



(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2023 118 085.1**

(51) Int Cl.: **G06F 12/02 (2006.01)**

(22) Anmeldetag: **07.07.2023**

(43) Offenlegungstag: **29.02.2024**

(30) Unionspriorität:
17/897,854 **29.08.2022** **US**

(71) Anmelder:
SanDisk Technologies LLC, Addison, TX, US

(74) Vertreter:
Murgitroyd Germany Patentanwaltsgesellschaft mbH, 80636 München, DE

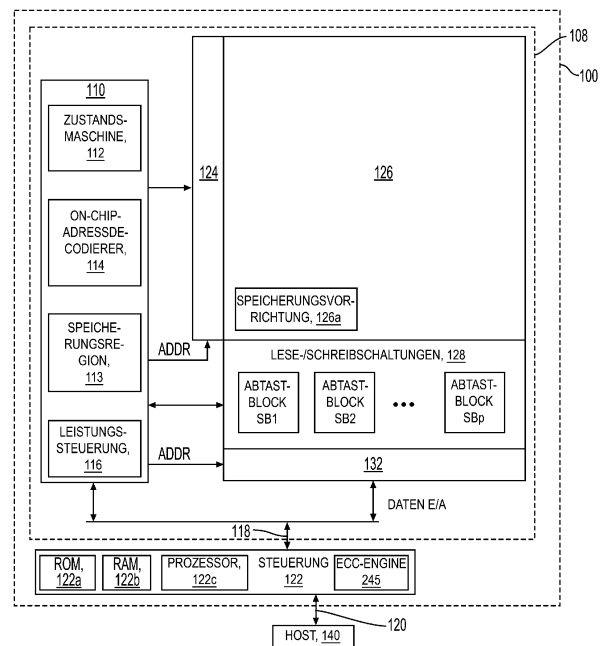
(72) Erfinder:
Yang, Xiang, San Jose, CA, US; Guo, Jiachen, San Jose, CA, US; Mukherjee, Shubhajit, San Jose, CA, US

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **TECHNIKEN ZUM SCHREIBEN AM GLEICHEN ORT OHNE LÖSCHEN IN EINER SPEICHERVORRICHTUNG**

(57) Zusammenfassung: Die Technik schließt eine Speichervorrichtung ein, die eine Datenschreibenweisung empfängt. Die Speichervorrichtung programmiert die Speicherzellen der Speicherblöcke in ein Format mit einem Bit pro Speicherzelle (SLC) mit einem ersten und einem zweiten SLC-Datenzustand. Als Reaktion darauf, dass die Daten, die in die Speicherzellen der Speicherblöcke programmiert werden, vor dem Abschluss der Datenschreibenweisung eine SLC-Grenze erreichen, ohne die Speicherzellen zu löschen, programmiert die Speichervorrichtung mindestens einige der Speicherzellen aus dem SLC-Format in ein Format mit zwei Bits pro Speicherzelle (MLC). Beim Programmieren aus dem SLC-Format in das MLC-Format verhindert die Speichervorrichtung die Programmierung einiger der Speicherzellen in dem ersten und zweiten SLC-Datenzustand, um einen ersten MLC-Datenzustand zu bilden, und programmiert andere Speicherzellen der SLC-Datenzustände, um einen zweiten, dritten und vierten MLC-Datenzustand zu bilden.



Beschreibung

HINTERGRUND

1. Gebiet

[0001] Die vorliegende Offenbarung bezieht sich allgemein auf Speichervorrichtungen und insbesondere auf Techniken zum Programmieren von Speichervorrichtungen.

2. Stand der Technik

[0002] Die Verwendung von Halbleiterspeichern in verschiedenen elektronischen Vorrichtungen, wie Mobiltelefonen, Digitalkameras, PDAs, medizinischer Elektronik, mobilen Rechenvorrichtungen, Servern, Solid-State-Laufwerken, nicht mobilen Rechenvorrichtungen und anderen Vorrichtungen, ist weit verbreitet. Ein Halbleiterspeicher kann einen nichtflüchtigen Speicher oder einen flüchtigen Speicher aufweisen. Ein nichtflüchtiger Speicher ermöglicht, dass Informationen gespeichert und behalten werden, selbst wenn der nichtflüchtige Speicher nicht mit einer Stromquelle, z. B. einer Batterie, verbunden ist.

[0003] Eine NAND-Speichervorrichtung schließen ein Array von Speicherzellen ein, die in einer Vielzahl von Wortleitungen angeordnet sind und eingerichtet sind, um in ein oder mehrere Datenbits pro Speicherzelle programmiert zu werden. Bei einigen Speichervorrichtungen kann die Speichervorrichtung eingerichtet sein, um in zwei verschiedenen Betriebsmodi zu arbeiten, einschließlich eines Hochleistungsmodus, in dem Speicherzellen in ein einzelnes Datenbit pro Speicherzellen programmiert werden, und eines Niederleistungsmodus, in dem mehrere Datenbits in jede Speicherzelle programmiert werden.

KURZDARSTELLUNG

[0004] Ein Gesichtspunkt der vorliegenden Offenbarung bezieht sich auf ein Verfahren zum Programmieren einer Speichervorrichtung. Das Verfahren schließt den Schritt des Vorbereitens einer Vielzahl von Speicherblöcken ein. Die Speicherblöcke schließen ein Array von Speicherzellen ein, die programmiert werden können, um ein oder mehrere Datenbits pro Speicherzelle zu halten. Das Verfahren fährt mit dem Schritt des Empfangens einer Datenschreibanweisung fort. Das Verfahren fährt mit dem Schritt des Programmierens der Speicherzellen der Speicherblöcke in ein Format mit einem Bit pro Speicherzelle (SLC) mit einem ersten SLC-Datenzustand und einem zweiten SLC-Datenzustand fort. Als Reaktion darauf, dass die Daten, die im SLC-Format in die Speicherzellen der Speicherblöcke programmiert werden, vor dem Abschluss der Datenschreibanweisung eine SLC-Grenze erreichen, ohne die in

das SLC-Format programmierten Speicherzellen zu löschen, fährt das Verfahren mit dem Schritt des Programmierens von mindestens einigen der Vielzahl von Speicherzellen aus dem SLC-Format in ein Format mit zwei Bits pro Speicherzelle (MLC) fort. Das Programmieren der Speicherzellen aus dem SLC-Format in das MLC-Format schließt ein Hemmen der Programmierung einiger der Speicherzellen in dem ersten und zweiten SLC-Datenzustand ein, um einen ersten MLC-Datenzustand zu bilden, und das Programmieren einiger der Speicherzellen des ersten und des zweiten SLC-Datenzustands, um einen zweiten MLC-Datenzustand, einen dritten MLC-Datenzustand und einen vierten MLC-Datenzustand zu bilden.

[0005] Gemäß einem anderen Gesichtspunkt der vorliegenden Offenbarung schließt das Verfahren als Reaktion darauf, dass eine MLC-Grenze erreicht ist, weiterhin den Schritt des Programmierens der Speicherzellen von mindestens einigen der Vielzahl von Speicherblöcken aus dem MLC-Format in ein Format mit drei oder mehr Bits pro Speicherzelle ein.

[0006] Gemäß noch einem anderen Gesichtspunkt der vorliegenden Offenbarung wird vor dem Programmieren der Speicherzellen von mindestens einem der Speicherblöcke aus dem MLC-Format in das Format mit drei oder mehr Bits pro Speicherzelle der mindestens eine Speicherblock gelöscht.

[0007] Gemäß noch einem weiteren Gesichtspunkt der vorliegenden Offenbarung ist das Format mit drei oder mehr Bits pro Speicherzelle ein das Format mit vier Bits pro Speicherzelle (QLC).

[0008] Gemäß einem weiteren Gesichtspunkt der vorliegenden Offenbarung ist die SLC-Grenze erreicht, wenn die Speicherzellen aller verfügbaren Speicherblöcke in das SLC-Format programmiert sind.

[0009] Gemäß noch einem weiteren Gesichtspunkt der vorliegenden Offenbarung ist die MLC-Grenze erreicht, wenn die Speicherzellen aller verfügbaren Speicherblöcke in das MLC-Format programmiert sind.

[0010] Gemäß noch einem weiteren Gesichtspunkt der vorliegenden Offenbarung schließt das Verfahren weiterhin den Schritt des Programmierens jeglicher Daten im SLC-Format oder im MLC-Format in das Format mit drei oder mehr Bits pro Speicherzelle in einem Hintergrundvorgang ein.

[0011] Gemäß einem anderen Gesichtspunkt der vorliegenden Offenbarung schließt der Schritt des Programmierens der Speicherzellen von mindestens einigen der Speicherblöcke aus dem SLC-Format in das MLC-Format eine Vielzahl von Programmier-

schleifen ein, die auf jede Wortleitung angewendet werden, die die Speicherzellen enthält, die programmiert werden. Die Programmierschleifen schließen jeweils einen Programmierimpuls und mindestens einen Verifizierimpuls ein.

[0012] Gemäß noch einem anderen Gesichtspunkt der vorliegenden Offenbarung schließt der Schritt des Programmierens der Speicherzellen von mindestens einigen der Speicherblöcke aus dem SLC-Format in das MLC-Format drei Programmierimpulse und keine Verifizierimpulse ein, die an jede Wortleitung angelegt werden, die Speicherzellen enthält, die programmiert werden.

[0013] Ein anderer Gesichtspunkt der vorliegenden Offenbarung bezieht sich auf eine Speichervorrichtung. Die Speichervorrichtung schließt eine Vielzahl von Speicherblöcken ein, die jeweils eine Vielzahl von Speicherzellen einschließen, die in einer Vielzahl von Wortleitungen angeordnet sind. Die Speichervorrichtung schließt eine Steuerschaltlogik ein, die in Kommunikation mit der Vielzahl von Speicherblöcken ist. Die Steuerschaltlogik ist eingerichtet, um eine Datenschreibanweisung zu empfangen. Die Steuerschaltlogik ist auch eingerichtet, um die Speicherzellen der Speicherblöcke in ein SLC-Format mit einem ersten SLC-Datenzustand und einem zweiten SLC-Datenzustand zu programmieren. Als Reaktion darauf, dass die Daten, die im SLC-Format in die Speicherzellen der Speicherblöcke programmiert werden, vor dem Abschluss der Datenschreibanweisung eine SLC-Grenze erreichen, ist die Steuerschaltlogik eingerichtet, um ohne die in das SLC-Format programmierten Speicherzellen zu löschen, mindestens einige der Speicherzellen aus dem SLC-Format in ein MLC-Format zu programmieren. Wenn die Speicherzellen aus dem SLC-Format in das MLC-Format programmiert werden, ist die Steuerschaltlogik eingerichtet, um die Programmierung einiger der Speicherzellen in dem ersten und zweiten SLC-Datenzustand zu verhindern, um einen ersten MLC-Datenzustand zu bilden und einige der Speicherzellen des ersten und des zweiten SLC-Datenzustands zu programmieren, um einen zweiten MLC-Datenzustand, einen dritten MLC-Datenzustand und einen vierten MLC-Datenzustand zu bilden.

[0014] Gemäß einem anderen Gesichtspunkt der vorliegenden Offenbarung ist die Steuerschaltlogik als Reaktion darauf, dass eine MLC-Grenze erreicht ist, weiterhin eingerichtet, um die Speicherzellen von mindestens einigen der Vielzahl von Speicherblöcken aus dem MLC-Format in ein Format mit drei oder mehr Bits pro Speicherzelle zu programmieren.

[0015] Gemäß noch einem anderen Gesichtspunkt der vorliegenden Offenbarung löscht die Steuerschaltlogik vor dem Programmieren der Speicherzel-

len von mindestens einem der Speicherblöcke aus dem MLC-Format in das Format mit drei oder mehr Bits pro Speicherzelle den Speicherblock.

[0016] Gemäß noch einem anderen Gesichtspunkt der vorliegenden Offenbarung ist das Format mit drei oder mehr Bits pro Speicherzelle ein QLC-Format.

[0017] Gemäß einem weiteren Gesichtspunkt der vorliegenden Offenbarung ist die SLC-Grenze erreicht, wenn die Speicherzellen aller verfügbaren Speicherblöcke in das SLC-Format programmiert sind.

[0018] Gemäß noch einem weiteren Gesichtspunkt der vorliegenden Offenbarung ist die MLC-Grenze erreicht, wenn die Speicherzellen aller verfügbaren Speicherblöcke in das MLC-Format programmiert sind.

[0019] Gemäß noch einem weiteren Gesichtspunkt der vorliegenden Offenbarung ist die Steuerschaltlogik eingerichtet, um in einem Hintergrundvorgang alle Daten im SLC-Format oder im MLC-Format in das Format mit drei oder mehr Bits pro Speicherzelle zu programmieren.

[0020] Gemäß einem anderen Gesichtspunkt der vorliegenden Offenbarung ist die Steuerschaltlogik beim Programmieren der Speicherzellen von mindestens einigen der Speicherblöcke aus dem SLC-Format in das MLC-Format eingerichtet, um eine Vielzahl von Programmierschleifen auf jede Wortleitung anzuwenden, die die Speicherzellen enthält, die programmiert werden. Die Programmierschleifen schließen jeweils einen Programmierimpuls und mindestens einen Verifizierimpuls ein.

[0021] Gemäß noch einem anderen Gesichtspunkt der vorliegenden Offenbarung schließt das Programmieren von mindestens einigen der Speicherblöcke aus dem SLC-Format in das MLC-Format ein, dass die Steuerschaltlogik zwei Programmierimpulse und keine Verifizierimpulse an jede Wortleitung anlegt, die die Speicherzellen enthält, die programmiert werden.

[0022] Noch ein weiterer Gesichtspunkt der vorliegenden Offenbarung bezieht sich auf eine Einrichtung, die eine Speichervorrichtung mit einer Vielzahl von Speicherblöcken einschließt. Jeder Speicherblock schließt eine Vielzahl von Speicherzellen ein, die in einer Vielzahl von Wortleitungen angeordnet sind. Die Einrichtung schließt außerdem ein Programmiermittel zum Programmieren von Daten in die Speicherzellen der Speicherblöcke ein. Das Programmiermittel ist eingerichtet, um eine Datenschreibanweisung zu empfangen. Das Programmiermittel ist außerdem eingerichtet, um die

Speicherzellen der Speicherblöcke in ein SLC-Format mit dem ersten und dem zweiten SLC-Datenzustand zu programmieren. Als Reaktion darauf, dass die Daten, die im SLC-Format in die Speicherzellen der Speicherblöcke programmiert werden, vor dem Abschluss der Datenschreibanweisung einen SLC-Grenzwert erreichen, ohne die in das SLC-Format programmierten Speicherzellen zu löschen, ist das Programmiermittel eingerichtet, um die Programmierung einiger Speicherzellen zu verhindern, um einen ersten MLC-Datenzustand zu bilden, einige Speicherzellen des zweiten SLC-Datenzustands in einen zweiten MLC-Datenzustand und einen vierten MLC-Datenzustand zu programmieren und einige Speicherzellen des ersten SLC-Datenzustands zu programmieren, um einen dritten MLC-Datenzustand zu bilden.

[0023] Gemäß einem anderen Gesichtspunkt der vorliegenden Offenbarung legt das Programmiermittel beim Programmieren der Speicherzellen aus dem SLC-Format in das MLC-Format drei Programmierimpulse und null Verifizierimpulse an eine ausgewählte Wortleitung der Vielzahl von Wortleitungen an.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0024] Eine detailliertere Beschreibung wird nachstehend unter Bezugnahme auf beispielhafte Ausführungsformen dargelegt, die in den beigefügten Figuren dargestellt sind. Es versteht sich, dass diese Figuren nur beispielhafte Ausführungsformen der Offenbarung darstellen und daher nicht als Einschränkung ihres Umfangs angesehen werden sollen. Die Offenbarung wird mit zusätzlicher Spezifität und Details durch die Verwendung der beigefügten Zeichnungen beschrieben und erläutert, in denen:

Fig. 1A ein Blockdiagramm einer beispielhaften Speichervorrichtung ist;

Fig. 1 B ein Blockdiagramm einer beispielhaften Steuerschaltung ist;

Fig. 2 Blöcke von Speicherzellen in einer beispielhaften zweidimensionalen Konfiguration des Speicherarrays von **Fig. 1A** darstellt;

Fig. 3A und **Fig. 3B** Querschnittansichten von beispielhaftem Floating-Gate mit NAND-Ketten darstellen;

Fig. 4A und **Fig. 4B** Querschnittansichten von beispielhaften ladungseinfangenden Speicherzellen in NAND-Ketten darstellen;

Fig. 5 ein beispielhaftes Blockdiagramm eines Erfassungsblocks SB1 von **Fig. 1** darstellt;

Fig. 6A ist eine perspektivische Ansicht eines Satzes von Blöcken in einer beispielhaften dreidimensionalen Konfiguration des Speicherarrays von **Fig. 1**;

Fig. 6B eine beispielhafte Querschnittsansicht eines Abschnitts von einem der Blöcke von **Fig. 6A** darstellt;

Fig. 6C eine grafische Darstellung des Speicherlochdurchmessers im Stapel von **Fig. 6B** darstellt;

Fig. 6D eine Nahansicht der Region 622 des Stapels von **Fig. 6B** darstellt;

Fig. 7A eine Draufsicht auf eine beispielhafte Wortleitungsschicht WLL0 des Stapels von **Fig. 6B** darstellt;

Fig. 7B eine Draufsicht auf eine beispielhafte obere dielektrische Schicht DL116 des Stapels von **Fig. 6B** darstellt;

Fig. 8 ein Schwellenwertverteilungsdiagramm einer Vielzahl von Speicherzellen ist, die in ein SLC-Format programmiert werden (ein Bit pro Speicherzelle);

Fig. 9 ein Schwellenwertverteilungsdiagramm einer Vielzahl von Speicherzellen ist, die in ein MLC-Format programmiert werden (zwei Bits pro Speicherzelle);

Fig. 10 ein Schwellenwertverteilungsdiagramm einer Vielzahl von Speicherzellen ist, die in ein QLC-Format programmiert werden (vier Bits pro Speicherzelle);

Fig. 11 ein Diagramm von Leistung im Vergleich zu Daten, die während eines beispielhaften Programmiervorgangs in eine Speichervorrichtung geschrieben werden, ist;

Fig. 12 ein Diagramm von Leistung im Vergleich zu Daten, die während eines anderen beispielhaften Programmiervorgangs in eine Speichervorrichtung geschrieben werden, ist;

Fig. 13 ein Flussdiagramm ist, das die Schritte des Programmierens einer Speichervorrichtung gemäß einer beispielhaften Ausführungsform darstellt;

Fig. 14A ein Schwellenspannungsverteilungsdiagramm einer Vielzahl von Speicherzellen vor Beginn eines Programmiervorgangs ist;

Fig. 14B ein Schwellenspannungsverteilungsdiagramm der Vielzahl von Speicherzellen nach einem SLC-Programmiervorgang ist;

Fig. 14C ein Schwellenspannungsverteilungsdiagramm der Vielzahl von Speicherzellen während eines Programmiervorgangs, bei dem die Speicherzellen von einem SLC-Format in ein MLC-Format programmiert werden, gemäß einer ersten Ausführungsform ist;

Fig. 14D ein Schwellenspannungsverteilungsdiagramm der Vielzahl von Speicherzellen ist,

die gemäß der ersten Ausführungsform in ein MLC-Format programmiert werden;

Fig. 15A ein Schwellenspannungsverteilungsdiagramm einer Vielzahl von Speicherzellen vor Beginn eines Programmiervorgangs ist;

Fig. 15B ein Schwellenspannungsverteilungsdiagramm der Vielzahl von Speicherzellen nach einem SLC-Programmiervorgang ist;

Fig. 15C ein Schwellenspannungsverteilungsdiagramm der Vielzahl von Speicherzellen nach einem ersten Programmierimpuls eines 3P0V-Programmiervorgangs zum Programmieren der Speicherzellen aus dem SLC-Format in das MLC-Format gemäß einer zweiten Ausführungsform ist;

Fig. 15D ein Schwellenspannungsverteilungsdiagramm der Vielzahl von Speicherzellen nach einem zweiten Programmierimpuls eines 3P0V-Programmiervorgangs zum Programmieren der Speicherzellen aus dem SLC-Format in das MLC-Format gemäß der zweiten Ausführungsform ist;

Fig. 15E ein Schwellenspannungsverteilungsdiagramm der Vielzahl von Speicherzellen nach einem dritten Programmierimpuls eines 3P0V-Programmiervorgangs zum Programmieren der Speicherzellen aus dem SLC-Format in das MLC-Format gemäß der zweiten Ausführungsform ist; und

Fig. 15F ein Schwellenspannungsverteilungsdiagramm der Vielzahl von Speicherzellen ist, die gemäß der zweiten Ausführungsform in ein MLC-Format programmiert werden.

BESCHREIBUNG DER UNTERSTÜTZENDEN AUSFÜHRUNGSFORMEN

[0025] Die vorliegende Offenbarung bezieht sich auf eine hybride Programmierertechnik, wobei die Speichervorrichtung in einem Hochleistungs-SLC-Modus arbeitet, wenn eine anfängliche Schreib Anforderung empfangen wird. Wenn die Speichervorrichtung eine SLC-Grenze erreicht, bevor der Schreibvorgang abgeschlossen ist, dann schaltet die Speichervorrichtung in einen Mittelleistungs-MLC-Modus um, und die Programmierung erfolgt durch Schreiben zusätzlicher Daten in die Speicherzellen ohne Löschung. Mit anderen Worten programmiert die Speichervorrichtung ohne Löschung zusätzliche Daten zu den Daten im SLC-Format. Dies wird erreicht, indem die Speicherzellen direkt aus den SLC-Datenzuständen in drei zusätzliche Datenzustände in einem sogenannten „Überspring“-Programmiervorgang programmiert werden, da während der MLC-Programmierung die Schwellenspannungen einiger der Speicherzellen bestimmte Datenzustände „überspringen“, um die Schwellenspannun-

gen ihrer endgültigen Datenzustände zu erreichen, wie nachstehend ausführlicher erörtert. Wenn die Speichervorrichtung eine MLC-Grenze erreicht, bevor der Schreibvorgang abgeschlossen ist, dann schaltet die Speichervorrichtung in einen QLC-Modus um, um den Programmiervorgang abzuschließen.

[0026] Fig. 1A ist ein Blockdiagramm einer beispielhaften Speichervorrichtung 100, die eingerichtet ist, um die vorstehend genannten Programmierertechniken durchzuführen. Die Speichervorrichtung 100 schließt einen Speicherchip 108 mit einer Speicherstruktur 126 von Speicherzellen, wie ein Array von Speicherzellen, eine Steuerschaltlogik 110 und Lese-/Schreibschaltungen 128, ein. Die Speicherstruktur 126 ist durch Wortleitungen über einen Zeilendecodierer 124 und durch Bitleitungen über einen Spaltendecodierer 132 adressierbar. Die Lese-/Schreibschaltungen 128 schließen mehrere Erfassungsblöcke SB1, SB2, ... SBp (Erfassungsschaltkreis) ein und ermöglichen das parallele Lesen oder Programmieren einer Seite von Speicherzellen. Üblicherweise ist eine Steuerung 122 in der gleichen Speichervorrichtung 100 (z. B. einer entfernbaren Speicherkarte) wie das eine oder die mehreren Speicherchips 108 enthalten. Befehle und Daten werden zwischen dem Host 140 und der Steuerung 122 über einen Datenbus 120 und zwischen der Steuerung und dem einen oder den mehreren Speicherchips 108 über Leitungen 118 übertragen.

[0027] Die Speicherstruktur 126 kann zweidimensional oder dreidimensional sein. Die Speicherstruktur 126 kann ein oder mehrere Arrays von Speicherzellen aufweisen, einschließlich eines dreidimensionalen Arrays. Die Speicherstruktur 126 kann eine monolithische dreidimensionale Speicherstruktur aufweisen, in der mehrere Speicherebenen über (und nicht in) einem einzelnen Substrat wie beispielsweise einem Wafer ohne dazwischen liegende Substrate ausgebildet sind. Die Speicherstruktur 126 kann jede Art von nichtflüchtigem Speicher aufweisen, der monolithisch in einer oder mehreren physischen Ebenen von Arrays von Speicherzellen ausgebildet ist, die über einen aktiven Bereich verfügen, der über einem Siliziumsubstrat angeordnet ist. Die Speicherstruktur 126 kann sich in einer nichtflüchtigen Speichervorrichtung befinden, die Schaltungen aufweist, die dem Betrieb der Speicherzellen zugeordnet sind, ganz gleich, ob sich die zugehörige Schaltung oberhalb oder innerhalb des Substrats befindet.

[0028] Die Steuerschaltlogik 110 arbeitet mit den Lese-/Schreibschaltungen 128 zusammen, um Speichervorgänge auf der Speicherstruktur 126 auszuführen, und schließt eine Zustandsmaschine 112, einen On-Chip-Adressdecodierer 114 und ein Leistungssteuerungsmodul 116 ein. Die Zustandsma-

schine 112 stellt eine Steuerung von Speichervorgängen auf Chipebene bereit.

[0029] Eine Speicherungsregion 113 kann zum Beispiel für Programmierparameter bereitgestellt werden. Die Programmierparameter können eine Programmspannung, eine Vorspannung für die Programmspannung, Positionsparameter, die die Positionen der Speicherzellen angeben, Parameter für die Dicke der Kontaktleitungsverbinder, eine Verifizierungsspannung und/oder dergleichen einschließen. Die Positionsparameter können eine Position einer Speicherzelle innerhalb des gesamten Arrays von NAND-Ketten, eine Position einer Speicherzelle innerhalb einer bestimmten NAND-Ketten-Gruppe, eine Position einer Speicherzelle in einer bestimmten Ebene und/oder dergleichen angeben. Die Parameter für die Dicke des Kontaktleitungsverbinders können die Dicke des Kontaktleitungsverbinders, das Substrat oder das Material, aus dem der Kontaktleitungsverbinder besteht, und/oder dergleichen angeben.

[0030] Der On-Chip-Adressdecodierer 114 stellt eine Adressschnittstelle zwischen derjenigen, die von dem Host oder einer Speichersteuerung verwendet wird, zu der Hardwareadresse, die von den Decodierern 124 und 132 verwendet wird, bereit. Das Leistungssteuermodul 116 steuert die Leistung und Spannungen, die den Wortleitungen und Bitleitungen während Speichervorgängen zugeführt werden. Es kann Treiber für Wortleitungen, SGS- und SGD-Transistoren und Sourceleitungen einschließen. Die Abtastblöcke können in einem Ansatz Bitleitungstreiber einschließen. Ein SGS-Transistor ist ein Auswahlgate-Transistor an einem Source-Ende einer NAND-Kette, und ein SGD-Transistor ist ein Auswahlgate-Transistor an einem Drain-Ende einer NAND-Kette.

[0031] In einigen Ausführungsformen können einige der Komponenten kombiniert werden. Bei verschiedenen Ausführungen können eine oder mehrere der Komponenten (allein oder in Kombination), die sich von der Speicherstruktur 126 unterscheiden, als mindestens eine Steuerschaltung betrachtet werden, die dazu eingerichtet ist, die hier beschriebenen Vorgänge durchzuführen. Die Steuerschaltlogik kann zum Beispiel eines oder eine Kombination von Steuerschaltlogik 110, Zustandsmaschine 112, Decodierern 114/132, Leistungssteuerungsmodul 116, Erfassungsblöcken SBb, SB2, ., SBp, Lese-/Schreibschaltungen 128, Steuerung 122 und so weiter einschließen.

[0032] Die Steuerschaltungen können eine Programmierschaltung einschließen, die so eingerichtet ist, dass sie einen Programm- und Prüfvorgang für einen Satz von Speicherzellen durchführt, wobei der eine Satz von Speicherzellen Speicherzellen auf-

weist, die so zugeordnet sind, dass sie einen Datenzustand aus einer Vielzahl von Datenzuständen darstellen, und Speicherzellen, die so zugeordnet sind, dass sie einen anderen Datenzustand aus der Vielzahl von Datenzuständen darstellen; wobei der Programm- und Prüfvorgang eine Vielzahl von Programm- und Prüfitationen aufweist; und wobei die Programmierschaltung in jeder Programmier- und Prüfitation eine Programmierung für die eine ausgewählte Wortleitung durchführt, nach der die Programmierschaltung ein Prüfsignal an die ausgewählte Wortleitung anlegt. Die Steuerschaltungen können auch eine Zählung enthalten, die so eingerichtet ist, dass sie die Anzahl der Speicherzellen ermittelt, die einen Verifizierungstest für den einen Datenzustand bestanden haben. Die Steuerschaltungen können auch eine Bestimmungsschaltung einschließen, die eingerichtet ist, um basierend auf einer Menge, um die die Zählung einen Schwellenwert überschreitet, zu bestimmen, ob ein Programmiervorgang abgeschlossen ist.

[0033] Fig. 1B ist zum Beispiel ein Blockdiagramm einer beispielhaften Steuerschaltung 150, die eine Programmierschaltung 151, eine Zählung 152 und eine Bestimmungsschaltung 153 aufweist.

[0034] Die Off-Chip-Steuerung 122 kann einen Prozessor 122c, Speichervorrichtungen (Speicher), wie ROM 122a und RAM 122b, und eine Fehlerkorrekturcode-Engine (ECC-Engine) 245 aufweisen. Die ECC-Maschine kann eine Anzahl von Lesefehlern korrigieren, die verursacht werden, wenn der obere Schwanz einer Vth-Verteilung zu hoch wird. In einigen Fällen können jedoch unkorrigierbare Fehler vorliegen. Die hierin bereitgestellten Techniken verringern die Wahrscheinlichkeit von unkorrigierbaren Fehlern.

[0035] Die Speichervorrichtung(en) 122a, 122b weisen Codes auf, wie beispielsweise einen Satz von Anweisungen, und der Prozessor 122c ist betreibbar, um den Satz von Anweisungen auszuführen und die hierin beschriebene Funktionalität bereitzustellen. Alternativ oder zusätzlich kann der Prozessor 122c auf Code von einer Speichervorrichtung 126a der Speicherstruktur 126 zugreifen, wie einen reservierten Bereich von Speicherzellen in einer oder mehreren Wortleitungen. Codes können zum Beispiel von der Steuerung 122 verwendet werden, um auf die Speicherstruktur 126 zuzugreifen, wie für Programmier-, Lese- und Löschvorgänge. Der Code kann Bootcode und Steuercode (z. B. einen Satz von Anweisungen) einschließen. Der Bootcode ist eine Software, welche die Steuerung 122 während eines Boot- oder Startvorgangs initialisiert und der Steuerung 122 ermöglicht, auf die Speicherstruktur 126 zuzugreifen. Der Code kann von der Steuerung 122 verwendet werden, um eine oder mehrere Speicherstrukturen 126 zu steuern. Beim Hochfahren ruft

der Prozessor 122c den Bootcode aus dem ROM 122a oder der Speichervorrichtung 126a zur Ausführung ab, und der Bootcode initialisiert die Systemkomponenten und lädt den Steuercode in den RAM 122b. Sobald der Steuercode in den RAM 122b geladen ist, wird er durch den Prozessor 122c ausgeführt. Der Steuercode enthält Treiber zum Durchführen grundlegender Aufgaben, wie Steuern und Zuweisen von Speicher, Priorisieren der Verarbeitung von Befehlen und Steuern von Eingabe- und Ausgabeports.

[0036] Im Allgemeinen kann der Steuercode Anweisungen einschließen, um die hierin beschriebenen Funktionen einschließlich der weiter unten erläuterten Schritte der Flussdiagramme durchzuführen und um die Spannungswellenformen einschließlich der weiter unten erläuterten bereitzustellen.

[0037] In einer Ausführungsform ist der Host eine Rechenvorrichtung (z. B. ein Laptop, ein Desktop, ein Smartphone, ein Tablet, eine Digitalkamera), die einen oder mehrere Prozessoren, eine oder mehrere prozessorlesbare Speichervorrichtungen (RAM, ROM, Flash-Speicher, Festplattenlaufwerk, Solid-State-Memory) einschließt, die einen prozessorlesbaren Code (z. B. eine Software) zum Programmieren der einen oder der mehreren Prozessoren speichert, um die hierin beschriebenen Verfahren durchzuführen. Der Host kann auch einen zusätzlichen Systemspeicher, eine oder mehrere Eingabe-/Ausgabeschnittstellen und/oder eine oder mehrere Eingabe-/Ausgabevorrichtungen einschließen, die mit dem einen oder den mehreren Prozessoren in Kommunikation stehen.

[0038] Andere Arten von nichtflüchtigen Speichern zusätzlich zu NAND-Flash-Speichern können ebenfalls verwendet werden.

[0039] Halbleiterspeichervorrichtungen schließen flüchtige Speichervorrichtungen ein, wie etwa Dynamic Random Access Memory-Vorrichtungen („DRAM-Vorrichtungen“) oder Static Random Access Memory-Vorrichtungen („SRAM-Vorrichtungen“), nichtflüchtige Speichervorrichtungen, wie etwa Resistive Random Access Memory („ReRAM“), Electrically Erasable Programmable Read Only Memory („EEPROM“), Flash-Speicher (der auch als Untergruppe eines EEPROM betrachtet werden kann), Ferroelectric Random Access Memory („FRAM“) und Magnetoresistive Random Access Memory („MRAM“), und andere Halbleiterelemente, die Informationen speichern können. Jeder Typ von Speichervorrichtung kann unterschiedliche Konfigurationen aufweisen. Zum Beispiel können Flash-Speichervorrichtungen in einer NAND- oder NOR-Konfiguration eingerichtet sein.

[0040] Die Speichervorrichtungen können aus passiven und/oder aktiven Elementen in beliebigen Kombinationen gebildet sein. Als ein nichteinschränkendes Beispiel schließen passive Halbleiterspeicherelemente ReRAM-Vorrichtungselemente ein, die in einigen Ausführungsformen ein widerstandsschaltendes Speicherelement wie ein Anti-Fuse- oder Phasenwechselmaterial, gefolgt von gegebenenfalls einem Lenkelement wie eine Diode oder einem Transistor einschließen. Weiterhin schließen als nichteinschränkendes Beispiel aktive Halbleiterspeicherelemente EEPROM- und Flash-Speichervorrichtungselemente ein, die in einigen Ausführungsformen Elemente einschließen, die einen Ladungsspeicherbereich enthalten, wie ein Floating-Gate, leitende Nanopartikel oder ein dielektrisches Ladungsspeichermaterial.

[0041] Mehrere Speicherelemente können derart eingerichtet sein, dass sie in Reihe geschaltet sind, oder derart, dass jedes Element einzeln zugänglich ist. Als nicht einschränkendes Beispiel enthalten Flash-Speichervorrichtungen in einer NAND-Konfiguration (NAND-Speicher) üblicherweise Speicherelemente, die in Reihe geschaltet sind. Eine NAND-Kette ist ein Beispiel für einen Satz von in Reihe geschalteten Transistoren, aufweisend Speicherzellen und SG-Transistoren.

[0042] Ein NAND-Speicherarray kann so eingerichtet werden, dass das Array aus mehreren Speicherketten besteht, wobei eine Kette aus mehreren Speicherelementen besteht, die sich eine einzelne Bitleitung teilen und auf die als Gruppe zugegriffen wird. Alternativ können Speicherelemente derart eingerichtet sein, dass jedes Element einzeln zugänglich ist, z. B. ein NOR-Speicherarray. NAND- und NOR-Speicherkonfigurationen sind Beispiele, und Speicherelemente können anderweitig eingerichtet sein. Die Halbleiterspeicherelemente, die in und/oder über einem Substrat angeordnet sind, können in zwei oder drei Dimensionen angeordnet sein, wie einer zweidimensionalen Speicherstruktur oder einer dreidimensionalen Speicherstruktur.

[0043] In einer zweidimensionalen Speicherstruktur sind die Halbleiterspeicherelemente in einer einzelnen Ebene oder einer einzelnen Speichervorrichtungsebene angeordnet. Üblicherweise sind in einer zweidimensionalen Speicherstruktur Speicherelemente in einer Ebene (z.B. in einer Ebene in x-y-Richtung) angeordnet, die sich im Wesentlichen parallel zu einer Hauptfläche eines Substrats erstreckt, das die Speicherelemente trägt. Das Substrat kann ein Wafer sein, über oder in dem die Schicht der Speicherelemente gebildet ist, oder es kann ein Trägersubstrat sein, das an den Speicherelementen befestigt wird, nachdem diese gebildet werden. Als ein nicht einschränkendes Beispiel

kann das Substrat einen Halbleiter, wie Silizium, einschließen.

[0044] Die Speicherelemente können auf der einzelnen Speichervorrichtungsebene in einem geordneten Array angeordnet sein, wie in einer Vielzahl von Zeilen und/oder Spalten. Jedoch können die Speicherelemente in nicht regelmäßigen oder nicht orthogonalen Konfigurationen angeordnet sein. Die Speicherelemente können jeweils zwei oder mehr Elektroden oder Kontaktleitungen, wie Bitleitungen und Wortleitungen, aufweisen.

[0045] Ein dreidimensionales Speicherarray ist so angeordnet, dass Speicherelemente mehrere Ebenen oder mehrere Speichervorrichtungsebenen belegen, wodurch eine Struktur in drei Dimensionen gebildet wird (d. h. in der x-, y- und z-Richtung, wobei die z-Richtung im Wesentlichen senkrecht ist und die x- und y-Richtungen im Wesentlichen parallel zur Hauptoberfläche des Substrats verlaufen).

[0046] Als nicht einschränkendes Beispiel kann eine dreidimensionale Speicherstruktur vertikal als Stapel mehrerer zweidimensionaler Speichervorrichtungsebenen angeordnet sein. Als weiteres nicht einschränkendes Beispiel kann ein dreidimensionales Speicherarray als mehrere vertikale Spalten (z.B. Spalten, die sich im Wesentlichen senkrecht zur Hauptfläche des Substrats erstrecken, d. h. in y-Richtung) angeordnet sein, wobei jede Spalte mehrere Speicherelemente aufweist. Die Spalten können in einer zweidimensionalen Konfiguration angeordnet sein, z.B. in einer x-y-Ebene, was in einer dreidimensionalen Anordnung von Speicherelementen mit Elementen auf mehreren vertikal gestapelten Speicherebenen resultiert. Andere Konfigurationen von Speicherelementen in drei Dimensionen können auch ein dreidimensionales Speicherarray bilden.

[0047] Als nicht einschränkendes Beispiel können in einer dreidimensionalen Anordnung von NAND-Ketten die Speicherelemente miteinander verbunden werden, um eine NAND-Kette innerhalb einer einzigen horizontalen (z. B. x-y) Speichervorrichtungsebene zu bilden. Alternativ können die Speicherelemente miteinander gekoppelt sein, um eine vertikale NAND-Kette zu bilden, die über mehrere horizontale Speichervorrichtungsebenen verläuft. Andere dreidimensionale Konfigurationen können in Betracht gezogen werden, wobei einige NAND-Ketten Speicherelemente in einer einzelnen Speicherebene enthalten, während andere Ketten Speicherelemente enthalten, die sich über mehrere Speicherebenen erstrecken. Dreidimensionale Speicherarrays können auch in einer NOR-Konfiguration und in einer ReRAM-Konfiguration ausgebildet sein.

[0048] Üblicherweise werden in einem monolithischen dreidimensionalen Speicherarray eine oder

mehrere Speichervorrichtungsebenen über einem einzelnen Substrat gebildet. Optional kann das monolithische dreidimensionale Speicherarray auch eine oder mehrere Speicherschichten mindestens teilweise innerhalb des einzelnen Substrats aufweisen. Als ein nicht einschränkendes Beispiel kann das Substrat einen Halbleiter, wie Silizium, einschließen. In einem monolithischen dreidimensionalen Array werden die Schichten, die die einzelnen Speichervorrichtungsebenen des Arrays bilden, üblicherweise auf den Schichten der darunter liegenden Speichervorrichtungsebenen des Arrays gebildet. Jedoch können Schichten von benachbarten Speichervorrichtungsebenen eines monolithischen dreidimensionalen Speicherarrays gemeinsam genutzt werden oder Zwischenschichten zwischen Speichervorrichtungsebenen aufweisen.

[0049] Auch können wiederum zweidimensionale Arrays getrennt gebildet und dann zusammengepackt werden, um eine nicht monolithische Speichervorrichtung mit mehreren Speicherschichten zu bilden. Zum Beispiel können nicht monolithische gestapelte Speicher konstruiert werden, indem Speicherebenen auf separaten Substraten gebildet und dann die Speicherebenen übereinander gestapelt werden. Die Substrate können vor dem Stapeln gedünnt oder von den Speichervorrichtungsebenen entfernt werden, aber da die Speichervorrichtungsebenen anfänglich über separaten Substraten gebildet werden, sind die resultierenden Speicherarrays keine monolithischen dreidimensionalen Speicherarrays. Weiterhin können mehrere zweidimensionale Speicherarrays oder dreidimensionale Speicherarrays (monolithisch oder nicht monolithisch) auf separaten Chips gebildet und dann zusammengepackt werden, um eine gestapelte Chip-Speichervorrichtung zu bilden.

[0050] Fig. 2 veranschaulicht Blöcke 200, 210 von Speicherzellen in einer beispielhaften zweidimensionalen Konfiguration des Speicherarrays 126 von Fig. 1. Das Speicherarray 126 kann viele derartige Speicherblöcke 200, 210 einschließen. Jeder Beispielblock 200, 210 schließt eine Anzahl von NAND-Ketten und entsprechenden Bitleitungen ein, z. B. BL0, BL1, .. die von den Blöcken gemeinsam genutzt werden. Jede NAND-Kette ist an einem Ende mit einem Drain-Side-Select-Gate (SGD) verbunden, und die Steuer-Gates der Drain-Select-Gates sind über eine gemeinsame SGD-Leitung verbunden. Die NAND-Ketten sind an ihrem anderen Ende mit einem Source-Side-Select-Gate (SGS) verbunden, das wiederum mit einer gemeinsamen Source-Leitung 220 verbunden ist. Einhundertzwei Wortleitungen, zum Beispiel WL0-WL111, erstrecken sich zwischen den SGSs und den SGDs. In einigen Ausführungsformen kann der Speicherblock mehr oder weniger als einhundertzwei Wortleitungen einschließen. Zum Beispiel schließt ein Speicherblock in

einigen Ausführungsformen einhundertvierundsechzig Wortleitungen ein. In einigen Fällen können im Speicherarray neben den Auswahl-Gate-Transistoren auch Dummy-Wortleitungen verwendet werden, die keine Benutzerdaten enthalten. Solche Dummy-Wortleitungen können die Randdatenwortleitung vor bestimmten Randeffekten abschirmen.

[0051] Ein Typ von nichtflüchtigem Speicher, der im Speicherarray bereitgestellt werden kann, ist ein Floating-Gate-Speicher wie vom Typ, der in **Fig. 3A** und **3B** gezeigt ist. Andere Typen nichtflüchtiger Speicher können jedoch ebenfalls verwendet werden. Wie nachstehend ausführlicher erörtert wird, verwendet in einem anderen Beispiel, das in **Fig. 4A** und **4B** gezeigt ist, eine Charge-Trapping-Speicherzelle ein nichtleitendes dielektrisches Material anstelle eines leitenden Floating-Gates, um Ladung nichtflüchtig zu speichern. Ein dreischichtiges Dielektrikum aus Siliziumoxid, Siliziumnitrid und Siliziumoxid („ONO“) ist zwischen einem leitenden Steuer-Gate und einer Oberfläche eines halbleitenden Substrats über dem Speicherzellenkanal angeordnet. Die Zelle wird programmiert, indem Elektronen aus dem Zellkanal in das Nitrid eingespritzt werden, wo sie eingefangen und in einem begrenzten Bereich gespeichert werden. Diese gespeicherte Ladung verändert dann die Schwellenspannung eines Abschnitts des Kanals der Zelle auf eine Weise, die nachweisbar ist. Die Zelle wird durch Einspritzen heißer Löcher in das Nitrid gelöscht. Eine ähnliche Zelle kann in einer Split-Gate-Konfiguration bereitgestellt werden, bei der sich ein dotiertes Polysilizium-Gate über einem Abschnitt des Speicherzellenkanals erstreckt und dadurch einen separaten Auswahl-Transistor bildet.

[0052] Bei einem anderen Ansatz werden NROM-Zellen verwendet. Zwei Bit werden zum Beispiel in jeder NROM-Zelle gespeichert, in der sich eine dielektrische ONO-Schicht über dem Kanal zwischen Source- und Drain-Diffusion erstreckt. Die Ladung für ein Datenbit ist in der an den Drain angrenzenden dielektrischen Schicht lokalisiert, und die Ladung für das andere Datenbit ist in der an die Source angrenzenden dielektrischen Schicht lokalisiert. Die Datenspeicherung in mehreren Zuständen wird durch separates Lesen binärer Zustände der physisch getrennten Ladungsspeicherungsbereiche innerhalb des Dielektrikums erreicht. Andere Arten nichtflüchtiger Speicher sind ebenfalls bekannt.

[0053] **Fig. 3A** veranschaulicht einen Querschnitt von beispielhaften Floating-Gate-Speicherzellen 300, 310, 320 in NAND-Ketten. In dieser Figur verläuft die Richtung einer Bitleitung oder NAND-Kette in die Seite hinein, und die Richtung einer Wortleitung verläuft von links nach rechts. Als Beispiel erstreckt sich die Wortleitung 324 über NAND-Ketten, welche die jeweiligen Kanalregionen 306, 316

und 326 einschließen. Die Speicherzelle 300 schließt ein Steuer-Gate 302, ein Floating-Gate 304, eine Tunneloxidschicht 305 und die Kanalregion 306 ein. Die Speicherzelle 310 schließt ein Steuer-Gate 312, ein Floating-Gate 314, eine Tunneloxidschicht 315 und die Kanalregion 316 ein. Die Speicherzelle 320 schließt ein Steuer-Gate 322, ein Floating-Gate 321, eine Tunneloxidschicht 325 und die Kanalregion 326 ein. Jede Speicherzelle 300, 310, 320 befindet sich in einer anderen jeweiligen NAND-Kette. Eine interpolare dielektrische (IPD) Schicht 328 ist ebenfalls veranschaulicht. Die Steuer-Gates 302, 312, 322 sind Abschnitte der Wortleitung. Eine Querschnittsansicht entlang des Kontaktleitungsverbinders 329 ist in **Fig. 3B** bereitgestellt.

[0054] Das Steuer-Gate 302, 312, 322 umschließt das Floating-Gate 304, 314, 321, wodurch der Oberflächenkontaktbereich zwischen dem Steuer-Gate 302, 312, 322 und dem Floating-Gate 304, 314, 321 vergrößert wird. Daraus ergibt sich eine höhere IPD-Kapazität, was zu einem höheren Kopplungsverhältnis führt, was das Programmieren und Löschen erleichtert. Bei der Verkleinerung von NAND-Speichervorrichtungen wird jedoch der Abstand zwischen benachbarten Zellen 300, 310, 320 kleiner, sodass zwischen zwei benachbarten Floating-Gates 302, 312, 322 fast kein Platz mehr für die Steuer-Gates 302, 312, 322 und die IPD-Schicht 328 vorhanden ist.

[0055] Als Alternative wurde, wie in **Fig. 4A** und **4B** gezeigt, die flache oder planare Speicherzelle 400, 410, 420 entwickelt, bei der das Steuer-Gate 402, 412, 422 flach oder planar ist; d. h. sie umschließt nicht das Floating-Gate und hat nur von oben Kontakt mit der Ladungsspeicherschicht 428. In diesem Fall ist ein hohes Floating-Gate nicht von Vorteil. Stattdessen wird das Floating-Gate deutlich dünner ausgeführt. Weiterhin kann das Floating-Gate zur Ladungsspeicherung verwendet werden, oder es kann eine dünne Ladungsfallschicht zum Einfangen von Ladung verwendet werden. Dieser Ansatz kann das Problem des ballistischen Elektronentransports vermeiden, bei dem ein Elektron während der Programmierung durch das Floating-Gate wandern kann, nachdem es durch das Tunneloxid getunnelt wurde.

[0056] **Fig. 4A** stellt einen Querschnitt von beispielhaften Charge-Trapping-Speicherzellen 400, 410, 420 in NAND-Ketten dar. Die Ansicht ist in einer Wortleitungsrichtung von Speicherzellen 400, 410, 420, die ein flaches Steuer-Gate und Charge-Trapping-Regionen als ein zweidimensionales Beispiel für Speicherzellen 400, 410, 420 in dem Speicherarray 126 aus **Fig. 1** aufweisen. Der Ladungseinfangspeicher kann in einer NOR- und NAND-Flash-Speichervorrichtung verwendet werden. Bei dieser Technologie wird ein Isolator wie eine SiN-Folie

zum Speichern von Elektronen verwendet, im Gegensatz zu einer Floating-Gate-MOSFET-Technologie, bei der ein Leiter wie dotiertes polykristallines Silizium zum Speichern von Elektronen verwendet wird. Zum Beispiel erstreckt sich eine Wortleitung 424 über NAND-Ketten, die jeweilige Kanalregionen 406, 416, 426 einschließen. Abschnitte der Wortleitung stellen Steuer-Gates 402, 412, 422 bereit. Unterhalb der Wortleitung befinden sich eine IPD-Schicht 428, die Charge-Trapping-Schichten 404, 414, 421, die Polysiliziumschichten 405, 415, 425 und die Tunnelschichten 409, 407, 408. Jede Charge-Trapping-Schicht 404, 414, 421 erstreckt sich kontinuierlich in einer jeweiligen NAND-Kette. Die flache Konfiguration des Steuer-Gates kann dünner gemacht werden als ein Floating-Gate. Außerdem können die Speicherzellen näher beieinander platziert werden.

[0057] Fig. 4B veranschaulicht einen Querschnitt der Struktur von **Fig. 4A** entlang des Kontaktleitungsverbinders 429. Die NAND-Kette 430 schließt einen SGS-Transistor 431, beispielhafte Speicherzellen 400, 433, ... 435 und einen SGD-Transistor 436 ein. Durchgänge in der IPD-Schicht 428 in den SGS- und SGD-Transistoren 431, 436 ermöglichen die Kommunikation zwischen den Steuer-Gate-Schichten 402 und den Floating-Gate-Schichten. Die Steuer-Gate-402-Schichten und Floating-Gate-Schichten können aus Polysilizium bestehen, und die Tunneloxidschicht kann beispielsweise aus Siliziumoxid bestehen. Die IPD-Schicht 428 kann ein Stapel aus Nitriden (N) und Oxiden (O) sein, wie beispielsweise in einer N-O-N-O-N-Konfiguration.

[0058] Die NAND-Kette kann auf einem Substrat gebildet werden, das eine p-Typ-Substratregion 455, einen n-Typ-Well 456 und einen p-Typ-Well 457 aufweist. N-Typ Source-/Drain-Diffusionsregionen sd1, sd2, sd3, sd4, sd5, sd6 und sd7 werden in dem p-Typ-Well gebildet. Eine Kanalspannung, V_{ch} , kann direkt an die Kanalregion des Substrats angelegt werden.

[0059] Fig. 5 veranschaulicht ein beispielhaftes Blockdiagramm des Erfassungsblocks SB1 von **Fig. 1**. In einem Ansatz weist ein Erfassungsblock mehrere Erfassungsschaltungen auf. Jeder Erfassungsschaltung sind Datenhaltespeicher zugeordnet. Die Erfassungsschaltungen 550a, 551a, 552a und 553a sind zum Beispiel den Datenhaltespeicher 550b, 551b, 552b bzw. 553b zugeordnet. Bei einem Ansatz können unterschiedliche Untergruppen von Bitleitungen unter Verwendung unterschiedlicher jeweiliger Erfassungsböcke abgetastet werden. Dies ermöglicht, dass die Prozessorlast, die zu den Erfassungsschaltungen gehört, durch einen jeweiligen Prozessor in jedem Erfassungsblock aufgeteilt und gehandhabt wird. Die Erfassungsschaltungssteuerung 560 in SB1 kann zum Beispiel mit dem

Satz der Erfassungsschaltungen und Haltespeicher kommunizieren. Die Erfassungsschaltungssteuerung 560 kann eine Vorladeschaltung 561 einschließen, die jeder Erfassungsschaltung eine Spannung zur Einstellung einer Vorladespannung bereitstellt. Bei einem möglichen Ansatz wird die Spannung jeder Erfassungsschaltung unabhängig bereitgestellt, z. B. über den Datenbus und einen lokalen Bus. Bei einem anderen möglichen Ansatz wird jeder Erfassungsschaltung gleichzeitig eine gemeinsame Spannung bereitgestellt. Die Erfassungsschaltungssteuerung 560 kann auch eine Vorladeschaltung 561, einen Speicher 562 und einen Prozessor 563 einschließen. Der Speicher 562 kann Codes speichern, die vom Prozessor ausgeführt werden können, um die hierin beschriebenen Funktionen auszuführen. Diese Funktionen können das Lesen der Haltespeicher 550b, 551b, 552b, 553b, die den Erfassungsschaltungen 550a, 551a, 552a, 553a zugeordnet sind, das Setzen von Bitwerten in den Haltespeicher und das Bereitstellen von Spannungen zum Einstellen von Vorladepegeln in den Erfassungsknoten der Erfassungsschaltungen 550a, 551a, 552a, 553a einschließen. Weitere Beispieldetails zur Steuerung der Erfassungsschaltung 560 und zu den Erfassungsschaltungen 550a, 551a, 552a, 553a werden nachfolgend bereitgestellt.

[0060] In einigen Ausführungsformen kann eine Speicherzelle ein Flag-Register einschließen, das einen Satz von Haltespeichern zum Speichern von Flag-Bits einschließt. In einigen Ausführungsformen kann eine Menge von Flag-Registern einer Menge von Datenzuständen entsprechen. In einigen Ausführungsformen können ein oder mehrere Flag-Register zur Steuerung eines Typs der Verifizierungstechnik bei der Verifizierung von Speicherzellen verwendet werden. In einigen Ausführungsformen kann die Ausgabe eines Flag-Bits die zugehörige Logik der Vorrichtung, z. B. die Adressdecodierungsschaltlogik, so modifizieren, dass ein bestimmter Block von Zellen ausgewählt wird. Ein Massenbetrieb (z. B. ein Löschvorgang usw.) kann unter Verwendung der im Flag-Register gesetzten Flags oder einer Kombination des Flag-Registers mit dem Adressregister, wie bei der impliziten Adressierung, oder alternativ durch direkte Adressierung mit dem Adressregister allein durchgeführt werden.

[0061] Fig. 6A ist eine perspektivische Ansicht eines Satzes von Blöcken 600 in einer beispielhaften dreidimensionalen Konfiguration des Speicherarrays 126 von **Fig. 1**. Auf dem Substrat befinden sich beispielhafte Blöcke BLK0, BLK1, BLK2, BLK3 von Speicherzellen (Speicherelementen) und ein Peripheriebereich 604 mit Schaltlogik zur Verwendung durch die Blöcke BLK0, BLK1, BLK2, BLK3. Zum Beispiel können die Schaltungen die Spannungstreiber 605 einschließen, die mit Steuer-Gate-Schichten der Blöcke BLK0, BLK1, BLK2, BLK3 verbunden

werden können. Bei einem Ansatz werden Steuer-Gate-Schichten auf einer gemeinsamen Höhe in den Blöcken BLK0, BLK1, BLK2, BLK3 gemeinsam angesteuert. Das Substrat 601 kann auch Schaltlogik unter den Blöcken BLK0, BLK1, BLK2, BLK3 zusammen mit einer oder mehreren unteren Metallschichten tragen, welche in Leiterbahnen strukturiert sind, um Signale der Schaltlogik zu übertragen. Die Blöcke BLK0, BLK1, BLK2, BLK3 sind in einer Zwischenregion 602 der Speichervorrichtung ausgebildet. In einer oberen Region 603 der Speichervorrichtung sind eine oder mehrere obere Metallschichten in Leiterbahnen strukturiert, um Signale der Schaltlogik zu führen. Jeder Block BLK0, BLK1, BLK2, BLK3 weist einen gestapelten Speicherzellenbereich auf, wobei abwechselnde Ebenen des Stapels Wortleitungen darstellen. In einem möglichen Ansatz hat jeder Block BLK0, BLK1, BLK2, BLK3 gegenüberliegende abgestufte Seiten, von denen sich vertikale Kontakte nach oben zu einer oberen Metallschicht erstrecken, um Verbindungen zu Leiterbahnen herzustellen. Während vier Blöcke BLK0, BLK1, BLK2, BLK3 beispielhaft veranschaulicht sind, können zwei oder mehr Blöcke verwendet werden, die sich in x- und/oder y-Richtung erstrecken.

[0062] Bei einem möglichen Ansatz stellt die Länge der Ebene in x-Richtung eine Richtung dar, in der sich Signalpfade zu Wortleitungen in der einen oder den mehreren oberen Metallschichten erstrecken (eine Wortleitungs- oder SGD-Leitungsrichtung), und die Breite der Ebene in y-Richtung stellt eine Richtung dar, in der sich Signalpfade zu Bitleitungen in der einen oder den mehreren oberen Metallschichten (eine Bitleitungsrichtung) erstrecken. Die z-Richtung stellt eine Höhe der Speichervorrichtung dar.

[0063] Fig. 6B veranschaulicht einen beispielhaften Querschnitt eines Abschnitts von einem der Blöcke BLK0, BLK1, BLK2, BLK3 aus Fig. 6A. Der Block weist einen Stapel 610 aus abwechselnd leitenden und dielektrischen Schichten auf. In diesem Beispiel weisen die leitfähigen Schichten zwei SGD-Schichten, zwei SGS-Schichten und vier Dummy-Wortleitungsschichten DWLD0, DWLD1, DWLS0 und DWLS1, zusätzlich zu den Datenwortleitungsschichten (Wortleitungen) WL0-WL111 auf. Die dielektrischen Schichten werden als DL0-DL116 bezeichnet. Weiterhin sind Regionen des Stapels 610 veranschaulicht, welche die NAND-Ketten NS1 und NS2 aufweisen. Jede NAND-Kette umfasst ein Speicherloch 618, 619, das mit Materialien gefüllt ist, die benachbart zu den Wortzeilen Speicherzellen bilden. Eine Region 622 des Stapels 610 ist in Fig. 6D ausführlicher gezeigt und wird nachstehend detaillierter erörtert.

[0064] Der 610-Stapel schließt ein Substrat 611, eine Isolierschicht 612 auf dem Substrat 611 und einen Abschnitt einer Sourceleitung SL ein. NS1 hat

ein Source-Ende 613 an einer Unterseite 614 des Stapels und ein Drain-Ende 615 an einer Oberseite 616 des Stapels 610. Kontaktleitungsverbinder (z. B. Schlitze, wie metallgefüllte Schlitze) 617, 620 können periodisch über den Stapel 610 als Verbindungen bereitgestellt werden, die sich durch den Stapel 610 erstrecken, wie zur Verbindung der Source-Leitung mit einer bestimmten Kontaktleitung oberhalb des Stapels 610. Die Kontaktleitungsverbinder 617, 620 können bei der Bildung der Wortleitungen verwendet und anschließend mit Metall gefüllt werden. Ein Abschnitt einer Bitleitung BL0 ist ebenfalls veranschaulicht. Eine leitfähige Durchkontaktierung 621 verbindet das Drain-Ende 615 mit BL0.

[0065] Fig. 6C veranschaulicht eine grafische Darstellung des Speicherlochdurchmessers im Stapel von Fig. 6B. Die vertikale Achse ist auf den Stapel von Fig. 6B ausgerichtet und veranschaulicht eine Breite (wMH), z. B. den Durchmesser der Speicherlöcher 618 und 619. Die Wortleitungsschichten WL0-WL111 von Fig. 6A werden als Beispiel wiederholt und befinden sich auf den jeweiligen Höhen z0-z111 im Stapel. In einer solchen Speichervorrichtung weisen die Speicherlöcher, die durch den Stapel geätzt werden, ein sehr hohes Seitenverhältnis auf. Zum Beispiel ist ein Verhältnis der Tiefe zum Durchmesser von etwa 25 bis 30 üblich. Die Speicherlöcher können einen kreisförmigen Querschnitt aufweisen. Aufgrund des Ätzprozesses kann die Speicherlochbreite über die Länge des Lochs variieren. Üblicherweise wird der Durchmesser von oben nach unten in dem Speicherloch immer kleiner. Das heißt, die Speicherlöcher verjüngen sich, wobei sie sich am Boden des Stapels verengen. In einigen Fällen kommt es zu einer leichten Verengung am oberen Ende des Lochs in der Nähe des Auswahlgates, sodass der Durchmesser etwas breiter wird, bevor er vom oberen Ende zum unteren Ende des Speicherlochs hin immer kleiner wird.

[0066] Aufgrund der ungleichmäßigen Breite des Speicherlochs kann die Programmiergeschwindigkeit, einschließlich der Programmflanke und der Löscheschwindigkeit der Speicherzellen in Abhängigkeit von ihrer Position entlang des Speicherlochs variieren, z. B. in Abhängigkeit von ihrer Höhe im Stapel. Bei einem kleineren Durchmesser des Speicherlochs ist das elektrische Feld über dem Tunneloxid relativ stärker, sodass die Programmier- und Löscheschwindigkeit relativ höher ist. Ein Ansatz besteht darin, Gruppen von benachbarten Wortleitungen zu definieren, für die der Speicherlochdurchmesser ähnlich ist, z. B. innerhalb eines definierten Durchmesserbereichs, und ein optimiertes Verifizierungsschema für jede Wortleitung in einer Gruppe anzuwenden. Verschiedene Gruppen können unterschiedlich optimierte Verifizierungsschemata aufweisen.

[0067] Fig. 6D veranschaulicht eine Nahaufnahme der Region 622 des Stapels 610 von Fig. 6B. Speicherzellen sind an den verschiedenen Ebenen des Stapels an dem Schnittpunkt einer Wortleitungsschicht und eines Speicherlochs ausgebildet. In diesem Beispiel sind die SGD-Transistoren 680, 681 über den Dummy-Speicherzellen 682, 683 und eine Datenspeicherzelle MC bereitgestellt. Eine Reihe von Schichten kann entlang der Seitenwand (SW) des Speicherlochs 630 und/oder innerhalb jeder Wortleitungsschicht, z. B. unter Verwendung von Atomschichtabscheidung, abgeschieden werden. Beispielsweise kann jede Säule (z. B. die Säule, die durch die Materialien innerhalb eines Speicherlochs 630 gebildet wird) eine Charge-Trapping-Schicht oder einen Charge-Trapping-Film 663 wie SiN oder ein anderes Nitrid, eine Tunnelschicht 664, einen Polysiliziumkörper oder Kanal 665 und einen dielektrischen Kern 666 einschließen. Eine Wortleitungsschicht kann ein sperrendes Oxid-/Block-High-k-Material 660, eine Metallbarriere 661 und ein leitendes Metall 662 wie Wolfram als Steuer-Gate einschließen. Es werden zum Beispiel die Steuer-Gates 690, 691, 692, 693 und 694 bereitgestellt. In diesem Beispiel werden alle Schichten mit Ausnahme des Metalls in dem Speicherloch 630 bereitgestellt. In anderen Ansätzen können einige der Schichten in der Steuer-Gate-Schicht sein. Zusätzliche Säulen sind gleichermaßen in den verschiedenen Speicherlöchern ausgebildet. Eine Säule kann eine säulenförmige aktive Fläche (AA) einer NAND-Kette bilden.

[0068] Wenn eine Speicherzelle programmiert wird, werden Elektronen in einem Teil der Ladungseinfangschicht gespeichert, welche der Speicherzelle zugeordnet ist. Diese Elektronen werden in die Ladungseinfangschicht aus dem Kanal und durch die Tunnelschicht gezogen. Die V_{th} einer Speicherzelle wird proportional zu der Menge der gespeicherten Ladung erhöht. Während eines Löschvorgangs kehren die Elektronen in den Kanal zurück.

[0069] Jedes der Speicherlöcher 630 kann mit einer Vielzahl von ringförmigen Schichten gefüllt sein, die eine Blockieroxidschicht, eine Ladungseinfangschicht 663, eine Tunnelschicht 664 und eine Kanalschicht aufweisen. Eine Kernregion jedes der Speicherlöcher 630 ist mit einem Körpermaterial gefüllt, und die Vielzahl von ringförmigen Schichten befindet sich zwischen der Kernregion und der Wortleitung in jedem der Speicherlöcher 630.

[0070] Die NAND-Kette kann als einen Floating-Body-Kanal aufweisend betrachtet werden, da die Länge des Kanals nicht auf einem Substrat ausgebildet ist. Weiterhin wird die NAND-Kette durch eine Vielzahl von Wortleitungsschichten bereitgestellt, die übereinander in einem Stapel angeordnet und

durch dielektrische Schichten voneinander getrennt sind.

[0071] Fig. 7A veranschaulicht eine Draufsicht auf eine beispielhafte Wortleitungsschicht WL0 des Stapels 610 von Fig. 6B. Wie vorstehend erwähnt, kann eine dreidimensionale Speichervorrichtung einen Stapel aus abwechselnd leitenden und dielektrischen Schichten aufweisen. Die leitenden Schichten stellen die Steuer-Gates der SG-Transistoren und Speicherzellen bereit. Die für die SG-Transistoren verwendeten Schichten sind SG-Schichten, und die für die Speicherzellen verwendeten Schichten sind Wortleitungsschichten. Weiterhin sind im Stapel Speicherlöcher gebildet und mit einem Ladungseinfangmaterial und einem Kanalmaterial gefüllt. Dadurch wird eine vertikale NAND-Kette gebildet. Sourceleitungen sind mit den NAND-Ketten unter dem Stapel verbunden, und Bitleitungen sind mit den NAND-Ketten über dem Stapel verbunden.

[0072] Ein Block BLK in einer dreidimensionalen Speichervorrichtung kann in Unterblöcke unterteilt werden, wobei jeder Unterblock eine NAND-Kette aufweist, die eine gemeinsame SGD-Steuerleitung aufweist. Siehe zum Beispiel die SGD-Leitungen/Steuerleitungen SGD0, SGD1, SGD2 und SGD3 in den Unterblöcken SBa, SBb, SBc bzw. SBd. Weiterhin kann eine Wortleitungsschicht in einem Block in Regionen unterteilt sein. Jede Region befindet sich in einem jeweiligen Unterblock und kann sich zwischen Kontaktleitungsverbindern (z. B. Schlitzen) erstrecken, die periodisch im Stapel gebildet werden, um die Wortleitungsschichten während des Herstellungsprozesses der Speichervorrichtung zu verarbeiten. Diese Bearbeitung kann das Ersetzen eines Opfermaterials der Wortleitungsschichten durch Metall einschließen. Im Allgemeinen sollte der Abstand zwischen den Kontaktleitungsverbindern relativ klein sein, um eine Begrenzung der Entfernung zu berücksichtigen, die ein Ätzmittel seitlich zurücklegen kann, um das Opfermaterial zu entfernen, und die das Metall zurücklegen kann, um einen Hohlraum zu füllen, der durch die Entfernung des Opfermaterials entstanden ist. Der Abstand zwischen den Kontaktleitungsverbindern kann zum Beispiel einige Zeilen von Speicherlöchern zwischen benachbarten Kontaktleitungsverbindern zulassen. Das Layout der Speicherlöcher und Kontaktleitungsverbinders sollte auch eine Begrenzung der Anzahl der Bitleitungen berücksichtigen, die sich über die Region erstrecken können, während jede Bitleitung mit einer anderen Speicherzelle verbunden ist. Nach der Bearbeitung der Wortleitungsschichten können die Kontaktleitungsverbinders optional mit Metall gefeilt werden, um eine Verbindung durch den Stapel bereitzustellen.

[0073] In diesem Beispiel befinden sich vier Zeilen von Speicherlöchern zwischen benachbarten Kon-

taktleitungsverbindern. Eine Zeile ist hier eine Gruppe von Speicherlöchern, die in x-Richtung ausgerichtet sind. Außerdem befinden sich die Zeilen von Speicherlöchern in einem gestaffelten Muster, um die Dichte der Speicherlöcher zu erhöhen. Die Wortleitungsschicht oder Wortleitung ist in Regionen WL0a, WL0b, WL0c und WL0d unterteilt, die jeweils durch eine Kontaktleitung 713 verbunden sind. Bei einem Ansatz kann die letzte Region einer Wortleitungsschicht in einem Block mit einer ersten Region einer Wortleitungsschicht in einem nächsten Block verbunden sein. Die Kontaktleitung 713 wiederum ist mit einem Spannungstreiber für die Wortleitungsschicht verbunden. Die Region WL0a weist entlang einer Kontaktleitung 712 beispielhafte Speicherlöcher 710, 711 auf. Die Region WL0b hat beispielhafte Speicherlöcher 714, 715. Die Region WL0c weist beispielhafte Speicherlöcher 716, 717 auf. Die Region WL0d hat beispielhafte Speicherlöcher 718, 719. Die Speicherlöcher sind auch in **Fig. 7B** gezeigt. Jedes Speicherloch kann Teil einer entsprechenden NAND-Kette sein. Zum Beispiel können die Speicherlöcher 710, 714, 716 und 718 Teil der NAND-Ketten NS0_SBa, NS1_SBb, NS2_SBc, NS3_SBd und NS4_SBe sein.

[0074] Jeder Kreis stellt den Querschnitt eines Speicherlochs an einer Wortleitungsschicht oder SG-Schicht dar. Jeder Kreis, der mit gestrichelten Linien gezeigt ist, stellt Speicherzellen dar, die durch die Materialien in dem Speicherloch und durch die benachbarte Wortleitungsschicht bereitgestellt werden. Zum Beispiel sind die Speicherzellen 720, 721 in WL0a, die Speicherzellen 724, 725 in WL0b, die Speicherzellen 726, 727 in WL0c und die Speicherzellen 728, 729 in WL0d. Diese Speicherzellen befinden sich auf einer gemeinsamen Höhe in dem Stapel.

[0075] Kontaktleitungsverbinder (z. B. Schlitze, wie metallgefüllte Schlitze) 701, 702, 703, 704 können sich zwischen und angrenzend an die Kanten der Regionen WL0a-WL0d befinden. Die Kontaktleitungsverbinder 701, 702, 703, 704 stellen eine Leiterbahn von der Unterseite des Stapels zur Oberseite des Stapels bereit. Beispielsweise kann eine Sourceleitung an der Unterseite des Stapels mit einer leitfähigen Leitung über dem Stapel verbunden sein, wobei die leitfähige Leitung mit einem Spannungstreiber in einer peripheren Region der Speichervorrichtung verbunden ist.

[0076] **Fig. 7B** veranschaulicht eine Draufsicht auf eine beispielhafte obere dielektrische Schicht DL116 des Stapels von **Fig. 6B**. Die dielektrische Schicht ist in die Regionen DL116a, DL116b, DL116c und DL116d unterteilt. Jede Region kann mit einem jeweiligen Spannungstreiber verbunden sein. Dies ermöglicht, dass ein Satz von Speicherzellen in einer Region einer Wortleitungsschicht gleich-

zeitig programmiert wird, wobei jede Speicherzelle in einer jeweiligen NAND-Kette ist, die mit einer jeweiligen Bitleitung verbunden ist. Eine Spannung kann auf jeder Bitleitung eingestellt werden, um die Programmierung während jeder Programmierspannung zu ermöglichen oder zu unterbinden.

[0077] Die Region DL116a weist die beispielhaften Speicherlöcher 710, 711 entlang einer Kontaktleitung 712 auf, die mit einer Bitleitung BL0 zusammenfällt. Eine Anzahl von Bitleitungen erstreckt sich über die Speicherlöcher und ist mit den Speicherlöchern verbunden, wie durch die „X“-Symbole angedeutet. BL0 ist mit einem Satz von Speicherlöchern verbunden, der die Speicherlöcher 711, 715, 717, 719 einschließt. Eine andere beispielhafte Bitleitung BL1 ist mit einem Satz von Speicherlöchern verbunden, der die Speicherlöcher 710, 714, 716, 718 einschließt. Die Kontaktleitungsverbinder (z. B. Schlitze, wie metallgefüllte Schlitze) 701, 702, 703, 704 aus **Fig. 7A** sind ebenfalls veranschaulicht, da sie sich vertikal durch den Stapel erstrecken. Die Bitleitungen können in einer Sequenz BL0-BL23 über die DL116-Schicht in der x-Richtung nummeriert sein.

[0078] Unterschiedliche Untersätze von Bitleitungen sind mit Speicherzellen in verschiedenen Zeilen verbunden. Zum Beispiel sind BL0, BL4, BL8, BL12, BL16, BL20 mit Speicherzellen in einer ersten Zeile von Zellen am rechten Rand jeder Region verbunden. BL2, BL6, BL10, BL14, BL18, BL22 sind mit Speicherzellen in einer benachbarten Zeile von Zellen verbunden, die an die erste Zeile am rechten Rand angrenzt. BL3, BL7, BL11, BL15, BL19, BL23 sind mit Speicherzellen in einer ersten Zeile von Zellen am linken Rand jeder Region verbunden. BL1, BL5, BL9, BL13, BL17, BL21 sind mit Speicherzellen in einer benachbarten Zeile von Speicherzellen verbunden, die an die erste Zeile am linken Rand angrenzt.

[0079] Die Speicherzellen der Speicherblöcke können programmiert werden, um ein oder mehrere Datenbits in mehreren Datenzuständen zu speichern. Jeder Datenzustand ist einer jeweiligen Schwellenspannung V_t zugeordnet. Zum Beispiel stellt **Fig. 8** eine Schwellenspannungs- V_t -Verteilung einer Gruppe von Speicherzellen dar, die gemäß einem Speicherungsschema mit einem Bit pro Speicherzelle (SLC) programmiert sind. Im SLC-Speicherungsschema gibt es zwei Gesamtdatenzustände, einschließlich des gelöschten Zustands (Er) und eines einzelnen programmierten Datenzustands (S1). **Fig. 9** veranschaulicht die Schwellenspannungs- V_t -Verteilung eines Speicherungsschemas mit zwei Bits pro Zelle (MLC), das vier Gesamtdatenzustände, nämlich den gelöschten Zustand (Er) und drei programmierte Datenzustände (S1, S2 und S3), einschließt. Jeder programmierte Datenzustand (S1-S3) ist einer jeweiligen Prüfspannung (V_{v1} - V_{v3})

zugeordnet, die während eines Prüfabschnitts eines Programmiervorgangs verwendet wird. **Fig. 10** stellt eine Schwellenspannungs- V_t -Verteilung eines Speicherungschemas mit vier Bits pro Zelle (QLC) dar, die sechzehn Gesamtdatenzustände, nämlich den gelöschten Zustand (Er) und fünfzehn programmierte Datenzustände (S1-S15), einschließt. Andere Speicherungschemata sind ebenfalls verfügbar, wie drei Bits pro Zelle (TLC) mit acht Datenzuständen oder fünf Bits pro Zelle (SPS) mit zweiunddreißig Datenzuständen.

[0080] Ein Programmiervorgang beginnt im Allgemeinen mit den Speicherzellen einer Seite (wie einer Wortleitung) im gelöschten Datenzustand Er. Die Schwellenspannungen der Speicherzellen, die in einen der programmierten Datenzustände programmiert werden sollen, werden dann auf die gewünschten Pegel in einer oder mehreren Programmier-/Verifizieriterationen oder Programmierschleifen erhöht, von denen jede die Anwendung eines Programmierimpulses V_{pgm} auf ein Steuer-Gate einer ausgewählten Wortleitung und optional einem Verifizierimpuls in einem Verifiziervorgang einschließt. Je nachdem, ob eine Sperrspannung an eine Bitleitung, die mit einer Speicherzelle gekoppelt ist, angelegt wird oder nicht, können Elektronen im Ladungseinfangmaterial der Speicherzelle erfasst oder nicht erfasst werden, um die Speicherzelle entweder zu programmieren oder nicht zu programmieren. Insbesondere wenn eine Sperrspannung an eine Bitleitung angelegt wird, werden Elektronen am Eintreten in das Ladungseinfangmaterial der Speicherzelle, die betriebsfähig mit dieser Bitleitung verbunden ist, gehindert, und somit nimmt die Schwellenspannung V_t der Speicherzelle nicht zu. Umgekehrt, wenn keine Sperrspannung an eine Bitleitung angelegt wird, werden Elektronen nicht am Eintreten in das Ladungseinfangmaterial der Speicherzelle, die betriebsfähig mit dieser Bitleitung verbunden ist, gehindert, und eine Schwellenspannung V_t dieser Speicherzelle nimmt zu.

[0081] In vielen Programmiervorgängen wird die Spannung des Programmierimpulses V_{pgm} um eine feste Schrittgröße dV_{pgm} zwischen Programmierschleifen erhöht. Während des optionalen Prüfvorgangs wird die Schwellenspannung V_t einer Speicherzelle, die programmiert wird, im Prüfvorgang überprüft, um zu sehen, ob sie sich auf dem gewünschten Pegel befindet. Wenn sich die Speicherzelle nicht auf dem gewünschten Pegel befindet, dann wird die weitere Programmierung dieser Speicherzelle fortgesetzt. Umgekehrt, wenn die Schwellenspannung V_t der Speicherzelle größer als ein gewünschter Pegel ist, der dem Datenzustand zugeordnet ist, auf den die Speicherzelle programmiert wird, dann wird in nachfolgenden Programmierschleifen eine weitere Programmierung dieser Speicherzelle gesperrt.

[0082] Wenn die Speicherzellen auf eine niedrige Anzahl von Bits pro Speicherzelle (z. B. SLC oder MLC) programmiert werden, ist der Prüfvorgang in einigen Programmier Techniken nicht notwendig und kann übersprungen werden, um die Leistung zu verbessern (die Programmierzeit zu reduzieren). Zum Beispiel werden in einer SLC-Programmier Technik, die als 1P0V bekannt ist, die Speicherzellen einer ausgewählten Wortleitung mit nur einem Programmierimpuls und null Prüfimpulsen programmiert. In einer MLC-Programmier Technik, die als 3P0V bekannt ist, schließt der Programmiervorgang drei Programmierimpulse (eine für jeden der programmierten Datenzustände in MLC) und null Verifizierimpulse ein.

[0083] In einigen Speicherchips können die Speicherblöcke eingerichtet sein, um in mehreren Speicherungschema-Modi, z. B. einem SLC-Modus und einem TLC-Modus oder einem SLC-Modus und einem QLC-Modus, zu arbeiten. Eine solche Speichervorrichtung kann eingerichtet sein, um zunächst Daten in Speicherblöcke zu schreiben, die in einem SLC-Modus arbeiten, was eine hohe Leistung bietet. Dann kann die Speichervorrichtung in einem Hintergrundvorgang, wenn die Leistung nicht wichtig ist, die Daten aus dem SLC-Format in ein Format mit mehreren Bit pro Speicherzelle (zum Beispiel QLC) für eine Langzeitspeicherung umprogrammieren.

[0084] Wenn eine Schreibanforderung mehr Daten enthält als die Speichervorrichtung Kapazität im SLC-Format aufweist, muss die Speichervorrichtung während des (Nicht Hintergrund-) Schreibvorgangs in den QLC-Modus übergehen, sobald die SLC-Kapazität erreicht wurde. Dies kann die Leistung erheblich reduzieren. Zum Beispiel stellt **Fig. 11** ein Diagramm von Leistung (wie in geschriebenen Gigabytes pro Sekunde [GB/s] gemessen) im Vergleich zu geschriebenen Daten (wie in geschriebenen Terabytes [TBW] gemessen) für eine Speichervorrichtung während eines beispielhaften Programmiervorgangs dar, bei dem die zu schreibenden Daten die SLC-Grenze der Speichervorrichtung überschreiten. Wie veranschaulicht, arbeitet die Speichervorrichtung bei einer sehr hohen Leistung, während Daten im SLC-Modus geschrieben werden, bis sie zu Beginn des Programmiervorgangs fünfundzwanzig Prozent (25 %) ihrer verfügbaren Datenkapazität (die SLC-Grenze) erreicht. Diese Grenze wird erreicht, wenn alle Speicherblöcke in der Speichervorrichtung, die zu Beginn des Programmiervorgangs verfügbar waren, in das SLC-Format programmiert wurden. Nach diesem Punkt kann die SLC-Programmierung nicht fortgesetzt werden, da keine verbleibenden offenen Blöcke vorhanden sind, und daher werden die bereits in das SLC-Format programmierten Daten zusammen mit zusätzlichen Daten in das QLC-Format umprogrammieren, bis entweder der Programmiervorgang abgeschlossen ist oder die Daten-

kapazität der Speichervorrichtung erreicht ist. In einer neuen oder vollständig gelöschten Speichervorrichtung, in der alle Speicherblöcke offene Blöcke sind, ist die Datenkapazität die maximale Datenkapazität der Speichervorrichtung, z. B. 256 GB, 512 GB, 1 TB usw. Wenn einige Daten bereits in die Speichervorrichtung programmiert wurden und im Langzeit-QLC-Format sind, dann ist die verfügbare Datenkapazität die verbleibende offene Datenkapazität der Speichervorrichtung zu Beginn des Programmiervorgangs.

[0085] Ein Gesichtspunkt der vorliegenden Offenbarung bezieht sich auf eine Programmierertechnik, die der vorstehend erörterten ähnlich ist, wobei die Speichervorrichtung jedoch von einem SLC-Modus in einen MLC-Modus übergeht, wenn die SLC-Grenze erreicht ist, und dann nur in einen QLC-Modus übergeht, wenn eine MLC-Grenze erreicht ist. Somit gibt es einen Mittelleistungs-MLC-Modus zwischen dem Hochleistungs-SLC-Modus und dem Niederleistungs-QLC-Modus, und die Speichervorrichtung kann mehr Daten bei einer höheren Leistung programmieren, bevor sie in den Niederleistungs-QLC-Modus eintritt. Unter Bezugnahme auf **Fig. 12** weist die Speichervorrichtung gemäß diesem Gesichtspunkt der vorliegenden Offenbarung während des Betriebs im SLC-Modus einen ersten (höchsten) Leistungspegel beim Programmieren von Daten von null bis fünfundzwanzig Prozent (0-25 %) der verfügbaren Datenkapazität der Speichervorrichtung zu Beginn des Programmiervorgangs auf. Sobald die Speichervorrichtung ihre SLC-Grenze erreicht, schaltet sie den Betrieb vom SLC-Modus in den MLC-Modus um und programmiert Daten bei einem zweiten (mittleren) Leistungspegel von fünfundzwanzig bis fünfzig Prozent (25-50 %) der verfügbaren Datenkapazität zu Beginn des Programmiervorgangs. Sobald die Speichervorrichtung ihre MLC-Grenze erreicht, schaltet sie schließlich den Betrieb vom MLC-Modus in den QLC-Modus um und programmiert alle verbleibenden zu schreibenden Daten bei einem dritten (niedrigsten) Leistungspegel von fünfzig bis einhundert Prozent (50-100 %) der gesamten Datenkapazität. Mit anderen Worten programmiert die Speichervorrichtung zuerst Daten bei einer hohen Leistung bis zur SLC-Grenze, programmiert dann Daten bei einer mittleren Leistung bis zur MLC-Grenze und programmiert die Daten von da an schließlich bei einer niedrigen Leistung im QLC-Format.

[0086] In einer beispielhaften Ausführungsform empfängt eine vollständig gelöschte Speichervorrichtung mit einer verfügbaren Kapazität von 1 TB Daten Anweisungen von einem Host, 1 TB Daten in die Speichervorrichtung zu schreiben. Die Speichervorrichtung arbeitet anfänglich bei einer sehr hohen Leistung im SLC-Modus, um die ersten 256 GB Daten in die Speicherblöcke der Speichervorrichtung

zu programmieren. Dann schaltet die Speichervorrichtung den Betrieb vom SLC-Modus in den MLC-Modus um. Ohne Löschen der Speicherblöcke werden die Speicherzellen direkt aus dem SLC-Format in das MLC-Format programmiert, um die nächsten 256 GB Daten (512 GB insgesamt) in die Speichervorrichtung zu programmieren. Schließlich, wenn die Speichervorrichtung an der MLC-Grenze ist, schaltet die Speichervorrichtung den Betrieb vom MLC-Modus in den QLC-Modus um und programmiert die verbleibenden 512 GB (1 TB insgesamt) Daten unter Verwendung eines Niederleistungs-QLC-Programmiervorgangs. Die Speicherblöcke können nach Erreichen der MLC-Grenze und vor der QLC-Programmierung gelöscht werden.

[0087] Wenn der Programmiervorgang abgeschlossen ist, bevor die Speichervorrichtung die Datenkapazität der Speichervorrichtung erreicht, programmiert die Speichervorrichtung in einem Hintergrundvorgang alle Daten, die im SLC-Format oder MLC-Format gespeichert sind, in das QLC-Format, wodurch verfügbare Datenkapazität für einen zukünftigen Programmiervorgang gemäß der vorstehend erörterten hybriden SLC-/MLC-/QLC-Programmierertechnik in einem zukünftigen Programmiervorgang freigegeben wird, während die bereits geschriebenen Daten im QLC-Format unverändert bleiben. In diesem Fall werden die verfügbare Datenkapazität und die SLC- und MLC-Grenzen entsprechend reduziert.

[0088] **Fig. 13** ist ein Flussdiagramm, das die Schritte einer ersten Ausführungsform zum Programmieren einer Speichervorrichtung unter Verwendung der vorstehend genannten Programmierertechniken darstellt. Bei Schritt 1300 empfängt die Speichervorrichtung eine Schreibanforderung von einer Host-Vorrichtung.

[0089] Bei Schritt 1302 wird eine erste Wortleitung WL0 eines Speicherblocks in einem SLC-Programmiervorgang in das SLC-Format programmiert. Wie in **Fig. 14A** gezeigt, beginnt die Programmierung von WL0 damit, dass sich alle Speicherzellen von WL0 im gelöschten Datenzustand (Er) befinden. Als Nächstes wird eine erste Seite von Daten in die Speicherzellen der ausgewählten Wortleitung geschrieben, indem mindestens ein Programmierimpuls an WL0 angelegt wird und eine Programmierung in den Speicherzellen gesperrt wird, um in dem gelöschten Datenzustand (Er) zu bleiben, sodass die Schwellenspannungen nur der Speicherzellen, die in den ersten programmierten Datenzustand (S1) programmiert werden, angehoben werden, wie in **Fig. 14B** veranschaulicht. In diesem Beispiel wird den Speicherzellen im gelöschten Datenzustand (Er) das „1“-Bit zugewiesen, und den Speicherzellen im S1-Datenzustand wird das „0“-Bit zugewiesen.

[0090] Bei Schritten 1304a-N wird der Prozess von Schritt 1302 für alle verbleibenden Wortleitungen WL1-WLN eines Speicherblocks und für alle verfügbaren Speicherblöcke in der Speichervorrichtung wiederholt, bis entweder der Schreibvorgang abgeschlossen ist oder bis alle verfügbaren Speicherzellen aller verfügbaren Speicherblöcke in das SLC-Format programmiert wurden. Mit anderen Worten sind die Speicherblöcke geschlossen, wobei alle Speicherzellen im SLC-Format sind, bevor eine der Wortleitungen in das MLC-Format programmiert wird, und wenn die Schreibenanforderung vom Host abgeschlossen ist, bevor die Speichervorrichtung die SLC-Grenze erreicht, dann wird der Vorgang beendet. Wenn die Schreibenanforderung in diesem Stadium abgeschlossen wird, können die in das SLC-Format programmierten Daten in einem Hintergrundvorgang zu einem späteren Zeitpunkt aus dem SLC-Format in das QLC-Format umprogrammiert werden.

[0091] Wenn zusätzliche zu schreibende Daten nach Erreichen der SLC-Grenze verbleiben, fährt der Prozess mit Schritt 1306 fort. In Schritt 1306 gibt die Speichervorrichtung eine erste Seite von Daten der ersten Wortleitung WL0 unter Verwendung eines internen Datenlatch- (IDL-) Lesevorgangs wieder, kombiniert die erste Seite von Daten mit zusätzlichen Benutzerdaten und schreibt sowohl die erste Seite von Daten als auch die zusätzlichen Daten direkt in die Speicherzellen von WL0, ohne die Speicherzellen der ersten Wortleitung zu löschen, ohne die ausgewählte Wortleitung zu löschen. Wie in **Fig. 14C** und **14D** veranschaulicht, werden die einigen der Speicherzellen des Datenzustands Er und S1 an der Programmierung gehindert, sodass sie bei ihren jeweiligen Schwellenspannungen bleiben, und andere Speicherzellen werden programmiert, um ihre jeweiligen Schwellenspannungen auf einen von drei neuen Datenzuständen zu erhöhen. Genauer gesagt werden einige der Speicherzellen in dem alten S1-Datenzustand auf den neuen S1_MLC- und S3_MLC-Datenzustand programmiert, und einige der Speicherzellen des alten gelöschten Er-Datenzustands werden auf den neuen S2_MLC-Datenzustand programmiert. Da einige der Speicherzellen andere Datenzustände „überspringen“ (zum Beispiel Er bis S2_MLC und S1 bis S3_MLC), kann dieser Programmiervorgang als „Überspring“-Schreibvorgang bezeichnet werden. Die Speicherzellen, die in dem alten gelöschten Er-Datenzustand und dem alten S1-Datenzustand verbleiben, werden jetzt in dem neuen S0_MLC-Datenzustand rekategorisiert.

[0092] Wie gezeigt, werden nach der MLC-Programmierung die Speicherzellen der ausgewählten Wortleitung WL0 auf fünf unterschiedliche Verteilungskurven verteilt. In einer ersten beispielhaften Ausführungsform schließt der MLC-Programmiervor-

gang ein Verifizieren ein. Das heißt, die Speicherzellen setzen die Programmierung in einer Vielzahl von Programmierimpulsen fort, bis ihre jeweiligen Schwellenspannungen Verifizierungsspannungen überschreiten, die ihren beabsichtigten endgültigen Datenzuständen zugeordnet sind, an welchem Punkt die weitere Programmierung gesperrt ist. Die Programmierung der Wortleitung WL0 wird fortgesetzt, bis eine Schwellenanzahl von Speicherzellen ihre beabsichtigten Datenzustände erreicht ist oder bis eine maximale Anzahl von Programmierschleifen erreicht ist, an welchem Punkt bestimmt wird, dass die Programmierung fehlgeschlagen ist.

[0093] Die ersten zwei Verteilungskurven schließen alle Speicherzellen mit Schwellenspannungen ein, die kleiner als die S1_MLC-Verifizierungsspannung Vv1 sind und als im S0_MLC-Datenzustand befindlich angesehen werden. Die dritte Verteilungskurve schließt die Speicherzellen mit Schwellenspannungen zwischen der S1_MLC-Verifizierungsspannung Vv1 und einer S2_MLC-Verifizierungsspannung Vv2 ein und werden als im S1_MLC-Datenzustand befindlich angesehen. Die vierte Verteilungskurve schließt die Speicherzellen ein, die Schwellenspannungen zwischen der S2_MLC-Verifizierungsspannung Vv2 und der S3_MLC-Verifizierungsspannung Vv3 aufweisen und als im S2_MLC-Datenzustand befindlich angesehen werden. Die fünfte Verteilungskurve schließt die Speicherzellen mit Schwellenspannungen ein, die größer als die S3_MLC-Verifizierungsspannung Vv3 sind und als im S3_MLC-Datenzustand befindlich angesehen werden.

[0094] In der beispielhaften Ausführungsform sind dem S0_MLC-Datenzustand zwei Bits zugeordnet, z. B. „1 (erste Seite), 1 (zweite Seite)“, dem S1_MLC-Datenzustand sind Bits „0, 1“ zugeordnet, dem S2_MLC-Datenzustand sind die Bits „1, 0“ zugeordnet, und dem S3_MLC-Datenzustand sind die Bits „0, 0“ zugeordnet. In anderen Ausführungsformen können die Datenzustände verschiedenen Bits zugeordnet sein.

[0095] Bei Schritten 1308a-N wird der Prozess von Schritt 1306 für alle verbleibenden Wortleitungen WL1-WLN des Speicherblocks und für alle verfügbaren Speicherblöcke in der Speichervorrichtung wiederholt, bis entweder der Schreibvorgang abgeschlossen ist oder bis alle verfügbaren Speicherzellen aller verfügbaren Speicherblöcke in das MLC-Format programmiert wurden.

[0096] Während dieses Prozesses, wenn die Schreibenanforderung abgeschlossen ist, bevor die Speichervorrichtung die MLC-Grenze erreicht, dann wird der Vorgang beendet, und die Daten werden aus dem SLC- oder MLC-Format in das QLC-Format für eine Langzeitspeicherung in einem Hin-

tergrundvorgang umprogrammiert. Wenn die Schreibanforderung nicht abgeschlossen ist, bevor die Speichervorrichtung die MLC-Grenze erreicht, werden die Speicherzellen von mindestens einem Speicherblock 1310 bei Schritt 1310 gelöscht, und das Programmieren in das QLC-Format beginnt bei Schritt 1312. Der Programmiervorgang aus dem MLC-Format in das QLC-Format kann das Lesen der Daten eines Speicherblocks, wobei Daten vorübergehend an einem anderen Ort gespeichert werden (wie einem erweiterten Speicherblock, der nicht zu der verfügbaren Datenkapazität zählt), das Löschen des Speicherblocks und das erneute Programmieren der Wortleitungen des Speicherblocks, um die vorherigen Daten und zusätzliche Daten einzuschließen, einschließen.

[0097] Unter Bezugnahme auf **Fig. 15C-15F** schließt in einer zweiten beispielhaften Ausführungsform der Prozess des Programmierens jeder der Wortleitungen WL0-WLN aus dem SLC-Format in das MLC-Format bei den Schritten 1306 und 1308 drei Programmierimpulse und null Verifizierimpulse (3P0V) ein. In dem ersten Programmierimpuls (**Fig. 15C**) werden einige der Speicherzellen des alten S1-Datenzustands in den neuen S1_MLC-Datenzustand programmiert, ohne zu verifizieren. In dem zweiten Programmierimpuls (**Fig. 15D**) werden einige der Speicherzellen des alten gelöschten Er-Datenzustands in den neuen S2_MLC-Datenzustand programmiert, ohne zu verifizieren. In dem dritten Programmierimpuls (**Fig. 15E**) werden einige der Speicherzellen des alten S1-Datenzustands in den neuen S3_MLC-Datenzustand programmiert, ohne zu verifizieren.

[0098] Eine sehr große Spannungsschrittgröße dV_{pgm} wird zwischen Programmierimpulsen verwendet, um die Größe des Programmierimpulses V_{pgm} von dem ersten Programmierimpuls zu dem zweiten Programmierimpuls zu erhöhen. In jedem der drei Programmierimpulse werden die Speicherzellen, die sich in ihren endgültigen Datenzuständen befinden, zum Programmieren gesperrt. Das heißt, während des ersten Programmierimpulses werden die Speicherzellen, die für den S0_MLC-Datenzustand bestimmt sind, gesperrt; während des zweiten Programmierimpulses werden die Speicherzellen, die für die Datenzustände S0_MLC und S1_MLC bestimmt sind, gesperrt; und während des dritten Programmierimpulses werden die Speicherzellen, die für die Datenzustände S0_MLC, S1_MLC und S2_MLC bestimmt sind, gesperrt. Keiner dieser Programmierimpulse folgt einem Verifiziervorgang, und somit wird der Programmiervorgang nach dem dritten Programmierimpuls abgeschlossen. Durch Überspringen der Verifizierung kann in dieser Ausführungsform die Leistung

durch Reduzieren der Programmierzeit verbessert werden.

[0099] In einigen Ausführungsformen kann die Speichervorrichtung zum Schreiben in das TLC-Format eingerichtet sein, statt beim Erreichen der MLC-Grenze mit dem Schreiben aus dem MLC-Format in das QLC-Format fortzufahren.

[0100] Verschiedene Begriffe werden hierin verwendet, um auf bestimmte Systemkomponenten Bezug zu nehmen. Verschiedene Firmen können auf die gleiche oder ähnliche Komponente mit unterschiedlichen Namen Bezug nehmen und diese Beschreibung beabsichtigt nicht, zwischen Komponenten zu unterscheiden, die sich im Namen, jedoch nicht in der Funktion unterscheiden. Soweit verschiedene in der folgenden Offenbarung beschriebene Funktionseinheiten als „Module“ bezeichnet werden, soll eine solche Charakterisierung die Bandbreite möglicher Implementierungsmechanismen nicht unangemessen einschränken. Zum Beispiel könnte ein „Modul“ als Hardwareschaltung implementiert werden, die kundenspezifische VLSI-Schaltungen (Very Large Scale Integration) oder Gate-Arrays oder handelsübliche Halbleiter, die Logikchips, Transistoren oder andere diskrete Komponenten einschließen, aufweist. In einem weiteren Beispiel kann ein Modul auch in einer programmierbaren Hardwarevorrichtung wie einem feldprogrammierbaren Gate Array (FPGA), einer programmierbaren Array-Logik, einer programmierbaren Logikvorrichtung oder dergleichen implementiert sein. Außerdem kann ein Modul auch, zumindest teilweise, durch Software implementiert werden, die von verschiedenen Prozessorentypen ausgeführt wird. Zum Beispiel kann ein Modul ein Segment von ausführbarem Code aufweisen, das einen oder mehrere physische oder logische Blöcke von Computeranweisungen enthält, die in ein Objekt, einen Prozess oder eine Funktion übersetzt werden. Es ist auch nicht erforderlich, dass die ausführbaren Abschnitte eines solchen Moduls physisch zusammen angeordnet sind, sondern sie können auch unterschiedliche Anweisungen aufweisen, die an verschiedenen Orten gespeichert sind und die, wenn sie zusammen ausgeführt werden, das identifizierte Modul aufweisen und den erklärten Zweck dieses Moduls erfüllen. Der ausführbare Code kann nur eine einzige Anweisung oder einen Satz mehrerer Anweisungen aufweisen, aber auch über verschiedene Codesegmente oder verschiedene Programme oder über mehrere Speichervorrichtungen usw. verteilt sein. Bei einer Software- oder Teilsoftwaremodul-Implementierung können die Softwareabschnitte auf einem oder mehreren computerlesbaren und/oder ausführbaren Speicherungsmedien gespeichert werden, die ein elektronisches, magnetisches, optisches, elektromagnetisches, infrarotes oder halbleiterbasiertes System, eine Einrichtung oder Vorrichtung oder jede geeig-

nete Kombination davon einschließen, aber nicht darauf beschränkt sind. Im Allgemeinen kann für die Zwecke der vorliegenden Offenbarung ein computerlesbares und/oder ausführbares Speicherungsmedium jedes greifbare und/oder nicht-transistorische Speicherungsmedium sein, das in der Lage ist, ein Programm zur Verwendung durch oder in Verbindung mit einem System, einer Einrichtung, einem Prozessor oder einer Vorrichtung zur Befehlsausführung zu enthalten und/oder zu speichern.

[0101] In ähnlicher Weise kann für die Zwecke der vorliegenden Offenbarung der Begriff „Komponente“ jede greifbare, physische und nicht-transistorische Vorrichtung aufweisen. Zum Beispiel kann eine Komponente in Form einer Hardwarelogikschaltung vorliegen, die kundenspezifische VLSI-Schaltungen, Gate-Arrays oder andere integrierte Schaltungen aufweist, oder sie kann aus handelsüblichen Halbleitern bestehen, die Logikchips, Transistoren oder andere diskrete Komponenten einschließen, oder aus anderen geeigneten mechanischen und/oder elektronischen Vorrichtungen. Darüber hinaus kann eine Komponente auch in programmierbaren Hardware-Vorrichtungen wie feldprogrammierbaren Gate-Arrays (FPGA), programmierbarer Array-Logik, programmierbaren Logikvorrichtungen usw. implementiert sein. Außerdem kann eine Komponente eine oder mehrere integrierte Schaltungsvorrichtungen auf Siliciumbasis, wie Chips, Die, Die-Ebenen und Gehäuse, oder andere diskrete elektrische Vorrichtungen aufweisen, die in einer elektrischen Kommunikationskonfiguration mit einer oder mehreren anderen Komponenten über elektrische Leiterbahnen, z. B. auf einer Leiterplatte (PCB) oder dergleichen, eingerichtet sind. Dementsprechend kann ein Modul, wie vorstehend definiert, in bestimmten Ausführungsformen durch eine Komponente verkörpert oder als Komponente implementiert werden, und in einigen Fällen können die Begriffe Modul und Komponente austauschbar verwendet werden.

[0102] Wenn hier der Begriff „Schaltung“ verwendet wird, schließt er eine oder mehrere elektrische und/oder elektronische Komponenten ein, die einen oder mehrere leitfähige Pfade bilden, die es ermöglichen, dass elektrischer Strom fließt. Eine Schaltung kann in Form einer Konfiguration mit geschlossenem oder offenem Regelkreis eingerichtet sein. In einer Konfiguration mit geschlossenem Regelkreis können die Schaltungskomponenten einen Rückführweg für den elektrischen Strom bereitstellen. Im Gegensatz dazu können die Schaltungskomponenten in einer Konfiguration mit offenem Regelkreis immer noch als eine Schaltung angesehen werden, auch wenn sie keinen Rückführweg für den elektrischen Strom einschließen. Zum Beispiel kann eine integrierte Schaltung unabhängig davon als eine Schaltung bezeichnet werden, ob die integrierte Schaltung an Masse gekoppelt ist (als Rückführweg für elektri-

schen Strom) oder nicht. In bestimmten beispielhaften Ausführungsformen kann eine Schaltung einen Satz integrierter Schaltungen, eine einzige integrierte Schaltung oder einen Abschnitt einer integrierten Schaltung aufweisen. Zum Beispiel kann eine Schaltung kundenspezifische VLSI-Schaltungen, Gate-Arrays, Logikschaltungen und/oder andere Formen integrierter Schaltungen einschließen und kann handelsübliche Halbleiter, wie Logikchips, Transistoren oder andere diskrete Vorrichtungen einschließen. In einem weiteren Beispiel kann eine Schaltung eine oder mehrere integrierte Schaltungsvorrichtungen auf Siliciumbasis, wie Chips, Die, Die-Ebenen und Gehäuse, oder andere diskrete elektrische Vorrichtungen aufweisen, die in einer elektrischen Kommunikationskonfiguration mit einer oder mehreren anderen Komponenten über elektrische Leiterbahnen, zum Beispiel einer Schaltungsplatine (PCB), eingerichtet sind. Eine Schaltung könnte auch als synthetisierte Schaltung in Bezug auf eine programmierbare Hardwarevorrichtung wie ein feldprogrammierbares Gate-Array (FPGA), programmierbare Array-Logik und/oder programmierbare Logikvorrichtungen usw. implementiert werden. In anderen beispielhaften Ausführungsformen kann eine Schaltung ein Netzwerk aus nicht integrierten elektrischen und/oder elektronischen Komponenten aufweisen (mit oder ohne integrierte Schaltungsvorrichtungen). Dementsprechend kann ein Modul, wie vorstehend definiert, in bestimmten Ausführungsformen durch eine Schaltung verkörpert oder als solche implementiert sein.

[0103] Es versteht sich, dass die hierin offenbarten Ausführungsformen in einigen Beispielen einen oder mehrere Mikroprozessoren und bestimmte gespeicherte Computerprogrammanweisungen aufweisen können, die den einen oder die mehreren Mikroprozessoren steuern, um in Verbindung mit bestimmten Nicht-Prozessor-Schaltungen und anderen Elementen einige, die meisten oder alle der hier offenbarten Funktionen zu implementieren. Alternativ könnten einige oder alle Funktionen durch eine Zustandsmaschine implementiert werden, die keine gespeicherten Programmanweisungen aufweist, oder in einer oder mehreren anwendungsspezifischen integrierten Schaltungen (ASICs) oder feldprogrammierbaren Gate-Arrays (FPGAs), in denen jede Funktion oder einige Kombinationen bestimmter Funktionen als kundenspezifische Logik implementiert sind. Eine Kombination dieser Ansätze kann ebenfalls verwendet werden. Weiterhin sind nachfolgende Bezugnahmen auf eine „Steuerung“ so definiert, dass sie einzelne Schaltungskomponenten, eine anwendungsspezifische integrierte Schaltung (ASIC), einen Mikrocontroller mit Steuersoftware, einen digitalen Signalprozessor (DSP), ein feldprogrammierbares Gate-Array (FPGA) und/oder einen Prozessor mit Steuersoftware oder Kombinationen davon aufweisen.

[0104] Außerdem sind die Begriffe „koppeln“, „gekoppelt“ oder „koppelt“, wo sie hierin verwendet werden, entweder als direkte oder indirekte Verbindung zu verstehen. Wenn also eine erste Vorrichtung an eine zweite Vorrichtung koppelt oder an diese gekoppelt ist, kann diese Verbindung durch eine direkte Verbindung oder durch eine indirekte Verbindung über andere Vorrichtungen (oder Komponenten) und Verbindungen erfolgen.

[0105] Was die Verwendung von Ausdrücken wie „eine Ausführungsform“, „eine (1) Ausführungsform“, eine „beispielhafte Ausführungsform“, „eine bestimmte Ausführungsform“ oder ähnlicher Begriffe betrifft, so sollen diese Begriffe anzeigen, dass ein spezifisches Merkmal, eine spezifische Struktur, eine spezifische Funktion, ein spezifischer Vorgang oder eine spezifische Eigenschaft, die im Zusammenhang mit der Ausführungsform beschrieben werden, in mindestens einer Ausführungsform der vorliegenden Offenbarung enthalten ist. Daher können sich Formulierungen wie „in einer (1) Ausführungsform“, „in einer Ausführungsform“, „in einer beispielhaften Ausführungsform“ usw. auf dieselbe Ausführungsform beziehen, müssen es aber nicht, sondern bedeuten „eine oder mehrere, aber nicht alle Ausführungsformen“, sofern nicht ausdrücklich etwas anderes angegeben ist. Weiterhin werden die Begriffe „aufweisend“, „mit“, „einschließend“ und Variationen davon in einer offenen Weise verwendet und sollten daher so ausgelegt werden, dass sie „einschließlich, aber nicht beschränkt auf ...“ bedeuten, sofern nicht ausdrücklich etwas anderes angegeben ist. Außerdem schließt ein Element, dem „weist ein/eine ... auf“ vorangestellt ist, ohne weitere Einschränkungen nicht aus, dass es zusätzliche identische Elemente in dem betreffenden Prozess, Verfahren, System, Artikel oder der Einrichtung gibt, die das Element einschließen.

[0106] Die Begriffe „ein/e/s“, und „der/die/das“ beziehen sich auch auf „eines oder mehrere“, sofern nicht ausdrücklich etwas anderes angegeben ist. Darüber hinaus zeigt die Formulierung „mindestens eines von A und B“, wie sie hierin und/oder in den folgenden Ansprüchen verwendet werden kann, wobei A und B Variablen sind, die ein bestimmtes Objekt oder Attribut anzeigen, eine Auswahl von A oder B oder sowohl A als auch B an, ähnlich wie die Formulierung „und/oder“. Sind in einer solchen Formulierung mehr als zwei Variablen vorhanden, so ist diese Formulierung so definiert, dass sie nur eine der Variablen, eine beliebige der Variablen, eine beliebige Kombination (oder Unterkombination) einer der Variablen und alle Variablen einschließt.

[0107] Weiterhin wird hier der Begriff „ungefähr“ oder „etwa“ auf alle numerischen Werte angewandt, unabhängig davon, ob sie ausdrücklich angegeben werden oder nicht. Diese Begriffe beziehen sich im

Allgemeinen auf einen Bereich von Zahlenwerten, die ein Fachmann als äquivalent zu den angegebenen Werten ansehen würde (z. B. weil sie dieselbe Funktion oder dasselbe Ergebnis aufweisen). In bestimmten Fällen können diese Begriffe numerische Werte einschließen, die auf die nächste signifikante Figur gerundet sind.

[0108] Darüber hinaus bedeutet eine Aufzählung von Elementen, die hierin dargelegt sind, nicht, dass einige oder alle der aufgeführten Elemente sich gegenseitig ausschließen und/oder einander einschließen, sofern nicht ausdrücklich etwas anderes angegeben ist. Weiterhin ist der Begriff „Satz“, wie er hierin verwendet wird, so auszulegen, dass er „einen oder mehrere“ bedeutet, und im Fall von „Sätzen“ ist er so auszulegen, dass er ein Vielfaches von (oder eine Vielzahl von) „einem oder mehreren“ und/oder „einem oder mehr“ gemäß der Mengenlehre bedeutet, sofern nicht ausdrücklich etwas anderes angegeben ist.

[0109] Die vorhergehende detaillierte Beschreibung wurde zu Zwecken der Veranschaulichung und Beschreibung vorgelegt. Es ist nicht beabsichtigt umfassend zu sein oder auf die genaue angegebene Form zu beschränken. Viele Modifikationen und Variationen sind unter Berücksichtigung der vorstehend genannten Beschreibung möglich. Die beschriebenen Ausführungsformen wurden gewählt, um die Prinzipien der Technologie und ihre praktische Anwendung am besten zu erklären, um dadurch anderen Fachleuten zu ermöglichen, die Erfindung in verschiedenen Ausführungsformen und mit verschiedenen Modifikationen, wie sie für die jeweilige beabsichtigte Verwendung geeignet sind, am besten zu nutzen. Der Schutzzumfang der Technologie wird durch die hieran angehängten Ansprüche definiert.

Patentansprüche

1. Verfahren zum Programmieren einer Speichervorrichtung, umfassend die Schritte:
 Vorbereiten einer Vielzahl von Speicherblöcken, wobei die Speicherblöcke ein Array von Speicherzellen einschließen, die programmiert werden können, um ein oder mehrere Datenbits pro Speicherzelle zu halten;
 Empfangen einer Datenschreibanweisung;
 Programmieren der Speicherzellen der Speicherblöcke in ein Format mit einem Bit pro Speicherzelle (SLC) mit einem ersten SLC-Datenzustand und einem zweiten SLC-Datenzustand;
 als Reaktion darauf, dass die Daten, die im SLC-Format in die Speicherzellen der Speicherblöcke programmiert werden, vor dem Abschluss der Datenschreibanweisung eine SLC-Grenze erreichen, ohne die in das SLC-Format programmierten Speicherzellen zu löschen, Programmieren von min-

destens einigen der Vielzahl von Speicherzellen aus dem SLC-Format in ein Format mit zwei Bits pro Speicherzelle (MLC); und wobei das Programmieren der Speicherzellen aus dem SLC-Format in das MLC-Format ein Verhindern der Programmierung einiger der Speicherzellen in dem ersten und zweiten SLC-Datenzustand einschließt, um einen ersten MLC-Datenzustand zu bilden, und das Programmieren einiger der Speicherzellen des ersten und des zweiten SLC-Datenzustands, um einen zweiten MLC-Datenzustand, einen dritten MLC-Datenzustand und einen vierten MLC-Datenzustand zu bilden.

2. Verfahren zum Programmieren der Speichervorrichtung gemäß Anspruch 1, wobei das Verfahren als Reaktion darauf, dass eine MLC-Grenze erreicht ist, weiterhin den Schritt des Programmierens der Speicherzellen von mindestens einigen der Vielzahl von Speicherblöcken aus dem MLC-Format in ein Format mit drei oder mehr Bits pro Speicherzelle einschließt.

3. Verfahren zum Programmieren der Speichervorrichtung gemäß Anspruch 2, wobei vor dem Programmieren der Speicherzellen von mindestens einem der Speicherblöcke aus dem MLC-Format in das Format mit drei oder mehr Bits pro Speicherzelle der mindestens eine Speicherblock gelöscht wird.

4. Verfahren zum Programmieren der Speichervorrichtung gemäß Anspruch 3, wobei das Format mit drei oder mehr Bits pro Speicherzelle ein Format mit vier Bits pro Speicherzelle (QLC) ist.

5. Verfahren zum Programmieren der Speichervorrichtung gemäß Anspruch 2, wobei die SLC-Grenze erreicht ist, wenn die Speicherzellen aller verfügbaren Speicherblöcke in das SLC-Format programmiert sind.

6. Verfahren zum Programmieren der Speichervorrichtung gemäß Anspruch 5, wobei die MLC-Grenze erreicht ist, wenn die Speicherzellen aller verfügbaren Speicherblöcke in das MLC-Format programmiert sind.

7. Verfahren zum Programmieren der Speichervorrichtung gemäß Anspruch 2, das weiterhin den Schritt des Programmierens jeglicher Daten im SLC-Format oder im MLC-Format in das Format mit drei oder mehr Bits pro Speicherzelle in einem Hintergrundvorgang einschließt.

8. Verfahren zum Programmieren der Speichervorrichtung gemäß Anspruch 1, wobei der Schritt des Programmierens der Speicherzellen von mindestens einigen der Speicherblöcke aus dem SLC-Format in das MLC-Format eine Vielzahl von Pro-

grammierschleifen einschließt, die auf jede Wortleitung angewendet werden, die Speicherzellen enthält, die programmiert werden, wobei die Programmierschleifen jeweils einen Programmierimpuls und mindestens einen Prüfpuls einschließen.

9. Verfahren zum Programmieren der Speichervorrichtung gemäß Anspruch 1, wobei der Schritt des Programmierens der Speicherzellen von mindestens einigen der Speicherblöcke aus dem SLC-Format in das MLC-Format drei Programmierimpulse und keine Verifizierungsimpulse einschließt, die an jede Wortleitung angelegt werden, die Speicherzellen enthält, die programmiert werden.

10. Speichervorrichtung, umfassend: eine Vielzahl von Speicherblöcken, die eine Vielzahl von Speicherzellen einschließen, die in einer Vielzahl von Wortleitungen angeordnet sind; eine Steuerschaltlogik in Kommunikation mit der Vielzahl von Speicherblöcken, wobei die Steuerschaltlogik eingerichtet ist zum; Empfangen einer Datenschreibenweisung, Programmieren der Speicherzellen der Speicherblöcke in ein Format mit einem Bit pro Speicherzelle (SLC) mit einem ersten SLC-Datenzustand und einem zweiten SLC-Datenzustand, als Reaktion darauf, dass die Daten, die im SLC-Format in die Speicherzellen der Speicherblöcke programmiert werden, vor dem Abschluss der Datenschreibenweisung eine SLC-Grenze erreichen, ohne die in das SLC-Format programmierten Speicherzellen zu löschen, Programmieren von mindestens einigen der Vielzahl von Speicherzellen aus dem SLC-Format in ein Format mit zwei Bits pro Speicherzelle (MLC), und wobei beim Programmieren der Speicherzellen aus dem SLC-Format in das MLC-Format die Steuerschaltlogik ein Programmieren einiger der Speicherzellen in den ersten und zweiten SLC-Datenzustand, um einen erste MLC-Datenzustand zu bilden, verhindert und einige der Speicherzellen des ersten und zweiten SLC-Datenzustands programmiert, um einen zweiten MLC-Datenzustand, einen dritten MLC-Datenzustand und einen vierten MLC-Datenzustand zu bilden.

Es folgen 15 Seiten Zeichnungen

Anhängende Zeichnungen

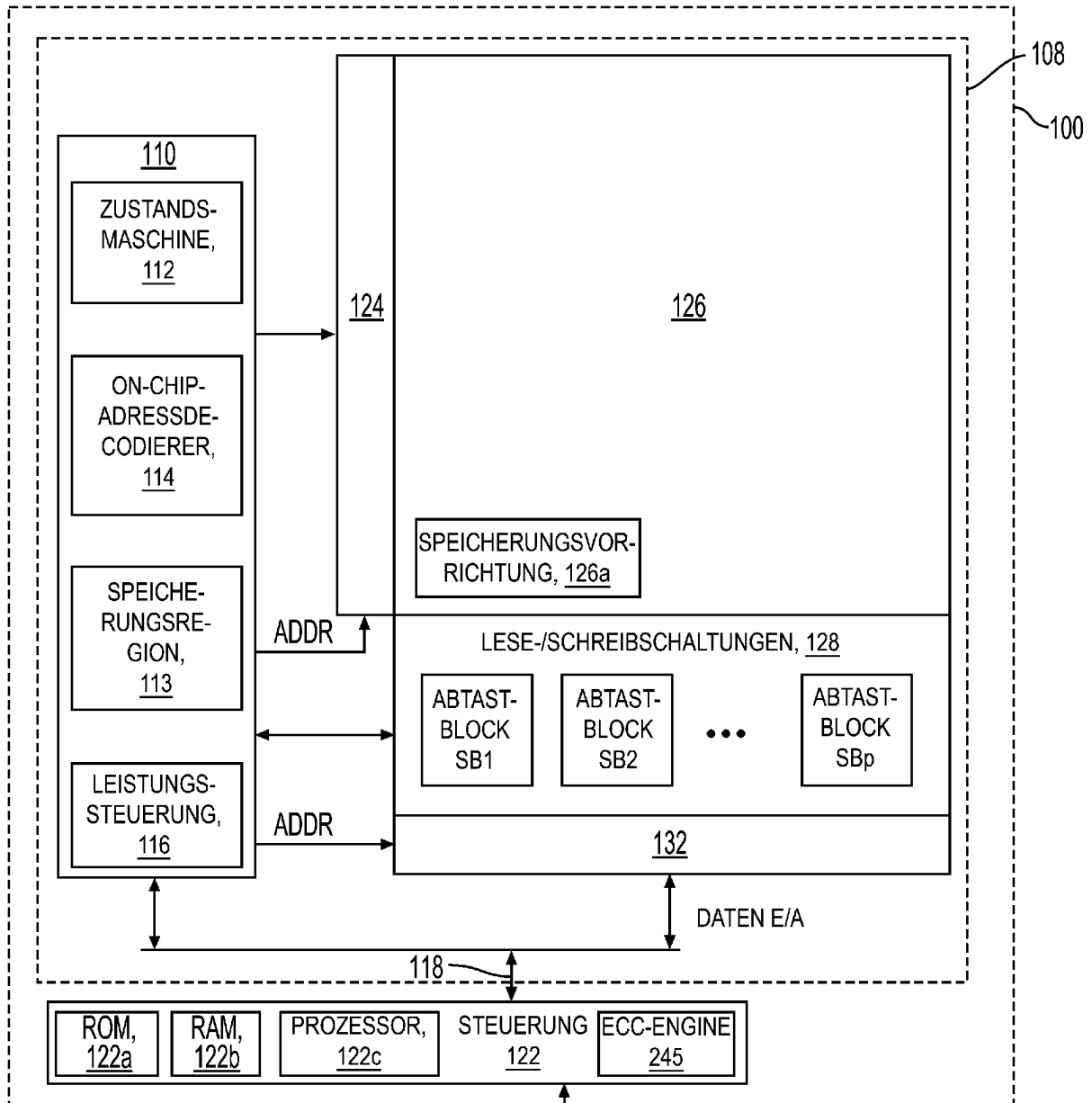
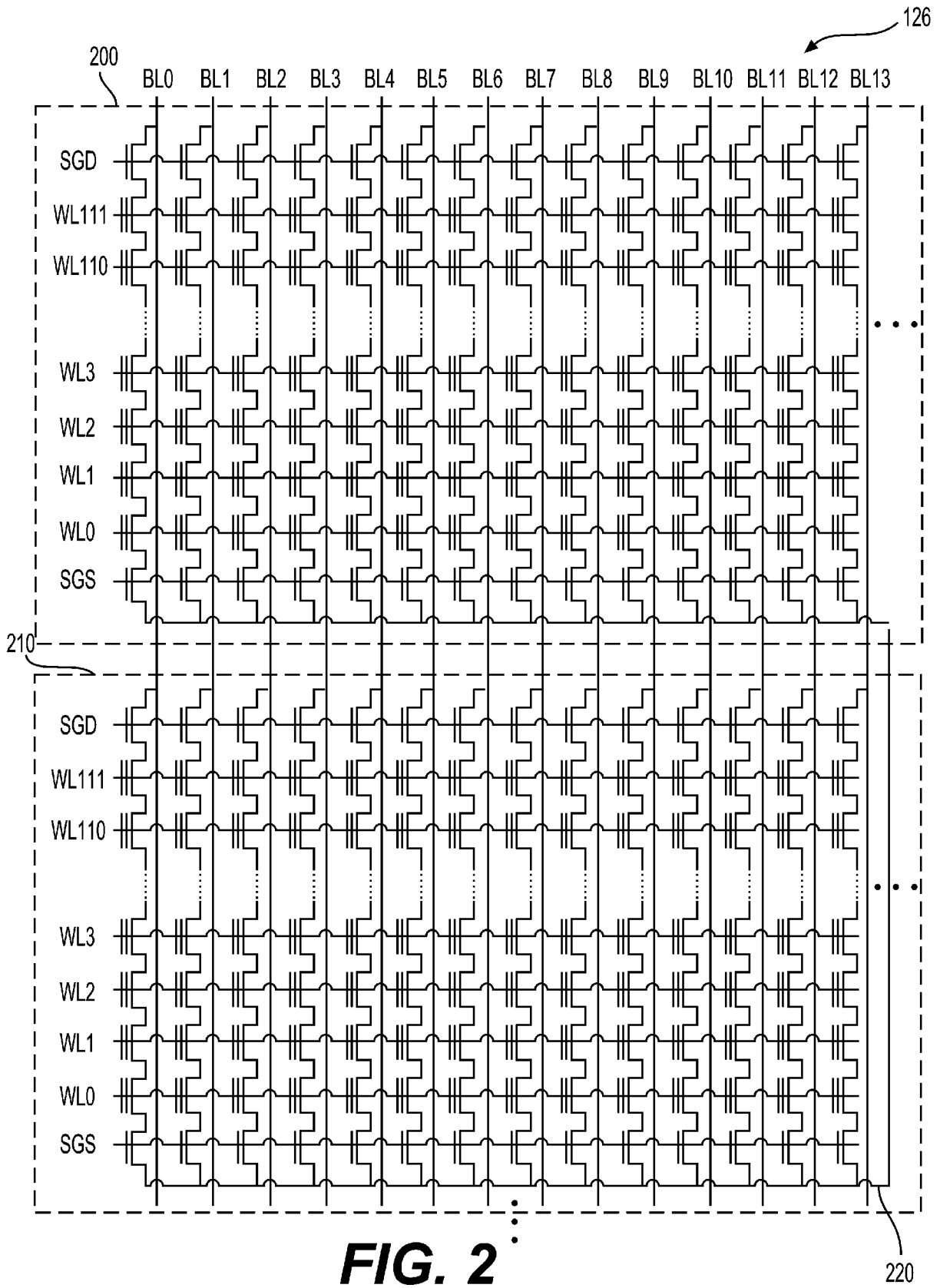


FIG. 1A



FIG. 1B



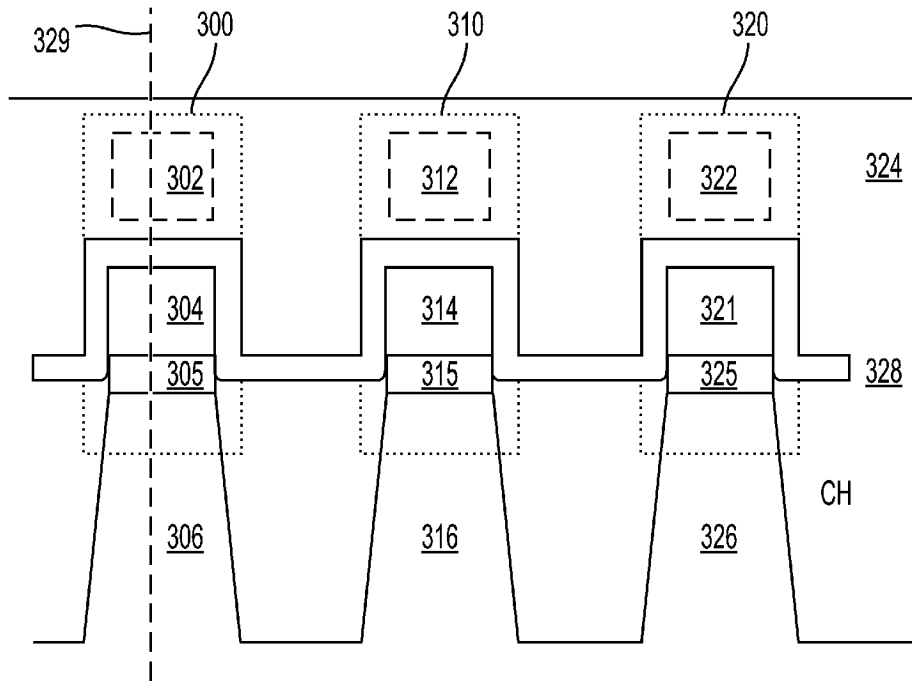


FIG. 3A

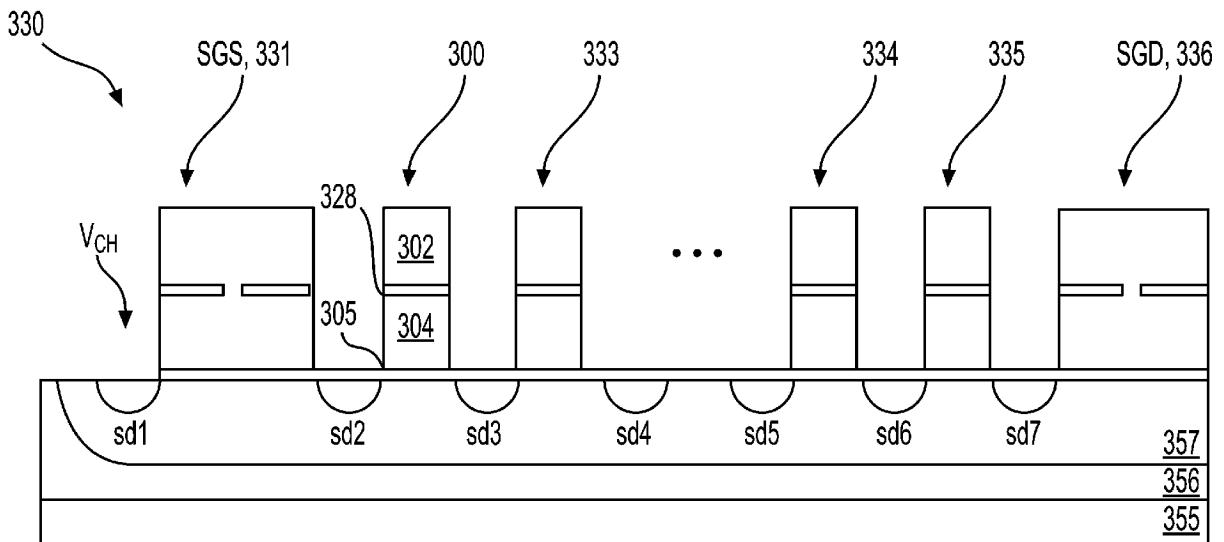


FIG. 3B

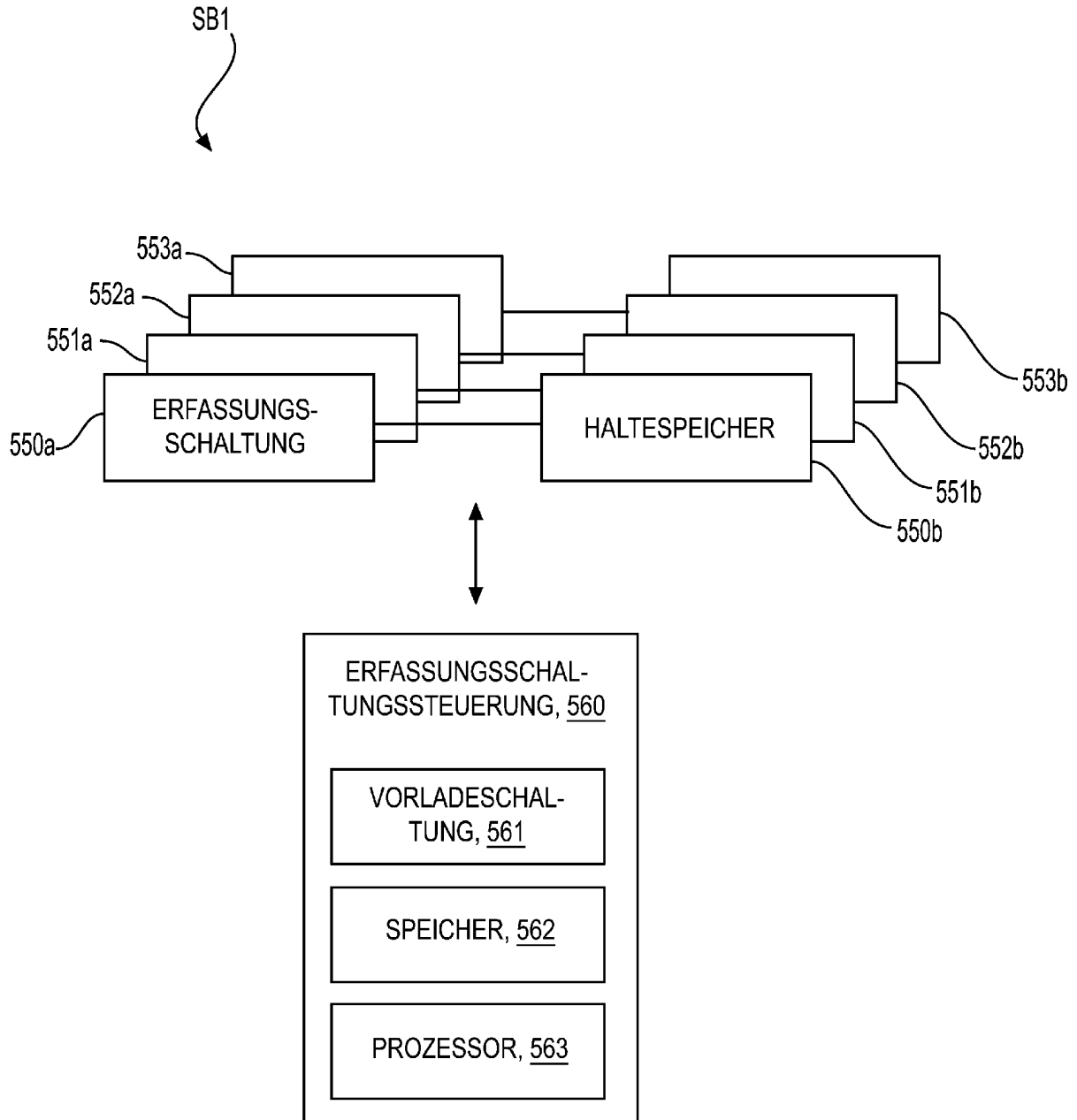


FIG. 5

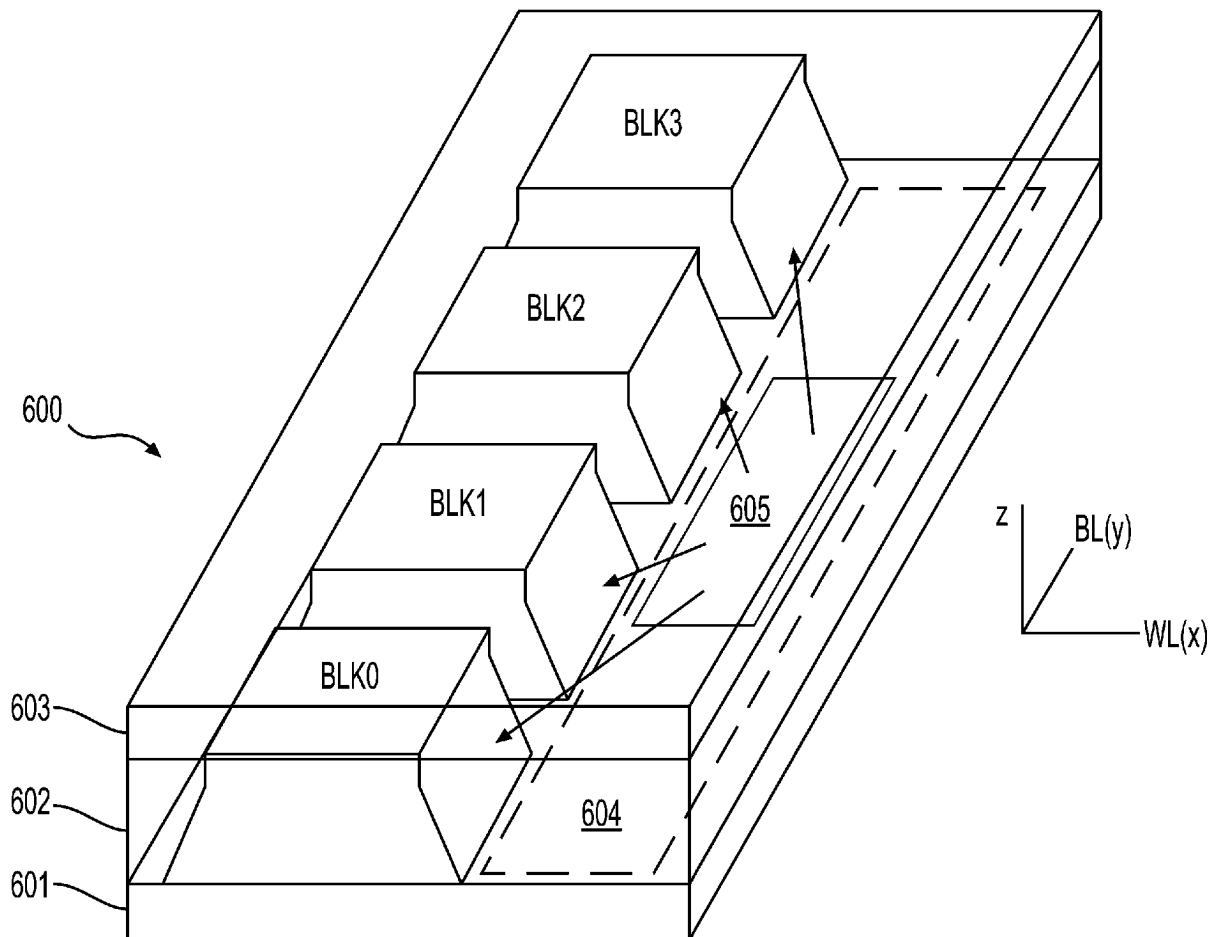


FIG. 6A

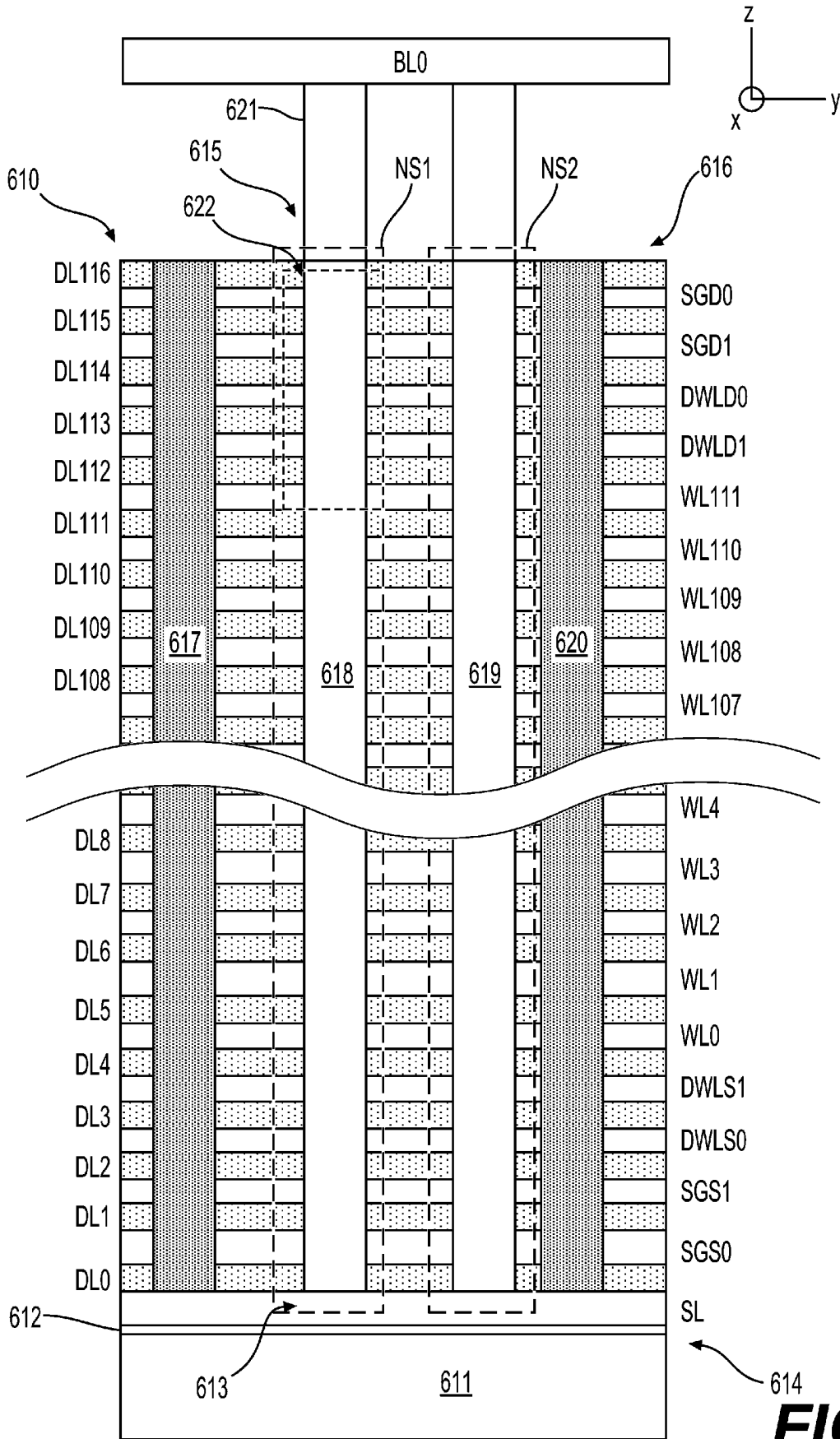


FIG. 6B

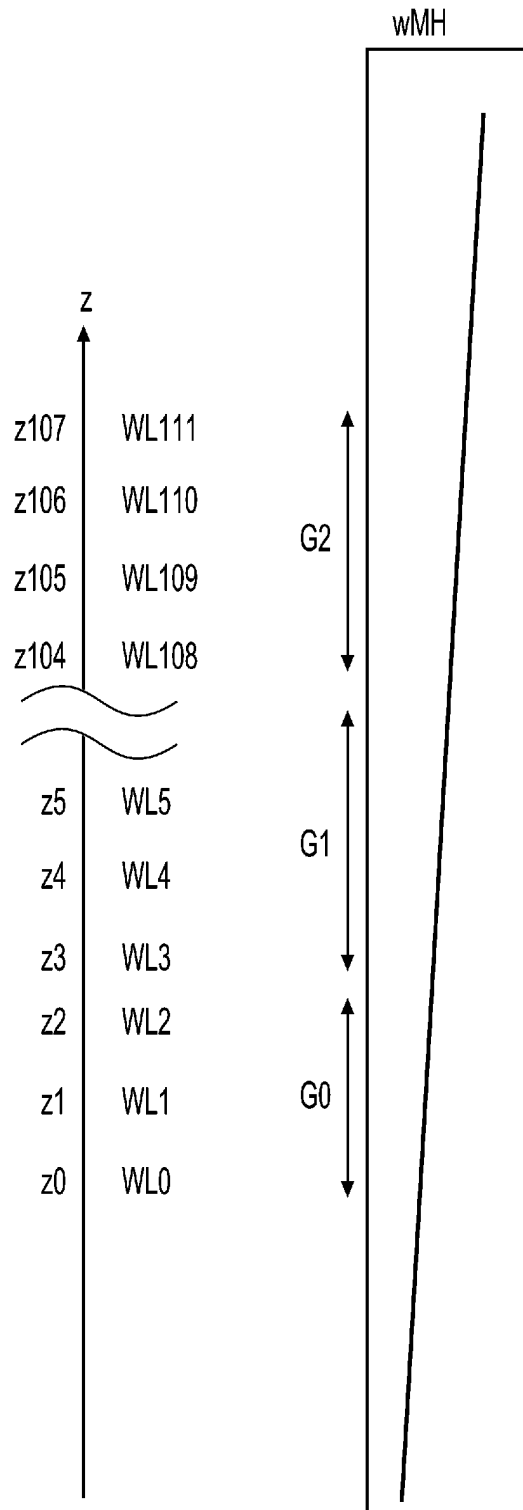


FIG. 6C

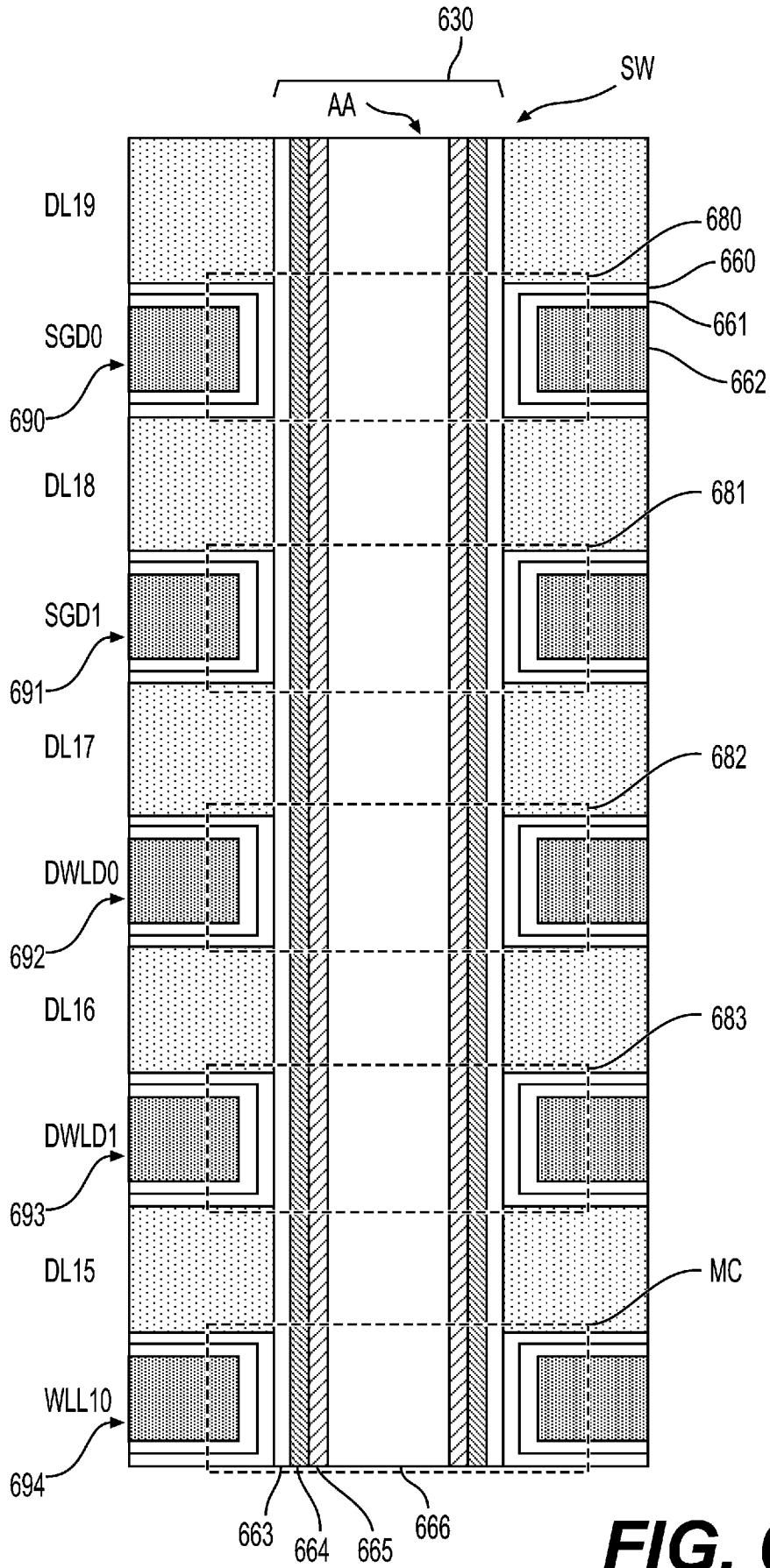


FIG. 6D

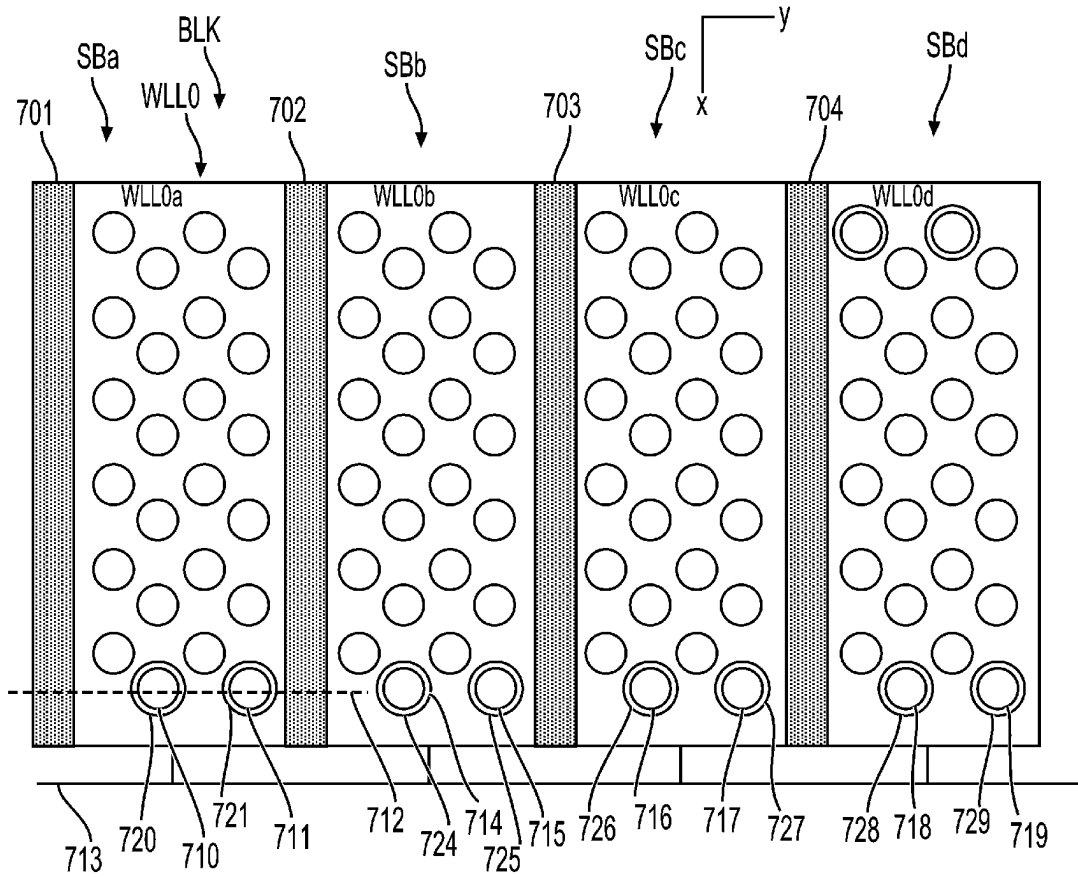


FIG. 7A

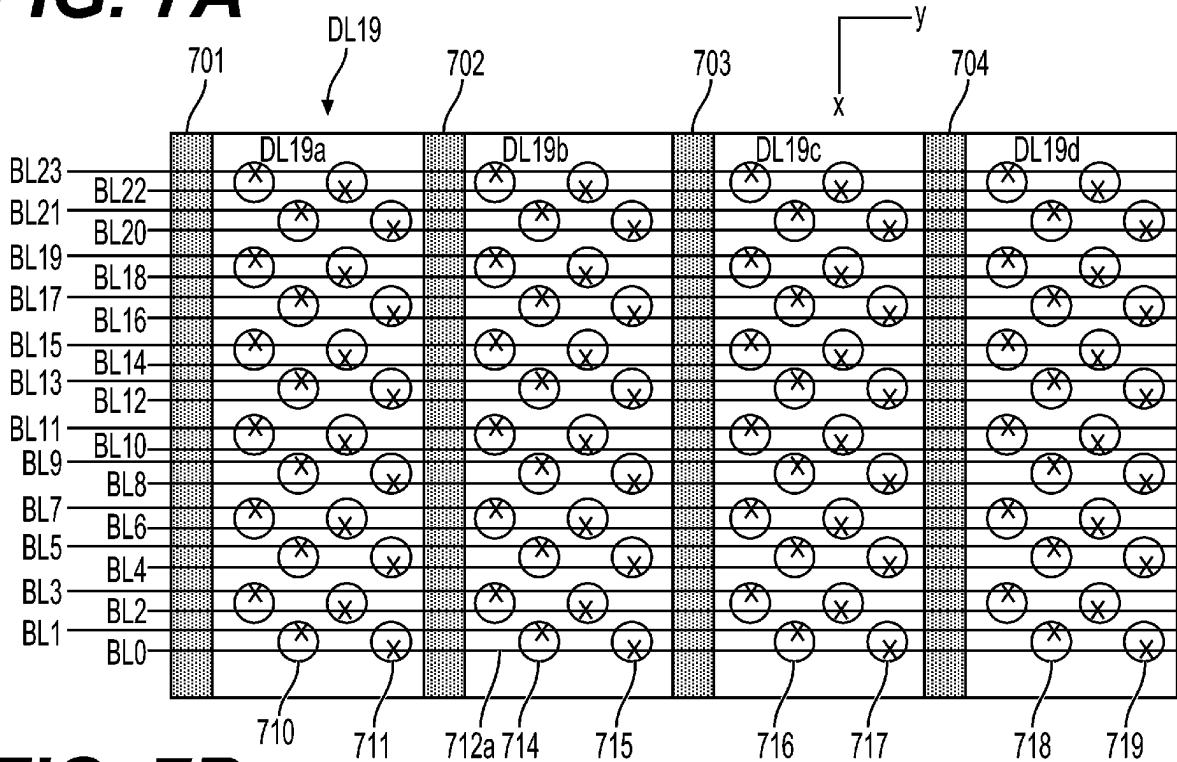


FIG. 7B

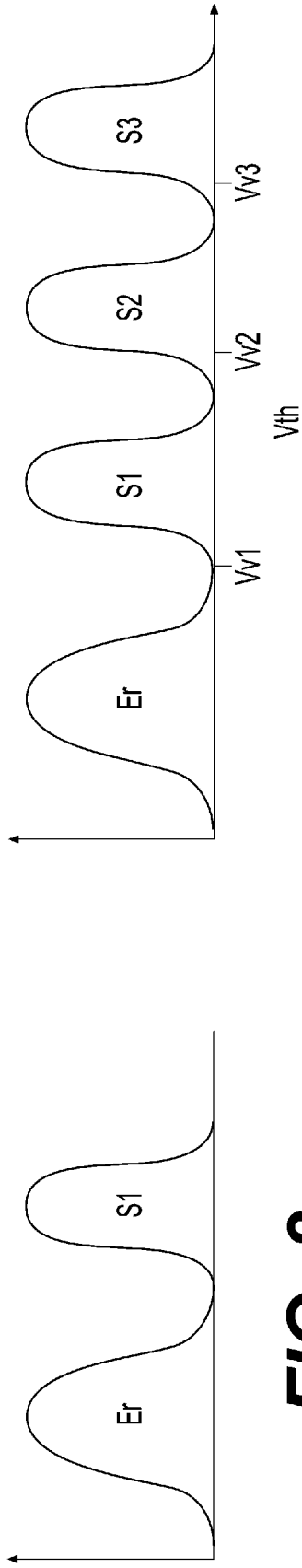


FIG. 8

FIG. 9

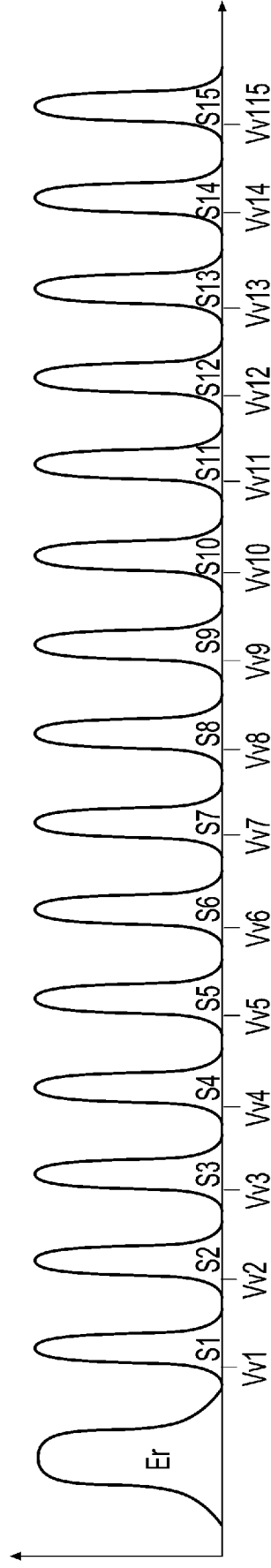
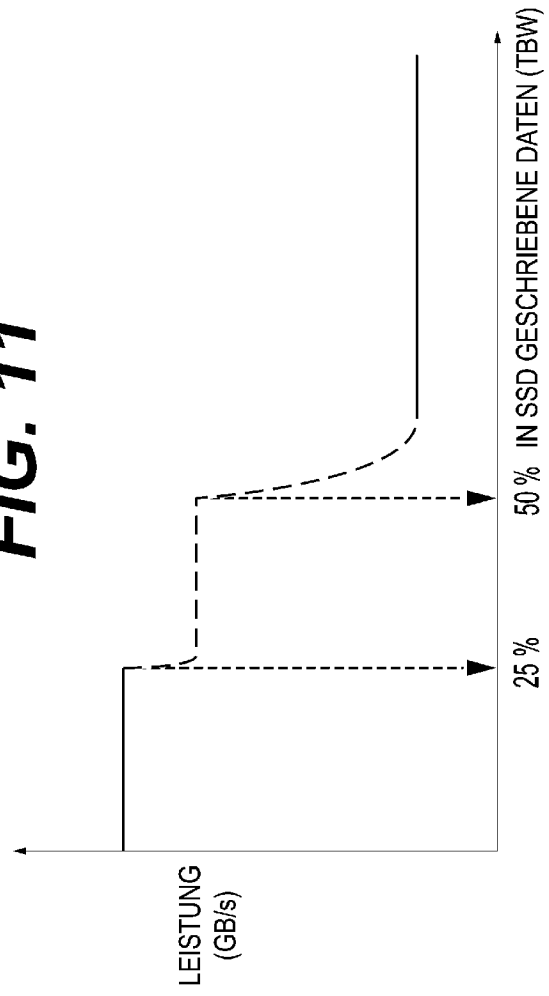
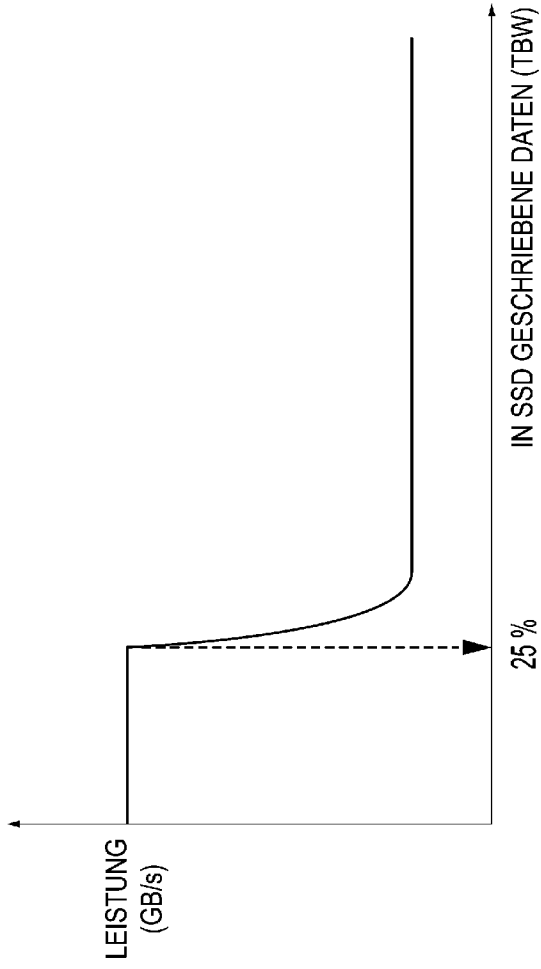


FIG. 10



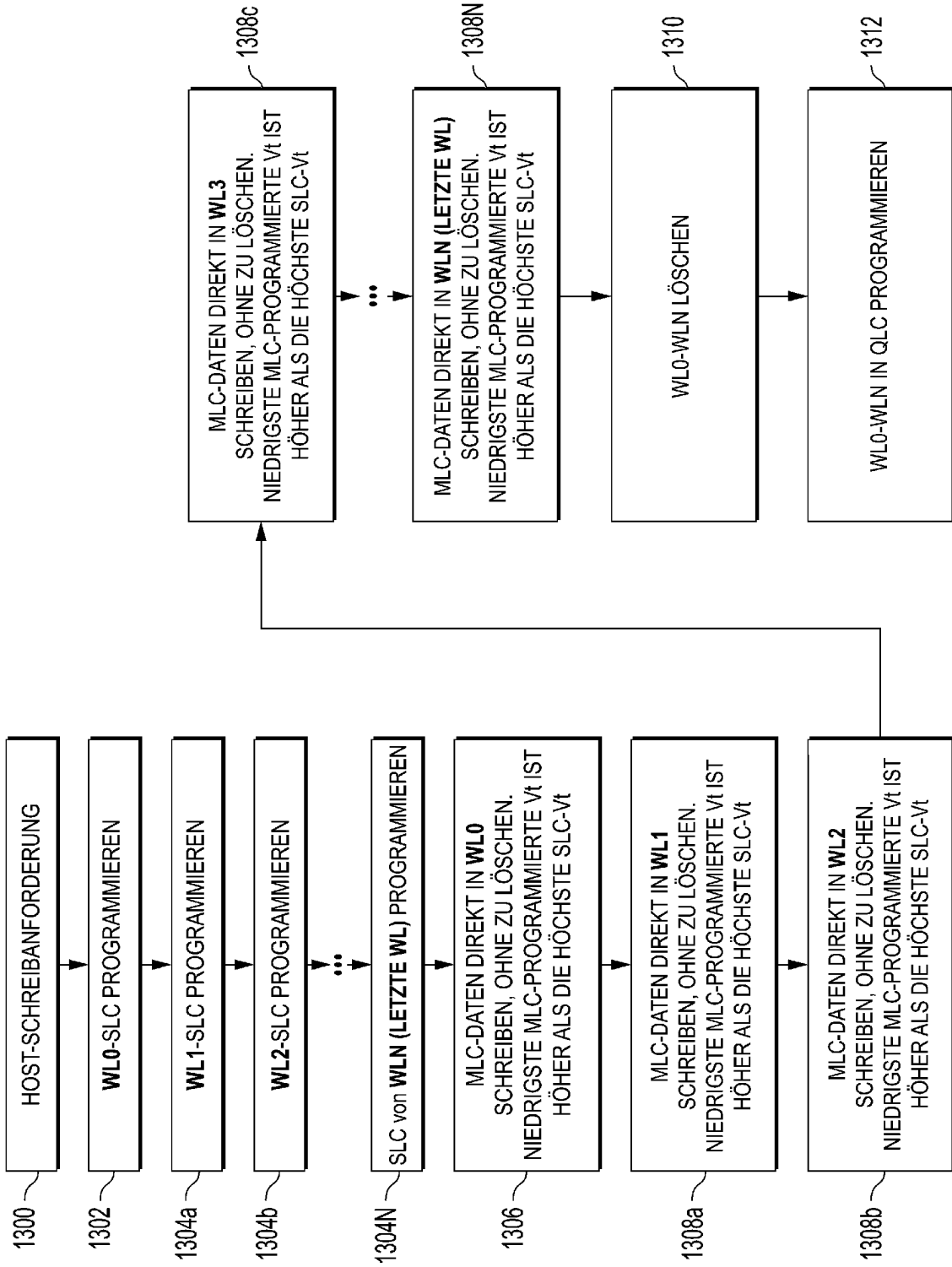


FIG. 13

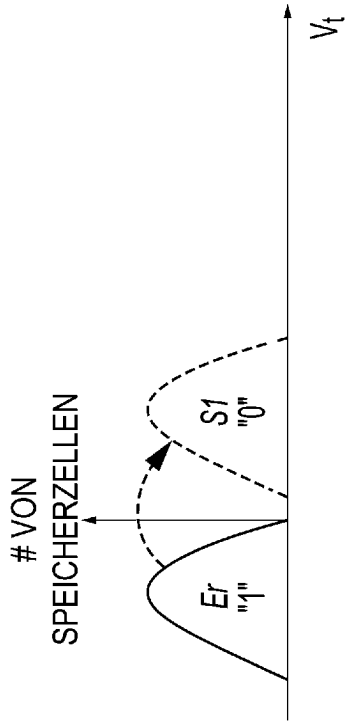


FIG. 14B

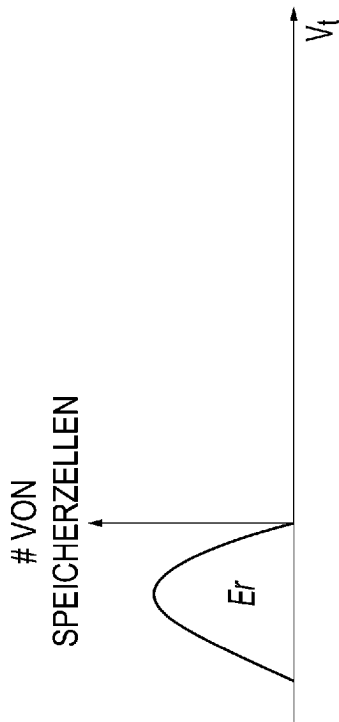


FIG. 14A

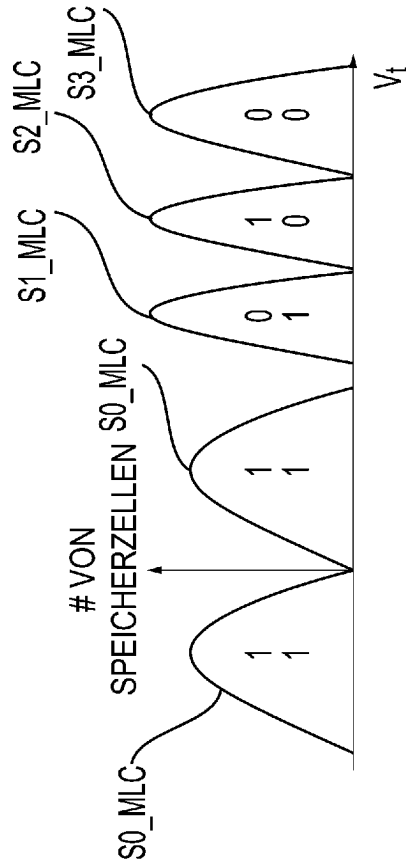


FIG. 14D

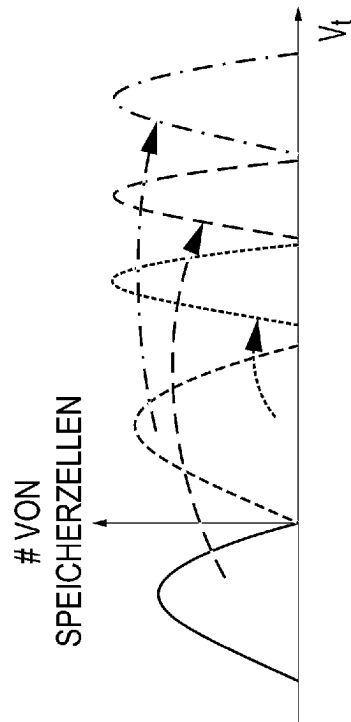


FIG. 14C

