnet veranschaulicht sind, die Abtastverstärker **201** in verschiedenen Architekturen auch (eine) unterschiedliche Die-Fläche(n) belegen können. In einer Speicherbank, die physische x2-Sektorstrukturen verwendet, können Abtastverstärker **201** beispielsweise zwischen jedem der vier Paare physikalischer Sektoren **202** verteilt und angeordnet sein. Wie in **Fig. 2** veranschaulicht, umfasst jeder physische Sektor **202** vier Löschsektoren **202_0**, **202_1**, **202_2** und **202_3**, die jeweils 32 KB Flashspeicherzellen aufweisen.

[0017] Um der Speicherbank in **Fig. 2** Zeilenredundanz hinzuzufügen, sind in einigen Ausführungsformen mehrere Abschnitte eines physischen Sektors, in Summe gleich einem Löschsektor von 32 KB Flashspeicherzellen, innerhalb jedes physischen Sektors verteilt. Ein Beispiel für eine solche verteilte Zeilenredundanz ist in **Fig. 3** veranschaulicht.

[0018] In **Fig. 3** umfasst die Speicherbank **300** Abtastverstärker **301**, die an acht physische Sektoren **302** gekoppelt sind, wobei jeder physische Sektor **302** 128 KB Flashspeicherzellen und einen 4-KB-Abschnitt des Zeilenredundanz-Löschenktors für die Speicherbank aufweist. Es sollte sich verstehen, dass in verschiedenen Architekturen die Abtastverstärker **301** (eine) unterschiedliche Die-Fläche(n) belegen können. In einer Speicherbank, die physische x2-Sektorstrukturen verwendet, können Abtastverstärker **301** beispielsweise zwischen jedem der vier Paare physikalischer Sektoren **302** verteilt und ange-

ordnet sein. Jeder physische Sektor **302** umfasst vier Löschsektoren **302_0**, **302_1**, **302_2** und **302_3**, die jeweils 32 KB Flashspeicherzellen aufweisen, wobei ein Abschnitt **304** des Zeilenredundanzsektors 4 KB Flashspeicherzellen aufweist.

[0019] Gemäß den hierin beschriebenen Techniken wird eine Flashspeicherbank mit Konfigurationsinformationen zum Ersetzen eines fehlerhaften Löschsektors mit den Abschnitten eines Zeilenredundanzsektors, der auf die normalen physischen Sektoren in der Speicherbank verteilt ist, bereitgestellt. Wenn beispielsweise ein Sortierablaufvorgang einen fehlerhaften Löschsektor in der Speicherbank identifiziert, wird der Ort des fehlerhaften Sektors in Konfigurations-Flipflops, Registern und/oder synthetisierter Logik in der Speicherbank aufgezeichnet. Die Konfigurations-Flipflops, Register und/oder synthetisierte Logik können permanent programmiert werden, um solche Ortsinformationen zu speichern, oder sie können beim Start mit solchen Informationen aktualisiert werden. Dann werden basierend auf diesen Ortsinformationen Speichervorgänge, die darauf abzielen, auf den fehlerhaften Sektor zuzugreifen, stattdessen auf dem Zeilenredundanzsektor durchgeführt. Beispielsweise kann die interne Logik der X-Decodierer diese Ortsinformationen verwenden, um zu bestimmen, dass die Zieladresse auf Speicherzellen in dem fehlerhaften Löschsektor gerichtet ist, und um die Zieladresse zu decodieren, um einen entsprechenden Abschnitt des verteilten Zeilenredundanzsektors, der in einem normalen physischen Sektor angeordnet ist, anzuzeigen. Auf diese Weise ist die Tatsache, dass die Zeilenredundanz auf mehrere normale physische Sektoren verteilt ist, nur eine Frage der Adressdecodierung. Lese- und Programmier-Speichervorgänge (die auf der Basis einer einzigen WL durchgeführt werden) sind nicht betroffen, und Löschtätigkeiten (die auf dem gesamten Zeilenredundanzsektor durchgeführt werden) können seriell immer nur an einem Ort in den normalen physischen Sektoren durchgeführt werden, ohne dass es zu einer erheblichen Leistungseinbuße kommt.

[0020] **Fig. 4** ist ein Blockdiagramm, das die Struktur eines beispielhaften physischen Sektors **402** mit verteilter Zeilenredundanz gemäß einigen Ausführungsformen der hierin beschriebenen Techniken veranschaulicht. Der physische Sektor **402** umfasst vier Löschsektoren **402_0** bis **402_3** und einen Abschnitt **404** eines Zeilenredundanzsektors. In dem Löschsektor **402_0** sind die SG-Leitungen **406_0**, die SLs **408_0** und die MG-Leitungen **410_0** angeordnet. In dem Löschsektor **402_3** sind die SG-Leitungen **406_3**, die SLs **408_3** und die MG-Leitungen **410_3** angeordnet. In dem Abschnitt **404** des Zeilenredundanzsektors sind die SG-Leitungen **416**, SLs **418** und die MG-Leitungen **420** angeordnet. In jedem der Löschsektoren **402_0** bis **403_3** und dem Abschnitt **404** des Zeilenredundanzsektors sind die jeweiligen

SG-Leitungen (z. B. **406_0** bis **406_3**, **416**) an die Selektions-Gates von Kernspeicherzellen gekoppelt; die jeweiligen SLs (**408_0** bis **408_3**, **418**) sind an die Source-Bereiche der Kernspeicherzellen gekoppelt; und die jeweiligen MG-Leitungen (**410_0** bis **410_3**, **420**) sind an die Speicher-Gates der Kernspeicherzellen gekoppelt. Die Kernspeicherzellen in dem physischen Sektor **402** (z. B. in den Löschsektoren **402_0** bis **402_3** und in dem Abschnitt **404** des Zeilenredundanzsektors) können irgendeine Art von Flashspeicherzellen sein, einschließlich in nicht einschränkender Weise Split-Gate-Speicherzellen, Zwei-Transistor(2T)-Speicherzellen, NAND-Speicherzellen, NOR-Speicherzellen, usw.

[0021] Es sollte sich verstehen, dass die verteilten Zeilenredundanzleitungen (z. B. die SG-Leitungen, die SL-Leitungen und die MG-Leitungen) in **Fig. 4** eine homogene Erweiterung der Leitungen in dem physischen Sektor **402** sind, und daher ist die Flächenoptimierung in dem physischen Sektor maximiert und die Einheitlichkeit der darin enthaltenen Speicheranordnungen bleibt erhalten. Es sollte sich auch verstehen, dass das Verteilen der Gesamtmenge der Zeilenredundanz-WLs dazu führt, dass zu jedem normalen physischen Sektor eine so geringe Menge zusätzlicher WLs hinzukommt, dass diese kaum irgendeine analoge Auswirkung (z. B. zusätzliche Kapazität, zusätzlichen Widerstand, usw.) auf die physischen Sektoren hat. Dies ist sehr effizient in Bezug auf die Die-Fläche und spart mit einem speziell dafür vorgesehenen Redundanzsektor dazugehörigen Overhead ein. Im Gegensatz zu einem konventionellen Zeilenredundanzschema, das einen speziell dafür vorgesehenen physischen Sektor verwendet, stellen die hierin beschriebenen Techniken daher ein effizientes Zeilenredundanzschema (per Die-Fläche) mit voller Funktionalität und Zuverlässigkeit und ohne jegliche andere Einbußen bereit.

[0022] **Fig. 5** ist ein Blockdiagramm, das eine beispielhafte Mikrocontrollereinheit (Mikrocontroller Unit, MCU) mit eingebettetem Flashspeicher gemäß einem Ausführungsbeispiel veranschaulicht. Es sollte sich verstehen, dass in **Fig. 5** die MCU **500** zum Zweck der Veranschaulichung vereinfacht wurde und deshalb keine vollständige Beschreibung sein soll. Es sollte sich auch verstehen, dass eine MCU einige, alle Komponenten der Ausführungsform in **Fig. 5** oder andere/zusätzliche Komponenten als diese umfassen kann. Ferner sollte es sich verstehen, dass in verschiedenen Ausführungsformen eine andere Halbleitervorrichtung als eine MCU - etwa ein Controller, ein Spezialzweck-Prozessor, ein Digitalsignalprozessor („DSP“), eine anwendungsspezifische integrierte Schaltung („ASIC“), ein feldprogrammierbares Gate-Array („FPGA“) oder dergleichen - die hierin beschriebenen Techniken für eine verteilte Zeilenredundanz verkörpern kann. Somit sollte die MCU **500** in **Fig. 5**

in einem veranschaulichenden, jedoch nicht in einem einschränkenden Sinn betrachtet werden.

[0023] Die MCU **500** wird als integrierte Schaltung (Integrated Circuit, IC) auf einem einzelnen Die aus Halbleitersubstrat verkörpert. Die MCU **500** umfasst unter anderem den Zentraleinheits(Central Processing Unit, CPU)-Kern **502** (der einen oder mehrere Prozessoren umfassen kann), den Nur-Lese-Speicher (Read-Only Memory, ROM) **504**, den statischen Direktzugriffsspeicher (Random Access Memory, SRAM) **506** und den Flashspeicher **510**, die über die Zwischenverbindung **508** aneinander gekoppelt sind. Es wird darauf hingewiesen, dass in verschiedenen Ausführungsformen unterschiedliche und/oder zusätzliche Elemente in der MCU mit jeglicher Möglichkeit oder jeglichem Typ von CPU-Kern umfasst sein können, und diese Elemente können an den CPU-Kern und/oder untereinander entweder direkt oder über Bus(se), Brücke(n) und/oder eine andere Beschaltung, die sich von einer Zwischenverbindung unterscheidet, gekoppelt sein.

[0024] Der CPU-Kern **502** ist konfiguriert, um Anweisungen auszuführen, die aus dem ROM **504**, dem SRAM **506**, dem Flashspeicher **510** und/oder anderen Speichermodulen (falls solche konfiguriert sind) abgerufen werden können. Der ROM **504** ist als Firmware programmiert, um Anweisungen und/oder Daten (z. B. Konfigurationsdaten) zu speichern, auf die der CPU-Kern **502** und andere Elemente der MCU **500** zugreifen können. Der SRAM **506** ist eine flüchtige Speichervorrichtung, die konfiguriert ist, um Daten und/oder Anweisungen zu speichern, ohne dass eine Auffrischung erforderlich ist (solange sie mit Strom versorgt wird), und in die in beliebiger Reihenfolge geschrieben oder von der in beliebiger Reihenfolge gelesen werden kann, unabhängig von dem letzten Speicherort, auf den zugegriffen wurde. Die Zwischenverbindung **508** ist konfiguriert, um eine Adress-, Daten- und Steuerschnittstelle zwischen der CPU **502** und den anderen MCU-Elementen bereitzustellen.

[0025] Der Flashspeicher **510** ist eine Speichervorrichtung mit einem eingebetteten nichtflüchtigen Speicher (Non-Volatile Memory, NVM), der Daten, Konfigurationsinformationen und/oder ausführbare Anweisungen speichern kann. Der Flashspeicher **510** ist an die Zwischenverbindung **508** gekoppelt (z. B. über einen Adress-, Daten- und Steuerbus) und ist konfiguriert, um Informationen von dieser zu empfangen und Informationen an diese zu senden. Der Flashspeicher **510** umfasst neben anderen Elementen den Flash-Controller **512**, den X-Decodierer **514**, den Y-Decodierer **516** und die Speicheranordnung **520**. Der Flash-Controller **512** ist mit verschiedenen Hardwarekomponenten (z. B. Transistoren, Logik-Gates und verschiedenen anderen analogen und/oder digitalen Schaltungen) konfiguriert, um eine

Zustands- und Befehlssteuerung für den Flashspeicher **510** bereitzustellen. Der Flash-Controller **512** ist beispielsweise gekoppelt, um Signale von verschiedenen Steuerleitungen und Bussen zu empfangen, um Steuer- und Befehlssignale zu senden (z. B. an den X-Decodierer **514**, den Y-Decodierer **516**, usw.) und um an die Speicheranordnung **520** während verschiedener Speichervorgänge angelegte Signale zu steuern.

[0026] Die Speicheranordnung **520** ist eine Anordnung von Flashspeicherzellen, die in Zeilen und Spalten eingerichtet sind, so dass auf jede Speicherzelle durch eine Zeilen- und Spaltenadresse über Zeilen- und Spaltentreiberschaltungen zugegriffen werden kann. Eine Speicherzelle kann ein oder mehrere Bits an Informationen in ihrem/ihren Speichertransistor(en) speichern, auf die in Lese-, Programmier- und Löschvorgängen zugegriffen werden kann. In der Ausführungsform der **Fig. 5** sind die Speicherzellen in der Speicheranordnung **520** in mehreren physischen Sektoren angeordnet, die als eine oder mehrere Speicherbänke eingerichtet sein können, und jeder physische Sektor kann eine Anzahl von Zeilen und Spalten von Speicherzellen umfassen.

[0027] Die Speicheranordnung **520** ist, neben anderen MCU-Elementen, an den X-Decodierer **514** über Selektions-Gate-Leitungen, Source-Leitungen und Speicher-Gate-Leitungen gekoppelt. Die Speicheranordnung **520** ist, neben anderen MCU-Elementen, über mehrere Bitleitungen auch an den Y-Decodierer **516** gekoppelt. Der X-Decodierer **514** und der Y-Decodierer **516** umfassen Beschaltungen, die konfiguriert sind, um während Speichervorgängen Adressinformationen von dem Flash-Controller **512** zu empfangen und die Adressinformationen zu dekodieren, um die entsprechende(n) Speicherzelle(n) zu bestimmen. Zum Beispiel enthält der X-Decodierer **514**, neben anderer Beschaltung, SG-Treiber, SL-Treiber, MG-Treiber und verschiedene periphere Beschaltungen (z. B. Ladepumpen, Digital-Analog-Wandler, Verstärker, usw.), die betreibbar sind, um während Speichervorgängen die adressierte(n) Reihe(n) von Speicherzellen zu identifizieren. Der Y-Decodierer **516** enthält, neben anderer Beschaltung, Bitleitungstreiber und verschiedene periphere Beschaltungen, die betreibbar sind, um während Speichervorgängen die adressierte(n) Spalte(n) von Speicherzellen zu identifizieren.

[0028] Die Speicheranordnung **520** umfasst die Abtastverstärker **520a**. Es wird darauf hingewiesen, dass in verschiedenen Architekturen die Abtastverstärker **520a** unterschiedliche Die-Fläche(n) der Speicheranordnungen **520** belegen können. In Ausführungsformen, die physische x2-Sektorstrukturen verwenden, können die Abtastverstärker beispielsweise zwischen jedem Paar physikalischer Sektoren in der Speicheranordnung angeordnet sein. Die

Abtastverstärker **520a** sind konfiguriert, um während Speichervorgängen Mehrbit-Wörter aus den Speicherzellen in der Speicheranordnung zu lesen und die Mehrbit-Wörter an Beschaltungen (z. B. Zwi-schenspeicher, Datenpuffer usw.), die durch den Flash-Controller **512** gesteuert werden, zu übertragen.

[0029] In einem Lese-Speichervorgang wird ein in der Speicheranordnung **520** gespeichertes Bit aus einer in dem Lesevorgang spezifizierten Adresse ausgelesen. Die Adressinformationen werden von dem Flash-Controller **512** weitergeleitet und durch den X-Decodierer **514** und den Y-Decodierer **516** decodiert, um die entsprechende Speicherzelle zu bestimmen. Als Beispiel für ein strombasiertes Abtasten wird, um ein Bit zu lesen, das an der decodierten Adresse gespeichert ist, eine Spannung an das Speicher-Gate der entsprechenden Speicherzelle angelegt. Die Spannung wird präzise erzeugt, so dass diese im Wesentlichen gleich weit von der Zielspannung, die notwendig ist, um den Speichertransistor anzuschalten, wenn ein „hohes“ Bit gespeichert wird, und der Zielspannung, die notwendig ist, um den Transistor anzuschalten, wenn ein „niedriges“ Bit gespeichert wird, entfernt ist, um klar zwischen zwei Zuständen zu unterscheiden. Wenn das Anlegen der Spannung während des Lesevorgangs einen Abtastverstärker **520a** veranlasst, ein Fließen eines wesentlichen Stromes zwischen Source und Drain der Speicherzelle, der gleich dem Zielstrombereich ist, zu detektieren, dann enthält die Speicherzelle ein „hohes“ Bit. Alternativ enthält, wenn das Anlegen der Spannung während des Lesevorgangs den Abtastverstärker **520a** nicht veranlasst, ein Fließen eines wesentlichen Stromes zwischen Source und Drain der Speicherzelle, der gleich dem Zielstrombereich ist, zu detektieren, die Speicherzelle dann ein „niedriges“ Bit.

[0030] In einem Programmier-Speichervorgang wird ein in der Speicheranordnung **520** gespeichertes Bit auf eine in dem Programmervorgang spezifizierte Adresse geschrieben. Die Adressinformationen werden an den X-Decodierer **514** und den Y-Decodierer **516** weitergegeben und von diesen decodiert, um die entsprechende Speicherzelle zu bestimmen. Eine positive Spannung wird an das Selektions-Gate der Speicherzelle angelegt, während eine höhere positive Spannung an deren Speicher-Gate angelegt wird. Die Haltespannungsbedingungen zwischen dem Speicher-Gate und der Source der Speicherzelle programmieren wirksam ein „niedriges“ Bit innerhalb des Speichertransistors der Speicherzelle, das weiter gespeichert wird, selbst nachdem die verschiedenen Spannungen entfernt wurden.

[0031] In einem Löschen-Speichervorgang werden die Bits, die in den Speicherzellen eines gegebenen Löschsektors der Speicheranordnung **520** gespeichert sind, basierend auf in dem Löschvorgang spe-

zifizierten Adressinformationen gelöscht. Die Adressinformationen werden an den X-Decodierer **514** und den Y-Decodierer **516** weitergeleitet und von diesen decodiert, um den Ort des angestrebten Löschsektors zu bestimmen. Um jegliche Ladung, die innerhalb des Speichertransistors einer Speicherzelle gespeichert sein kann, zu „löschen“ und ihren Zustand in ein „hohes“ Bit zurückzuführen, wird eine präzise Spannung an die Source der Speicherzelle angelegt. Der Drain der Speicherzelle wird „geflutet“ (oder auf einer gewissen Haltespannung gehalten), eine präzise erzeugte hohe negative Spannung wird an deren Speicher-Gate angelegt und ihr Selektionsgatter wird geerdet. Die Haltespannungsbedingungen zwischen dem Speicher-Gate und der Source der Speicherzelle löschen den Speichertransistor wirksam auf den „hohen“ Bit-Zustand.

[0032] Entsprechend den hierin beschriebenen Techniken für die Zeilenredundanz wird ein fehlerhafter Löschsektor während eines Sortierablaufvorgangs identifiziert. Der fehlerhafte Löschsektor wird dann im Flashspeicher **520** durch einen verteilten Zeilenredundanzsektor ersetzt, z. B. durch Programmieren/Speichern der Ortsinformationen des fehlerhaften Sektors in Konfigurations-Flipflops, Registern und/oder anderer Logik. Im Gebrauch, wenn ein Lese- oder Programmierspeichervorgang für eine Adresse in dem fehlerhaften Löschsektor eingeleitet wird, dekodiert der X-Decodierer **514** die Adresse, um eine Zeile von Speicherzellen in einem entsprechenden Abschnitt des verteilten Zeilenredundanzsektors zu identifizieren. Der normale physische Sektor, welcher der identifizierten Zeile entspricht, wird bestimmt und die angeforderten Daten werden aus dem Abschnitt des Zeilenredundanzsektors, der innerhalb des physischen Sektors angeordnet ist, gelesen oder in diesen programmiert. Wenn ein Löschvorgang gegenüber dem fehlerhaften Löschsektor eingeleitet wird, dekodiert der X-Decodierer **514** die Adresse in dem Löschvorgang und bestimmt, dass er die Speicherzellen in dem verteilten Zeilenredundanzsektor löschen muss. Die Steuerlogik (z. B. in dem Flash-Controller **512**) legt dann die notwendigen Spannungen seriell oder parallel an jeden Abschnitt des Zeilenredundanzsektors an, der in jedem der physischen Sektoren angeordnet ist, und löscht dadurch die Speicherzellen in dem verteilten Zeilenredundanzsektor.

[0033] In verschiedenen Ausführungsformen führt die Verwendung des hierin beschriebenen verteilten Zeilenredundanzschemas zu einer Gesamtreduzierung der für ein Flashspeicherprodukt verwendeten Die-Fläche. Das verteilte Redundanzschema verbessert die Herstellungsausbeute und überwindet die Auswirkungen zeilenbezogener Herstellungsfehler. Das hierin beschriebene Zeilenredundanzschema stellt beispielsweise das Hinzufügen von Zeilenredundanz-Wortleitungen, -Source-Leitungen und

-Speicher-Gate-Leitungen zu mehreren physischen Sektoren einer Speicherbank bereit, wodurch die notwendige Redundanz auf normale physische Sektoren verteilt wird, anstatt einen separaten redundanten Sektor zu verwenden. Dadurch kann die Logik (z. B. Konfigurations-Flipflops, Register, synthetisierte Logik, Treiberschaltungen, usw.), die für den Betrieb der Zeilenredundanzleitungen erforderlich ist, als Teil oder zusammen mit der ähnlichen Logik, die für den Betrieb der normalen physischen Sektoren verwendet wird, angeordnet werden. Es sollte sich verstehen, dass das hierin beschriebene Zeilenredundanzschema besonders für nichtflüchtige Speicherarchitekturen (z. B. eingebettete Flashspeicher), die einen schnellen Zugriff erfordern, aber einen großen Die-Flächen-Mehraufwand vermeiden müssen, nützlich ist. Ein Beispiel für solche Architekturen ist eine Flashspeicherarchitektur, die in Halbleiterprodukten für die Automobilindustrie verwendet wird, welche im Allgemeinen hohe Fertigungsausbeuten mit geringen Fehlerraten und geringen Auswirkungen auf die Die-Fläche erfordern.

[0034] Verschiedene Ausführungsformen der hierin beschriebenen Techniken zur verteilten Zeilenredundanz im Flashspeicher können verschiedene Vorgänge umfassen. Diese Vorgänge können durch Hardwarekomponenten, Firmware oder einer Kombination daraus durchgeführt werden. Wie hierin verwendet, können die Begriffe „gekoppelt an“ und „gekoppelt mit“ direkt oder indirekt über eine oder mehrere dazwischenliegende Komponenten gekoppelt bedeuten. Jegliche der Signale, die über verschiedene hierin beschriebene Busse bereitgestellt werden, können mit anderen Signalen zeitgemultiplext und über einen oder mehrere übliche Busse bereitgestellt werden. Zusätzlich kann die Zwischenverbindung zwischen Schaltungskomponenten oder -blöcken als Busse oder als einzelne Signalleitungen gezeigt werden. Jeder der Busse kann alternativ eine oder mehrere einzelne Signalleitungen sein, und jede der einzelnen Signalleitungen kann alternativ Busse sein.

[0035] Obgleich die Vorgänge jeglichen/jeglicher Verfahren(s) hierin in einer bestimmten Reihenfolge gezeigt und beschrieben sein können, kann die Reihenfolge der Vorgänge eines jeden Verfahrens verändert werden, so dass gewisse Vorgänge in einer umgekehrten Reihenfolge durchgeführt werden können oder so dass bestimmte Vorgänge zumindest teilweise gleichzeitig mit anderen Vorgängen durchgeführt werden können. In weiteren Ausführungsformen können Anweisungen oder Teilverfahren von einzelnen Vorgängen auf intermittierende und/oder abwechselnde Weise erfolgen.

[0036] In der vorstehenden Beschreibung wurde die Erfindung anhand spezifischer beispielhafter Ausführungsformen davon beschrieben. Es ist jedoch of-

fensichtlich, dass verschiedene Modifizierungen und Veränderungen daran vorgenommen werden können, ohne dass vom breiteren Wesen und Schutzbereich der Erfindung, wie in den beigefügten Ansprüchen dargelegt, abgewichen wird. Die Beschreibung und die Zeichnungen sind dementsprechend in einem veranschaulichenden Sinn und nicht in einem einschränkenden Sinn zu betrachten.

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Diese Liste der vom Anmelder aufgeführten Dokumente wurde automatisiert erzeugt und ist ausschließlich zur besseren Information des Lesers aufgenommen. Die Liste ist nicht Bestandteil der deutschen Patent- bzw. Gebrauchsmusteranmeldung. Das DPMA übernimmt keinerlei Haftung für etwaige Fehler oder Auslassungen.

Zitierte Patentliteratur

- US 15/850779 [0001]
- US 62/551314 [0001]

Patentansprüche

1. Eine Halbleitervorrichtung, die Folgendes beinhaltet:
einen eingebetteten Flashspeicher, der eine Speicherbank beinhaltet, welche mehrere physische Sektoren umfasst, wobei jeder physische Sektor eine Vielzahl von Löschsektoren beinhaltet, und wobei: mehrere Abschnitte eines zusätzlichen Löschsektors jeweils auf die mehreren physischen Sektoren verteilt sind; und
die mehreren Abschnitte des zusätzlichen Löschsektors als Zeilenredundanzsektor für die Speicherbank konfiguriert sind.
2. Halbleitervorrichtung gemäß Anspruch 1, wobei: die Speicherbank einen fehlerhaften Löschsektor umfasst; und
die mehreren Abschnitte des zusätzlichen Löschsektors konfiguriert sind, um den fehlerhaften Löschsektor innerhalb der Speicherbank zu ersetzen.
3. Halbleitervorrichtung gemäß Anspruch 2, wobei der eingebettete Flashspeicher konfiguriert ist, um während eines Speichervorgangs auf der Speicherbank auf die mehreren Abschnitte des zusätzlichen Löschsektors statt auf den fehlerhaften Löschsektor zuzugreifen.
4. Halbleitervorrichtung gemäß Anspruch 3, wobei der Speichervorgang ein Lesevorgang oder ein Programmervorgang ist.
5. Halbleitervorrichtung gemäß Anspruch 3, wobei der Speichervorgang ein Löschvorgang ist, der seriell auf den mehreren physischen Sektoren durchgeführt wird, um jeden der mehreren Abschnitte des zusätzlichen Löschsektors zu löschen.
6. Halbleitervorrichtung gemäß Anspruch 3, wobei der Speichervorgang ein Löschvorgang ist, der parallel auf den mehreren physischen Sektoren durchgeführt wird, um jeden der mehreren Abschnitte des zusätzlichen Löschsektors zu löschen.
7. Halbleitervorrichtung gemäß Anspruch 2, wobei der fehlerhafte Löschsektor während eines Sortierablaufvorgangs auf dem eingebetteten Flashspeicher bestimmt wird.
8. Halbleitervorrichtung gemäß Anspruch 1, wobei die Halbleitervorrichtung Folgendes beinhaltet:
erste Wortleitungen (WLs), die an die ersten Speicherzellen in den mehreren physischen Sektoren gekoppelt sind; und
zweite WLs, die zusätzlich zu den ersten WLs auf die mehreren physischen Sektoren verteilt sind, wobei die zweiten WLs an zweite Speicherzellen in den mehreren Abschnitten des zusätzlichen Löschsektors gekoppelt sind.

9. Halbleitervorrichtung gemäß Anspruch 1, wobei die Speicherbank zusätzlich zu den mehreren physischen Sektoren einen oder mehrere physische Sektoren, die keinen Abschnitt des zusätzlichen Löschsektors umfassen, beinhaltet.
10. Ein System, das Folgendes beinhaltet:
eine Mikrocontrollereinheit; und
einen in der Mikrocontrollereinheit eingebetteten Flashspeicher, wobei der Flashspeicher Folgendes beinhaltet:
eine Speicherbank, die mehrere physische Sektoren umfasst, wobei jeder physische Sektor eine Vielzahl von Löschsektoren beinhaltet, und wobei:
mehrere Abschnitte eines zusätzlichen Löschsektors jeweils auf die mehreren physischen Sektoren verteilt sind; und
die mehreren Abschnitte des zusätzlichen Löschsektors als Zeilenredundanzsektor für die Speicherbank konfiguriert sind.
11. System gemäß Anspruch 10, wobei:
die Speicherbank einen fehlerhaften Löschsektor umfasst; und
die mehreren Abschnitte des zusätzlichen Löschsektors konfiguriert sind, um den fehlerhaften Löschsektor innerhalb der Speicherbank zu ersetzen.
12. System gemäß Anspruch 11, wobei der Flashspeicher konfiguriert ist, um während eines Speichervorgangs auf der Speicherbank auf die mehreren Abschnitte des zusätzlichen Löschsektors statt auf den fehlerhaften Löschsektor zuzugreifen.
13. System gemäß Anspruch 12, wobei der Speichervorgang ein Lesevorgang oder ein Programmervorgang ist.
14. System gemäß Anspruch 12, wobei der Speichervorgang ein Löschvorgang ist, der seriell auf den mehreren physischen Sektoren durchgeführt wird, um jeden der mehreren Abschnitte des zusätzlichen Löschsektors zu löschen.
15. System gemäß Anspruch 12, wobei der Speichervorgang ein Löschvorgang ist, der parallel auf den mehreren physischen Sektoren durchgeführt wird, um jeden der mehreren Abschnitte des zusätzlichen Löschsektors zu löschen.
16. Ein Verfahren zum Konfigurieren eines eingebetteten Flashspeichers mit Zeilenredundanz, wobei das Verfahren Folgendes beinhaltet:
Bereitstellen von ersten Wortleitungen (WLs), die in mehreren physischen Sektoren einer Speicherbank in dem eingebetteten Flashspeicher angeordnet sind, wobei jeder physische Sektor eine Vielzahl von Löschsektoren beinhaltet; und
Bereitstellen von zweiten WLs, die zusätzlich zu den ersten WLs in den mehreren physischen Sektoren

angeordnet sind, wobei die zweiten WLs an mehrere Abschnitte eines zusätzlichen Löschsektors gekoppelt sind, die jeweils auf die mehreren physischen Sektoren verteilt sind, und wobei der zusätzliche Löschsektor als ein Zeilenredundanzsektor für die Speicherbank konfiguriert ist.

17. Verfahren gemäß Anspruch 16, das ferner Folgendes beinhaltet:

Bestimmen, dass die Speicherbank einen fehlerhaften Löschsektor umfasst; und

Konfigurieren des eingebetteten Flashspeichers, um den fehlerhaften Löschsektor durch die mehreren Abschnitte des zusätzlichen Löschsektors in der Speicherbank zu ersetzen.

18. Verfahren gemäß Anspruch 17, das ferner das Konfigurieren des eingebetteten Flashspeichers für einen Speichervorgang, der auf die mehreren Abschnitte des zusätzlichen Löschsektors statt auf den fehlerhaften Löschsektor zugreifen soll, beinhaltet.

19. Verfahren gemäß Anspruch 18, wobei der Speichervorgang ein Lesevorgang oder ein Programmervorgang ist.

20. Verfahren gemäß Anspruch 17, das ferner das Konfigurieren des eingebetteten Flashspeichers für einen Löschtorgang, der seriell auf den mehreren physischen Sektoren durchgeführt werden soll, um jeden der mehreren Abschnitte des zusätzlichen Löschsektors zu löschen, beinhaltet.

Es folgen 5 Seiten Zeichnungen

Anhängende Zeichnungen

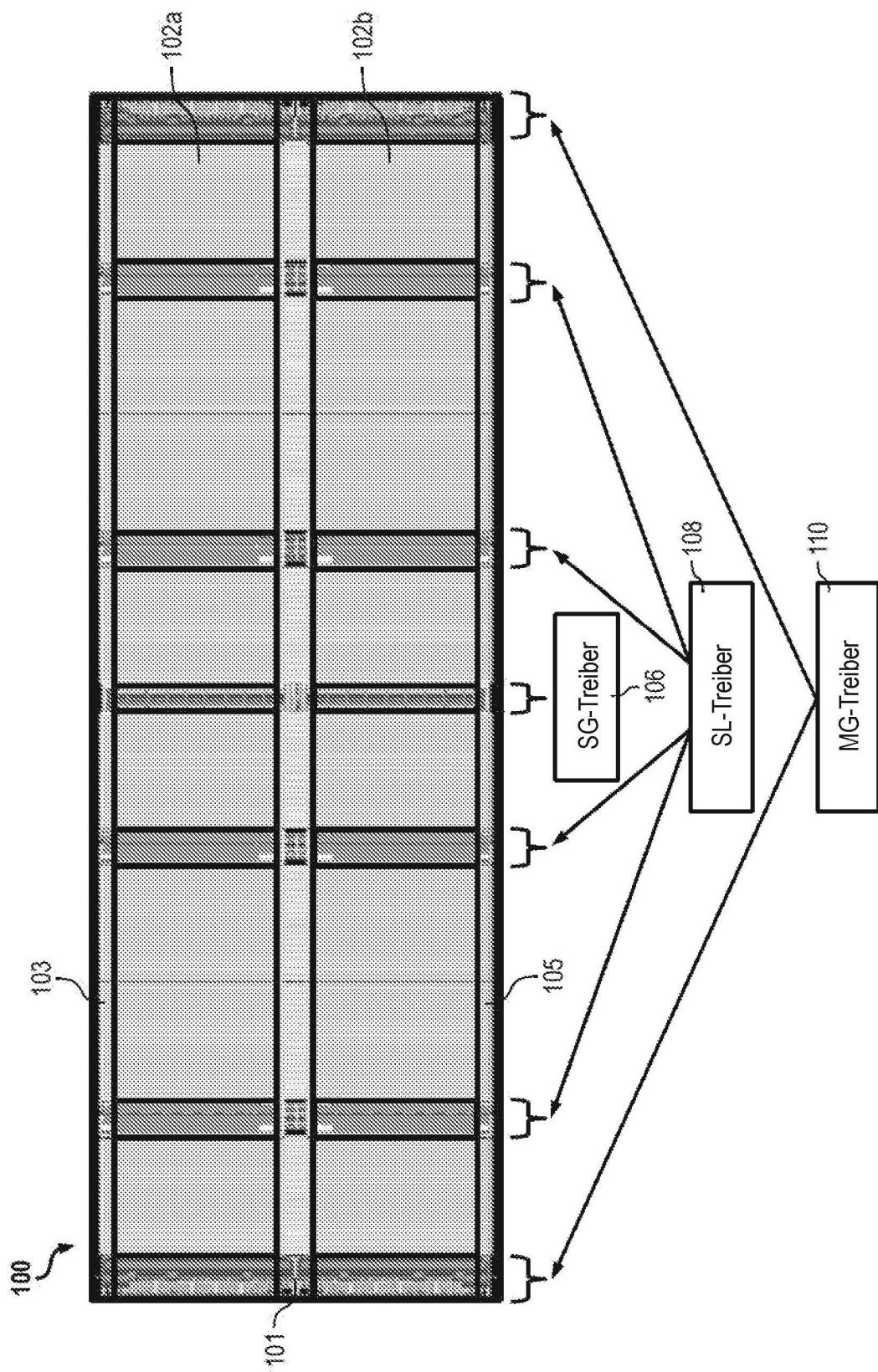


FIG. 1

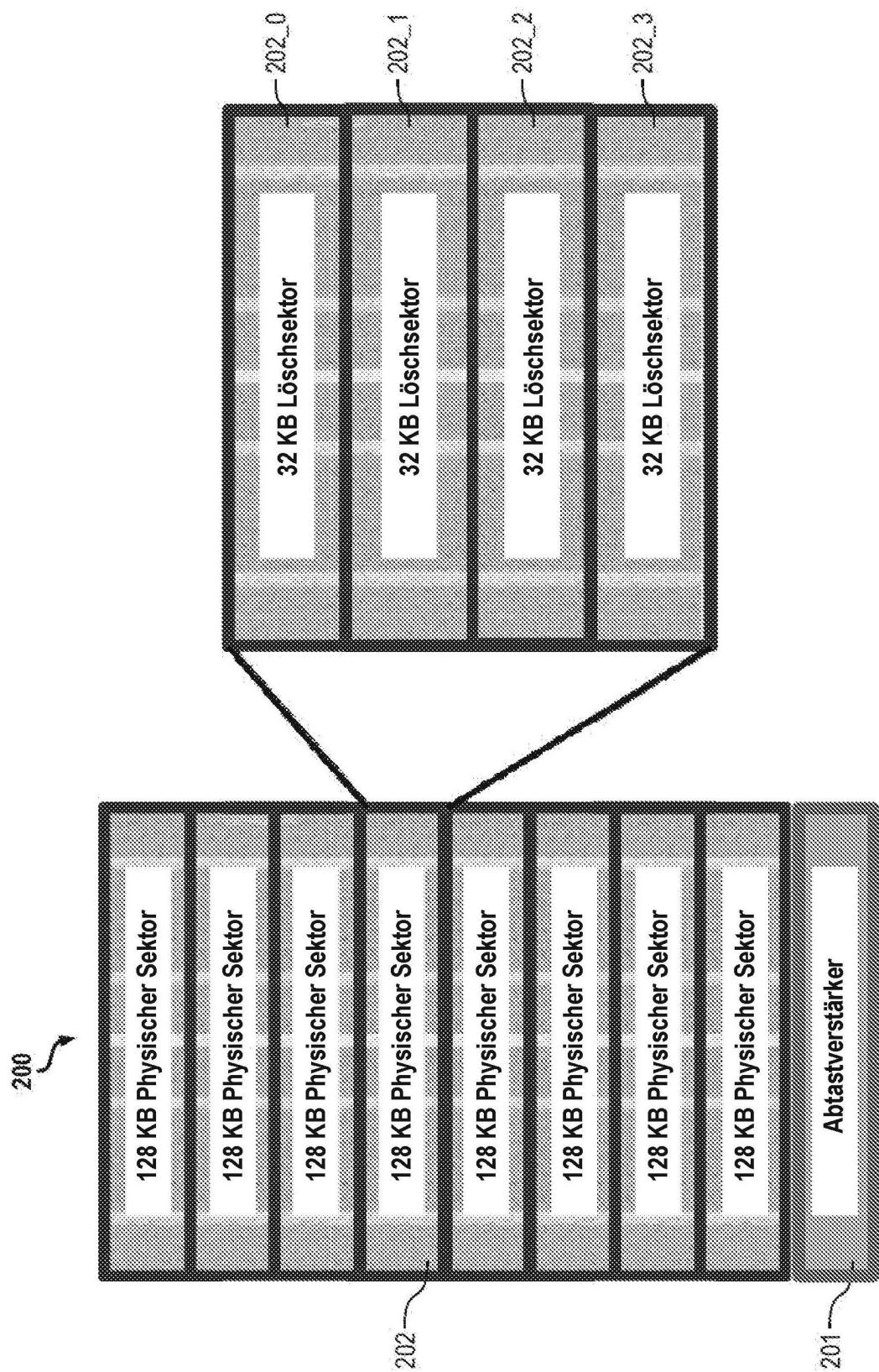


FIG. 2

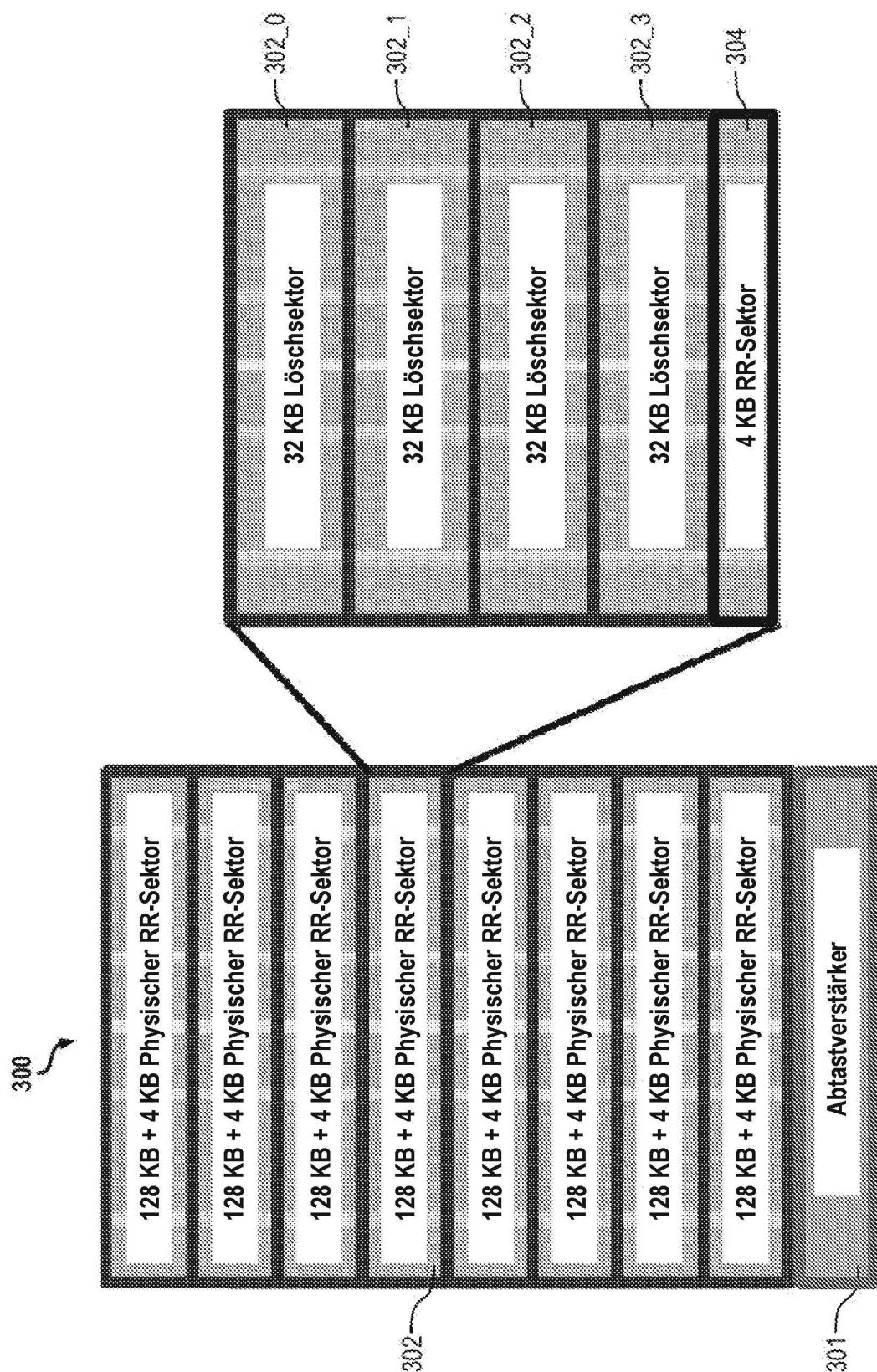


FIG. 3

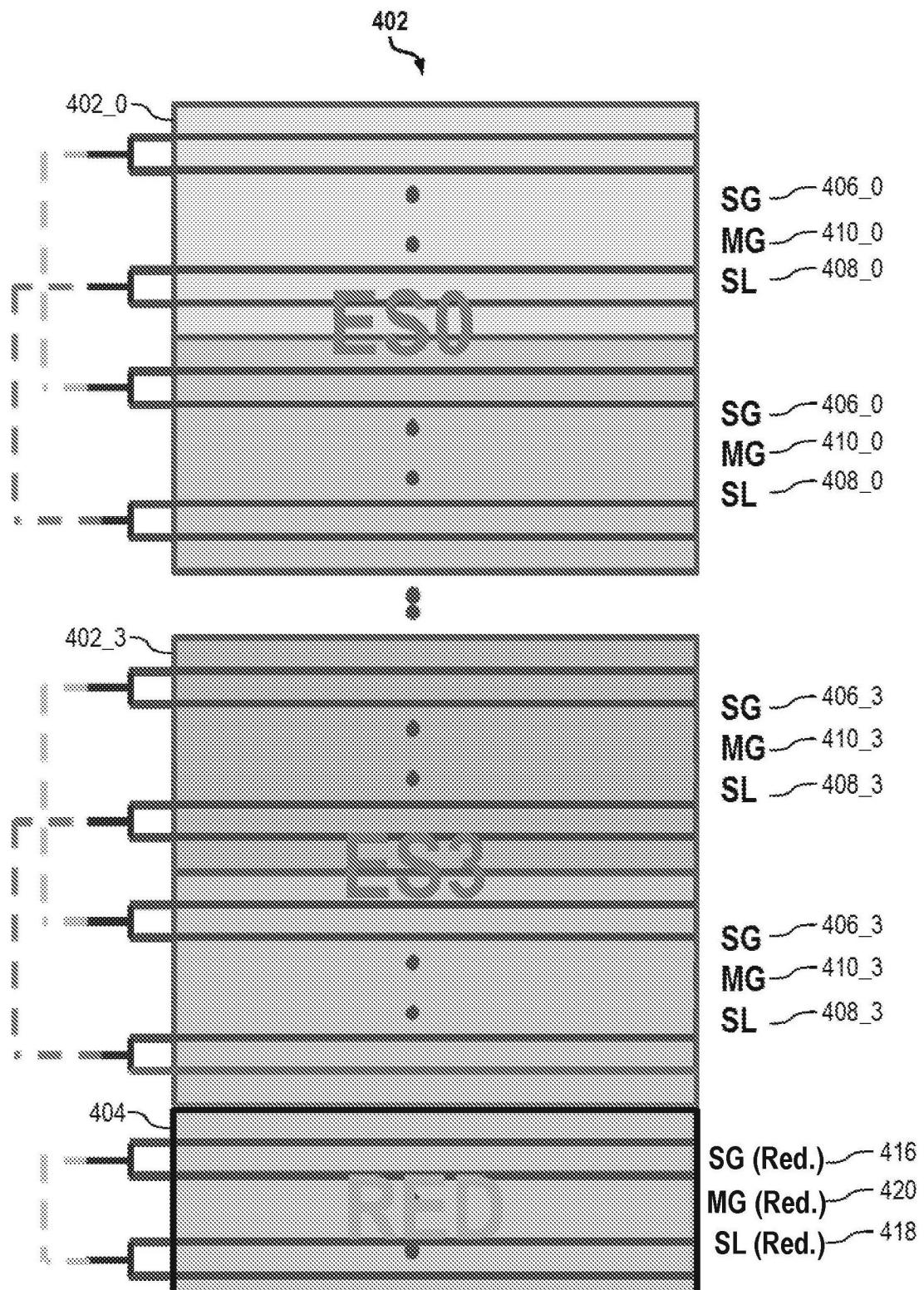


FIG. 4

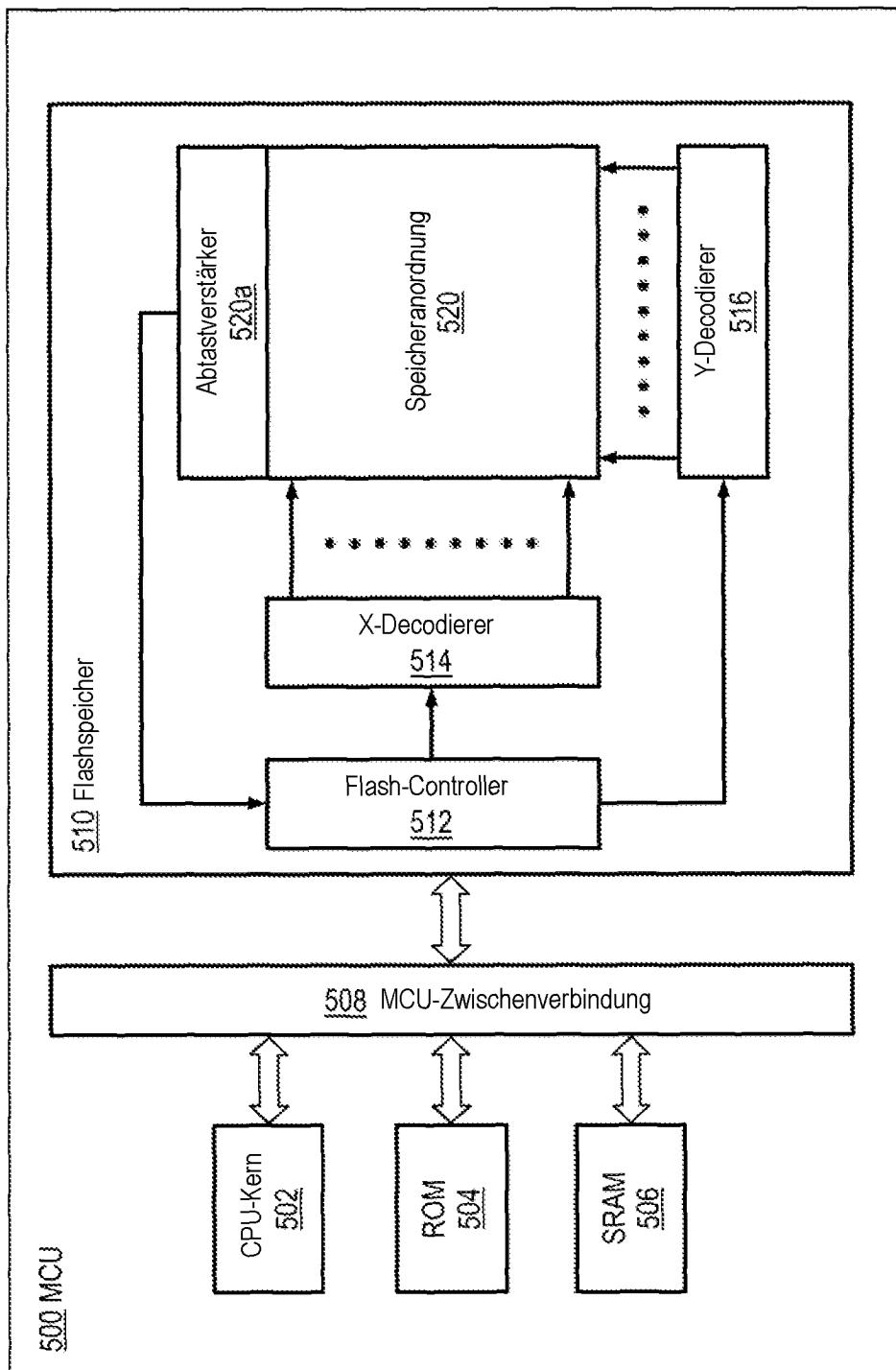


FIG. 5