



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 601 29 710 T2 2008.06.05**

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 312 095 B1**

(51) Int Cl.⁸: **G11C 16/00 (2006.01)**

(21) Deutsches Aktenzeichen: **601 29 710.5**

(86) PCT-Aktenzeichen: **PCT/US01/25678**

(96) Europäisches Aktenzeichen: **01 962 213.3**

(87) PCT-Veröffentlichungs-Nr.: **WO 2002/017330**

(86) PCT-Anmeldetag: **16.08.2001**

(87) Veröffentlichungstag
der PCT-Anmeldung: **28.02.2002**

(97) Erstveröffentlichung durch das EPA: **21.05.2003**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **01.08.2007**

(47) Veröffentlichungstag im Patentblatt: **05.06.2008**

(30) Unionspriorität:
643151 21.08.2000 US

(84) Benannte Vertragsstaaten:
**AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT,
LI, LU, MC, NL, PT, SE, TR**

(73) Patentinhaber:
SanDisk Corp., Milpitas, Calif., US

(72) Erfinder:
**CONLEY, Kevin M., San Jose, CA 95120, US;
GUTERMAN, Daniel C., Fremont, CA 94539, US;
GONZALEZ, Carlos J., Los Gatos, CA 95032, US**

(74) Vertreter:
**Patentanwälte Maxton Langmaack & Partner,
50968 Köln**

(54) Bezeichnung: **ANORDNUNG UND STRUKTUR FÜR ZUVERLÄSSIGE DATEIKOPIEOPERATION FÜR
NICHT-FLÜCHTIGE SPEICHER**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

Technischer Bereich

[0001] Diese Erfindung gehört zum Bereich nicht-flüchtiger Halbleiterspeicherarchitekturen und deren Arbeitsweise und hat Anwendung auf elektrisch löschbare, programmierbare Nur-Lese Flash Speicher (EEPROMs).

Hintergrund

[0002] Eine übliche Anwendung von Flash EEPROM Einheiten ist ein Massenspeicher-Subsystem für elektronische Geräte. Derartige Subsysteme werden häufig entweder in herausnehmbaren Speicherkarten implementiert, die in mehreren Hostsysteme eingesetzt werden können, oder als nicht-herausnehmbare eingebettete Speicher im Hostsystem. In beiden Implementierungen enthält das Subsystem eine oder mehrere Flash Einheiten und häufig einen Subsystem-Controller.

[0003] EEPROM Flash Einheiten bestehen aus einem oder mehreren Arrays von Transistorzellen, wobei jede Zelle dazu in der Lage ist, ein oder mehrere Bit Daten nicht-flüchtig zu speichern. Daher benötigen Flash Speicher keine Spannung, um die darin programmierten Daten beizubehalten. Sobald eine Zelle einmal programmiert ist, muss diese wie auch immer gelöscht werden, bevor diese mit neuen Datenwerten wieder programmiert werden kann. Diese Zellenarrays sind in Gruppen partitioniert, um effiziente Implementierungen von Lese-, Programmier- und LösCHFunktionen zur Verfügung zu stellen. Eine typische Flash Speicherarchitektur zur Massenspeicherung ordnet große Gruppen von Zellen in löschraren Blöcken an. Jeder Block ist ferner in einen oder mehrere adressierbare Sektoren partitioniert, welche die Basiseinheit für Lese- und Programmierfunktionen sind.

[0004] Der Subsystemcontroller führt eine Anzahl von Funktionen durch, zu denen die Übersetzung der logischen Blockadressen (LBA) des Subsystems in eine physikalische Chip, Block und Sektoradresse gehört. Der Controller steuert auch den Betrieb der Low-Level Flash Schaltkreise durch eine Reihe von Kommandos, die der Flash Speichereinheit über einen Interfacebus übermittelt werden. Eine weitere Funktion, die der Controller durchführt, ist die Bewahrung der Integrität der im Subsystem gespeicherten Daten mit verschiedenen Mitteln (z.B. durch Verwendung eines Fehlerkorrekturcodes oder ECC).

[0005] Beim Betrieb derartiger Systeme, wobei die gelöschten Blöcke mehrere Sektoren enthalten, kann es gelegentlich erforderlich werden, die Inhalte von einem Sektor von Daten in einen anderen gelöschten Sektor zu kopieren. Dies kann zum Beispiel auftreten, wenn die Daten in einem Teil eines Sektors eines Blockes durch neue Daten ersetzt werden, die vom Host in das Subsystem geschrieben werden, aber die physikalische Nähe der neuen Daten der ursprünglichen Sektoren die anderen unbeeinflussten Daten im Block beibehalten werden sollen. [Fig. 1](#) zeigt ein Beispiel, wie dies durchgeführt werden kann. Die unbeeinflussten Daten der Sektoren werden aus dem ursprünglichen Block gelesen und werden anschließend in gelöschte Sektoren in einen anderen Block kopiert. Die neuen Daten vom Host werden gleichermaßen in gelöschte Sektoren im neuen Block kopiert. Nach Beendigung dieser Operationen wird der ursprüngliche Block mit ersetzten Daten gelöscht.

[0006] Während eines derartigen Betriebes wie oben beschrieben, ist es notwendig, wenn konventionelle Flash Speicher verwendet werden, die Daten, die vom Speicher gelesen werden, zu transferieren und diese danach in den Speicher, der programmiert wird, zu transferieren. Diese Datentransfervorgänge verursachen eine Wartezeit, die der Größe des Sektors geteilt durch die Flash Speicherinterfacebusweite multipliziert mit der Taktzeit des Busses entspricht. Sehr häufig werden diese Vorgänge in der gleichen physikalischen Speichereinheit durchgeführt.

[0007] Um ferner die Auswirkungen, die diese Datentransferwartezeiten auf den Gesamtbetrieb haben, zu erläutern, ist [Fig. 1](#) hier detaillierter beschrieben. Zur Veranschaulichung wird angenommen, dass ein Block aus sechzehn Sektoren besteht, wobei jeder inkrementell zugewiesene Sektoradressen von 0 bis 15 hat, wie in [Fig. 1](#) gezeigt. Für das Beispiel, bei dem erwünscht ist, neue Daten in Sektor 7 bis Sektor 9 des Datenblocks neu zu schreiben, ist die Schreibsequenz wie folgt:

1. Weise einen ungenutzten, gelöschten Block (Löschblock 1) für die Sektor-Programmierung zu,
2. Selektiere Adresse des ursprünglichen Blocks, lese Sektor 0 des ursprünglichen Blocks, transferiere Daten vom Flash EEPROM zum Controller,
3. Selektiere Adresse des neu zugewiesenen Blocks, transferiere Daten zurück zum Flash EEPROM und programmiere in Sektor 0 des neu zugewiesenen Blocks,

4. Wiederhole Schritte 2 und 3 für Sektoren 1 bis 6,
5. Selektiere Adresse des neu zugewiesenen Blocks, transferiere neue Host-Daten für Seite 7 zum Flash EEPROM und programmiere in Sektoren des neu zugewiesenen Blocks,
6. Wiederhole Schritt 5 für Sektoren 8 und 9,
7. Selektiere Adresse des ursprünglichen Blocks, lese Sektor 10 des ursprünglichen Blocks, transferiere Daten vom Flash EEPROM zum Controller,
8. Selektiere Adresse des neu zugewiesenen Blocks, transferiere Daten zurück zum Flash EEPROM und programmiere in Sektor 10 des neu zugewiesenen Blocks,
9. Wiederhole Schritte 7 und 8 für Sektoren 11 bis 15 und
10. Lösche den ursprünglichen (nun ersetzten) Block für zukünftige Schreiboperationen.

[0008] **Fig. 2** zeigt eine typische interne Architektur 4000 einer Flash EEPROM Einheit gemäß dem Stand der Technik. Die Hauptmerkmale beinhalten einen E/A Bus **411** und Kontrollsignale **412** zum Anschluss an einem externen Controller, ein Speichersteuerschaltkreis **450**, um die internen Speicheroperationen mit Registern für Kommando, Adresse und Status zu steuern, ein oder mehrere Arrays **400** von Flash EEPROM Zellen, wobei jedes Array seinen eigenen Reihendecoder (XDEC) **401** und Spaltencontroller (Y-DEC) **402**, eine Gruppe von Leseverstärkern und Programmierkontrollschaltkreisen (SA/PROG) und ein Datenregister **404** hat.

[0009] Falls erwünscht wird eine Vielzahl von Arrays **400** zusammen mit zugehörigen X Decodern, Y Decodern, Programmier/Prüf-Schaltkreisen, Datenregistern und desgleichen zur Verfügung gestellt, was zum Beispiel in US-Patent 5,890,192 gelehrt wird, erteilt am 30.03.1999 und übertragen auf Sandisk Corporation, die Anmelderin dieser Anmeldung.

[0010] Der externe E/A Interfacebus **411** und Kontrollsignale **412** könnten konfiguriert sein wie folgt:

CS – Chip Select	zum Aktivieren des Flash Speicherinterfaces,
RS – Read Strobe	zum Anzeigen, dass AD Bus für eine Daten-Leseoperationen verwendet wird,
WS – Write Strobe	zum Anzeigen, dass der AD Bus für eine Daten-Schreiboperationen verwendet wird,
AS – Address Strobe	zeigt an, dass der AD Bus zum Transferieren von Adressinformationen verwendet wird,
AD[7:0] – Address/Data Bus	zum Transferieren von Daten zwischen Controller und Flash Speicherkommandos, Adressen und Datenregistern.

[0011] Dieses Interface ist nur als Beispiel aufgeführt, da andere Signalkonfigurationen verwendet werden können, um die gleiche Funktionalität zu ermöglichen. Obwohl dieses Diagramm nur ein Flash Speicherarray **400** mit seinen zugehörigen Komponenten zeigt, versteht es sich, dass eine Vielzahl von Arrays in einem einzelnen Flash Speicherchip vorhanden sein können, die ein gemeinsames Interface und Speicherkontrollschaltkreise zusammen verwenden, aber separate XDEC, YDEC, SA/PROG and DATA REG Schaltungen haben, um parallele Lese- und Programmieroperationen zu ermöglichen.

[0012] Daten vom EEPROM System 4000 Datenregister **404** zu einem externen Controller über die an den E/A Bus AD[7:0] **411** koppelnden Datenregister. Das Datenregister **404** ist auch mit dem Leseverstärker/Programmierschaltung **454** gekoppelt. Die Anzahl der Elemente des Datenregisters, das mit jedem Leseverstärker/Programmierschaltungselement gekoppelt ist, kann von der Anzahl von gespeicherten Bit in jeder Flash EEPROM Zelle abhängen. Jede Flash EEPROM Zelle kann eine Vielzahl von Bit, wie zum Beispiel 2 oder 4 haben, falls Mehrzustands-Speicherzellen verwendet werden.

[0013] Reihendecoder **401** decodiert Reihenadressen für Array **400**, um den physikalischen Sektor zu selektieren, auf den zugegriffen wird. Reihendecoder **401** empfängt Reihenadressen über interne Reihenadressleitungen **419** von Speicherkontrolllogik **450**. Spaltendecoder **402** empfängt Spaltenadressen über interne Spaltenadressleitungen **429** von Speicherkontrolllogik **450**.

[0014] Es ist üblich, dass an jeden Sektorwert von Hostdaten, der in einen Sektor programmiert wird, ein Fehlererkennungs- und Korrekturcode (ECC) angehängt wird, der dazu verwendet werden könnte, die Gültigkeit der gespeicherten Daten zu erkennen. Einige derartiger Systeme würden das Ereignis des Transfers als Ge-

legenheit verwenden, um die Gültigkeit der Daten, die gelesen werden, zu überprüfen, um sicherzustellen, dass die Daten korrekt in den neuen Ort programmiert worden sind. Aufgrund der Gesamtgenauigkeit dieser Speichereinheiten ist das Auftreten von Fehlern während derartiger Prüfungen selten.

[0015] **Fig. 3** zeigt eine typische Flash Kartenarchitektur, die einen einzelnen Controller **301** hat, der Host und Speicherkontrollfunktionen durchführt, und ein Flash Speicherarray, das aus einer oder mehreren Speichereinheiten besteht. Der Systemcontroller und der Flash Speicher sind mit Bus **302** verbunden, der dem Controller **301** ermöglicht, Kommandos zu lesen, Adressen und Daten zum und vom Flash Speicherarray zu transferieren.

[0016] **Fig. 4** zeigt ein Timingdiagramm einer Seitenkopieroperation von einer Quelladresse (SRC[N]) zu einer Zieladresse (DST[N]). Das READ Signal zeigt an, dass ein Lesen vom Quellsektor stattfindet. Das XFER Signal zeigt einen Datentransfer zwischen dem Flash Datenregister und dem Controller an. Das R/WB Signal zeigt die Richtung des Datentransfers an (high ist ein Lesen vom Flash zum Controller und low ist ein Schreiben vom Controller zum Flash). Das PROG Signal zeigt an, dass eine Programmieroperation auf der Zielseite stattfindet.

Typische Timingwerte:

$T_{RD} = 25 \text{ us}$ $T_X = 26,4 \text{ us}$ $T_{PRG} = 300 \text{ us}$ $T_{ERA} = 500\text{--}3000 \text{ us}$ (nicht gezeigt), so dass die Gesamtdauer zum Kopieren eines einzelnen Sektors (wie in **Fig. 4** gezeigt)

$$T_{COPY} = T_{RD} + 2 \cdot T_X + T_{PRG} = 377 \text{ us}$$

beträgt. Dies macht die Transferzeit zu 14% der gesamten Seitenkopieroperation aus.

[0017] Es ist eine generelle Entwicklung in nicht-flüchtigen Speicherdesigns, die Anzahl der Zellen, die zu einer Zeit programmiert und gelesen werden können, zu erhöhen, um die Schreib- und Leseperformance dieser Einheiten zu erhöhen. Dies kann durch Erhöhen der Anzahl der Speicherzellen in einem einzelnen Die, Erhöhen der Seitengröße innerhalb einer einzelnen Ebene, paralleles Programmieren mehrerer Chips oder eine Kombination dieser drei Techniken erreicht werden. Das Resultat von einigen dieser besteht darin, dass die Datentransferlänge für die größere Anzahl von Zellen, die parallel programmiert oder mit dem Programm gelesen werden, erhöht werden, und Lesezeiten etwas oder gar nicht verändert werden. Dadurch ist der Aufwand von Kopieren von Daten, die eine derart erhöhte Parallelität haben, derart höher. Als Beispiel zeigen die gleichen Zeitwerte, die im obigen Beispiel verwendet wurden, den Einfluss auf ein System mit einer vierfachen Erhöhung der Parallelität:

$$T_{COPY} = T_{RD} + 8 \cdot T_X + T_{PRG} = 536,2 \text{ us}$$

[0018] Dies macht die Datentransferzeit zu 39% der gesamten Kopieroperation von vier Seiten aus.

[0019] US-Patent 6,040,997 beschreibt eine Technik, die der Patentanmelderin und anderen bekannt ist, bei der ein Flash Speichersystem einen internen Puffer enthält, so dass aus einer ersten Position im Flash Speicher gelesene Daten wieder in eine zweite Position im Flash Speicher geschrieben werden, ohne die Notwendigkeit diese Daten aus dem Flash Speicher zu transferieren. Dies verbessert die Datentransferzeit. Jedoch stellt diese bekannte Technik keine Fehlerprüfung und -korrektur zur Verfügung, sondern geht nur davon aus, dass die aus der ersten Position im Flash Speicher gelesenen Daten fehlerfrei sind.

[0020] Was benötigt wird ist ein Verfahren, bei dem die Effizienz der Kopieroperation ohne Datentransferverzögerung derart verwendet werden kann, dass die Integrität der Daten, die kopiert werden, geprüft werden kann.

ZUSAMMENFASSUNG

[0021] Ein nicht-flüchtiges Speichersystem ist gestaltet wie in Anspruch 1 angegeben. Es enthält Schaltungen, die, nachdem die Daten aus Speicherzellen in einer physikalisch adressierten Position gelesen (abgetastet) wurden, anschließend dazu verwendet werden, um diese Daten in eine andere physikalisch adressierte Position zu programmieren. Dies erhöht die Gesamtleistung des Speichersystems, die unter beiden Datentransfers aus und anschließend zurück in die Speicherschaltung leidet, durch Beseitigen der Wartezeit. Während der Programmieroperation werden die Daten zusätzlich zu einer Kontrollschaltung transferiert, so dass

Datengültigkeit überprüft wird. Dies erhöht weiter die Gesamtleistung des Systems durch Ermöglichen, dass die Daten mit der Programmieroperation gleichzeitig überprüft werden, ohne dass darunter die Systemzuverlässigkeit leidet.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0022] [Fig. 1](#) zeigt die Schritte, die beim Schreiben neuer Daten in eine logische Adresse in einem EEPROM gemäß dem Stand der Technik involviert sind,

[0023] [Fig. 2](#) zeigt ein Blockdiagramm eines typischen EEPROMs gemäß dem Stand der Technik,

[0024] [Fig. 3](#) zeigt eine typische Flash-basierte Speichersubsystemarchitektur gemäß dem Stand der Technik,

[0025] [Fig. 4](#) zeigt ein Timingdiagramm einer Seitenkopieroperation gemäß dem Stand der Technik,

[0026] [Fig. 5](#) zeigt eine schematische Darstellung eines nicht-flüchtigen Speichers gemäß einer Ausgestaltung der vorliegenden Erfindung, und

[0027] [Fig. 6](#) stellt die Sequenz von Ereignissen dar, die während einer Kopieroperation gemäß einer Ausgestaltung der vorliegenden Erfindung stattfindet.

DETAILLIERTE BESCHREIBUNG

[0028] [Fig. 5](#) zeigt ein Diagramm, das einen nicht-flüchtigen Halbleiterspeicher darstellt, der gemäß einer Ausgestaltung der vorliegenden Erfindung ausgestaltet ist. Das gezeigte Speichersystem besteht aus zwei separat adressierten Speicherarrays **400-0** und **400N**, wobei jede zum Adressieren einer bestimmten Wortleitung, Abtasten der Schwellspannungen der Zellen auf dieser Wortleitung und Speichern der abgetasteten Werte zur späteren Verwendung zugehörige Schaltungen hat, so wie zuvor mit Bezug auf [Fig. 2](#) beschrieben wurde. Das Speichersystem hat auch mit jedem Array in Zusammenhang stehende Programmierschaltungen zum Löschen einer Gruppe von Zellen, die von einer oder mehreren Wortleitungen gesteuert werden, und zum Programmieren der Zellen entlang einer von adressierten Wortleitungen. [Fig. 5](#) zeigt ein System mit zwei separaten Arrays mit deren zugehörigen Schaltung und eine separate Controllerschaltung, wobei es sich aber versteht, dass diese Erfindung auch für Speichersysteme mit jeder gewünschten Anzahl von derartigen Arrays von nicht-flüchtigen Speicherzellen gilt und für Systeme, in denen die Controllerschaltung in der gleichen Halbleitereinheit wie eine oder mehrere aus der Vielzahl von Speicherarrays enthalten ist.

[0029] In dieser Erfindung sind die Schaltungen zum Durchführen des Speicherns von abgetasteten Daten und zur Bereitstellung der Programmierdaten derart gestaltet, dass das Speicherelement für beide Operationen physikalisch das gleiche ist. Auf diese Weise, sobald Daten gelesen und im Lesedatenregister gespeichert sind, können diese zum Steuern einer Programmieroperation verwendet werden, ohne die Notwendigkeit von weiterem Datentransfer.

[0030] Der Adressierungsmechanismus ist derart gestaltet, dass nach einer Leseoperation auf einer bestimmten adressierten Wortleitung die Adresse geändert werden kann, um eine andere Wortleitung im gleichen Array zu adressieren zum Programmieren der letzteren Wortleitung mit den Daten aus dem Lesedatenregister. Auf diese Weise enthält die Schaltung alles Notwendige zum Durchführen einer effizienten Kopieroperation von einer Wortleitung auf eine andere Wortleitung ohne die Notwendigkeit, Daten außerhalb der Speichereinheit zu befördern.

[0031] Für das Beispiel, bei dem erwünscht ist, neue Daten in Sektor 7 bis Sektor 9 der 16 Datenblöcke neu zu schreiben, ist die Schreibsequenz wie folgt:

1. Weise ungenutzten Block (gelöschter Block 1) für die Sektorschreiboperation zu.
2. Wähle Adresse des ursprünglichen Blocks, Lese Seite 0 des ursprünglichen Blocks in Datenregister.
3. Wähle Adresse des neu zugewiesenen Blocks, Beginne Programmieroperation im neu zugewiesenen Block. Transferiere gleichzeitig Daten vom Datenregister zur Controllerschaltung, um Datengültigkeit zu prüfen.
4. Prüfe Speicherstatusregister und Status der Datengültigkeit. Bei Fehler rufe Fehlerbehebungsmechanismus auf.
5. Wiederhole Schritte 2 bis 4 für Seiten 1 bis 6.

6. Wähle Adresse des neu zugeordneten Blocks, Transferiere neue Daten für Seite 7 vom Controller zum Flash EEPROM und Beginne Programmieroperation im neu zugeordneten Block.
7. Prüfe Speicherstatusregister. Bei Fehler rufe Fehlerbehebungsmechanismus auf.
8. Wiederhole Schritte 6 und 7 für Seiten 8 und 9.
9. Wähle Adresse des ursprünglichen Blocks, Lese Seite 10 des ursprünglichen Blocks in Datenregister.
10. Wähle Adresse des neu zugewiesenen Blocks, Beginne Programmieroperation von Seite 10 in neu zugewiesenen Block. Transferiere gleichzeitig Daten vom Datenregister zur Controllerschaltung, um Datengültigkeit zu prüfen.
11. Prüfe Speicherstatusregister und Status der Datengültigkeit, und
12. Wiederhole Schritte 9 bis 11 für Seiten 11 bis 15. Bei Fehler rufe Fehlerbehebungsmechanismus auf.

[0032] **Fig. 6** stellt die Sequenz von Ereignissen dar, die während einer Kopieroperation gemäß einer Ausgestaltung der vorliegenden Erfindung auftreten. Die Pfeile zeigen die Richtung des Datenflusses während des Lesens (1), Master- zu Slave-Registertransfer (2), Programmieren (3) und Datentransfer (3). Datentransfer und Programmiervorgänge haben beide eine (3), was andeutet, dass diese parallele Vorgänge sind.

[0033] Da dieser Speicher als ein oder mehrere getrennte Speicherzellenarrays ausgelegt ist, können viele dieser Kopiervorgänge parallel durchgeführt werden, da getrennte Speicherarrays vorhanden sind. Dies erhöht weiter die Leistungsfähigkeit.

[0034] Es tritt gelegentlich auf, dass Daten in einer oder mehreren Zellen, sobald diese abgetastet wurden, nicht die ursprünglichen Daten widerspiegeln, die in diese Zellen programmiert wurden. In derartigen Fällen ist es wünschenswert, durch Verwenden eines externen Schaltungsmittels zum Prüfen die Gültigkeit dieser Daten zu überprüfen. In einer Ausgestaltung wird Datenredundanz verwendet, zum Beispiel ECC, der gleichzeitig mit den ursprünglichen Daten in eine Gruppe von Zellen programmiert wird, der, wenn aus diesen Zellen gelesen wird, die Gültigkeit dieser Daten prüfen kann. In Fällen, bei denen ein Fehler detektiert wird, kann diese Datenredundanz auch dazu verwendet werden, die fehlerhaften Daten zu korrigieren. Die in **Fig. 5** gezeigte Ausgestaltung ermöglicht es, Daten, sobald diese abgetastet wurden, zu einer Schaltung außerhalb der Speicherschaltung zwecks Prüfung zu transferieren. Dies ist auf solche Art und Weise implementiert, um zu ermöglichen, dass diese Transferoperation erfolgt, ohne dass der Inhalt der ursprünglich abgetasteten Daten verändert wird. Dies ermöglicht, dass die Transfer- und Prüfoperation gleichzeitig mit der Programmieroperation erfolgt.

[0035] Da viele getrennte Speicherzellenarrays vorhanden sind und da die Zeitdauer zum Durchführen von Programmiervorgängen signifikant länger als die Zeitdauer zum Durchführen des Transfers und Prüfung ist, können mehrere Programmiervorgänge gleichzeitig mit mehreren Transfer- und Prüfvorgängen durchgeführt werden. Auf diese Art und Weise kann die Gültigkeit der Daten, die während Kopiervorgängen programmiert werden, ohne einen Performancenachteil gewährleistet werden. Und durch Gewährleisten der Gültigkeit der Daten wird die Systemzuverlässigkeit erhöht.

Patentansprüche

1. Nichtflüchtiges Speichersystem, das Folgendes umfasst:
eine Matrix von nichtflüchtigen Speicherzellen (400);
ein erstes Register (454);
ein zweites Register (404); und
eine Steuerungsschaltung (450) zum Lesen von Daten aus der genannten Matrix, Speichern der gelesenen Daten in dem genannten ersten Register (454) und Programmieren von einer oder mehreren Stellen der genannten Matrix mit Daten aus dem genannten ersten Register,
dadurch gekennzeichnet, dass
die genannte Steuerungsschaltung (450) auch die genannten gelesenen Daten in dem genannten zweiten Register (404) speichert und, gleichzeitig mit der Programmierung der genannten ein oder mehreren Stellen der genannten Matrix mit den genannten Daten aus dem ersten Register (454), die Validität der Daten in dem genannten zweiten Register (404) prüft.
2. System nach Anspruch 1, bei dem die genannte Steuerung bewirkt, dass in dem genannten zweiten Register gespeicherte Daten während der genannten Programmierung aus dem genannten Speichersystem gelesen werden.
3. System nach Anspruch 1 oder Anspruch 2, wobei die genannten Speicherzellen Flash-Speicherzellen

umfassen.

4. System nach einem der vorherigen Ansprüche, das ferner eine ECC-Schaltung zum Prüfen der Validität der genannten Daten in dem genannten zweiten Register umfasst.

5. System nach einem der vorherigen Ansprüche, wobei das genannte zweite Register mehrere individuelle Elemente umfasst, die während der genannten Programmierung adressiert und gelesen werden können.

Es folgen 5 Blatt Zeichnungen

Anhängende Zeichnungen

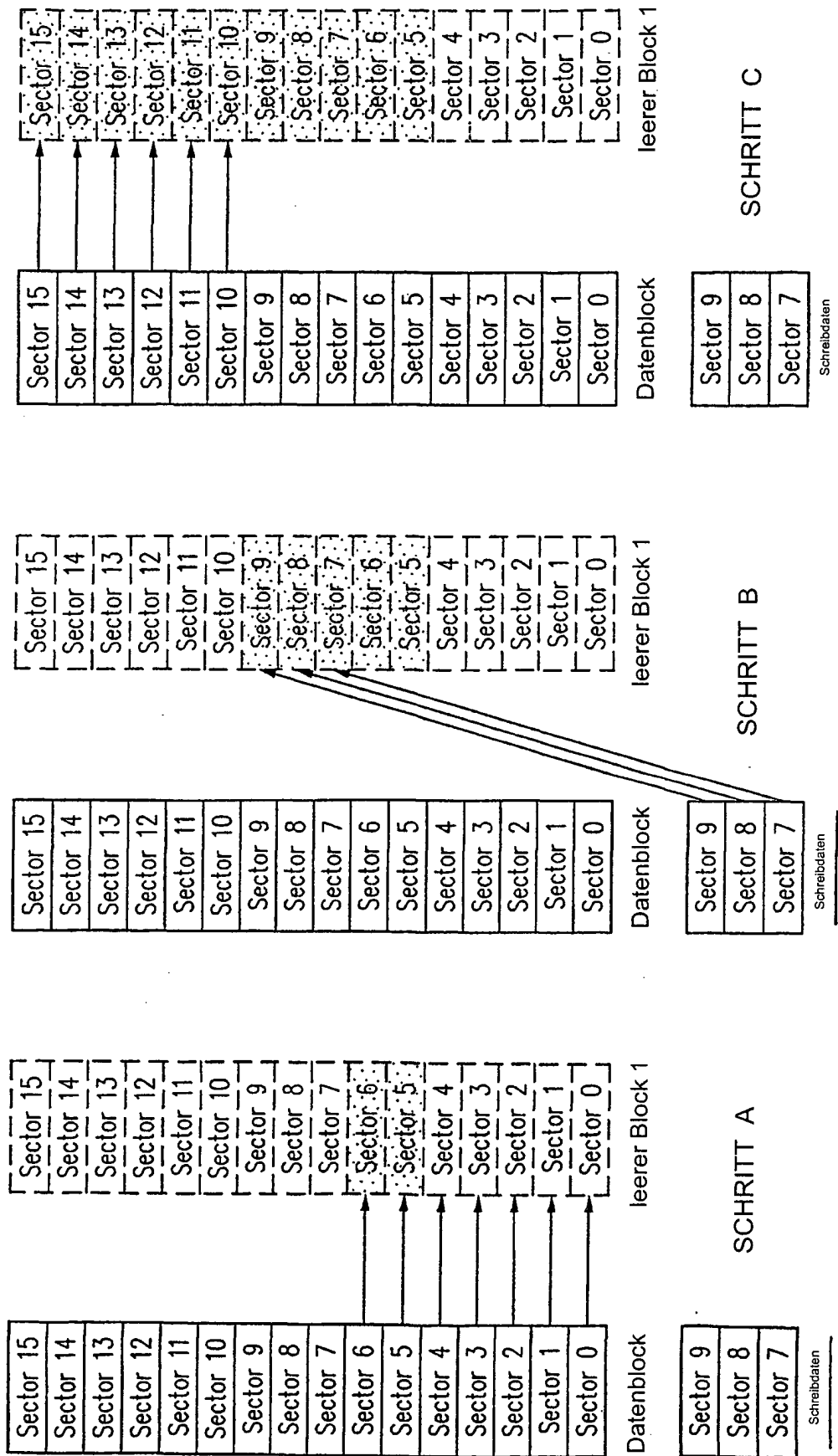


FIG. 1

(Stand der Technik)

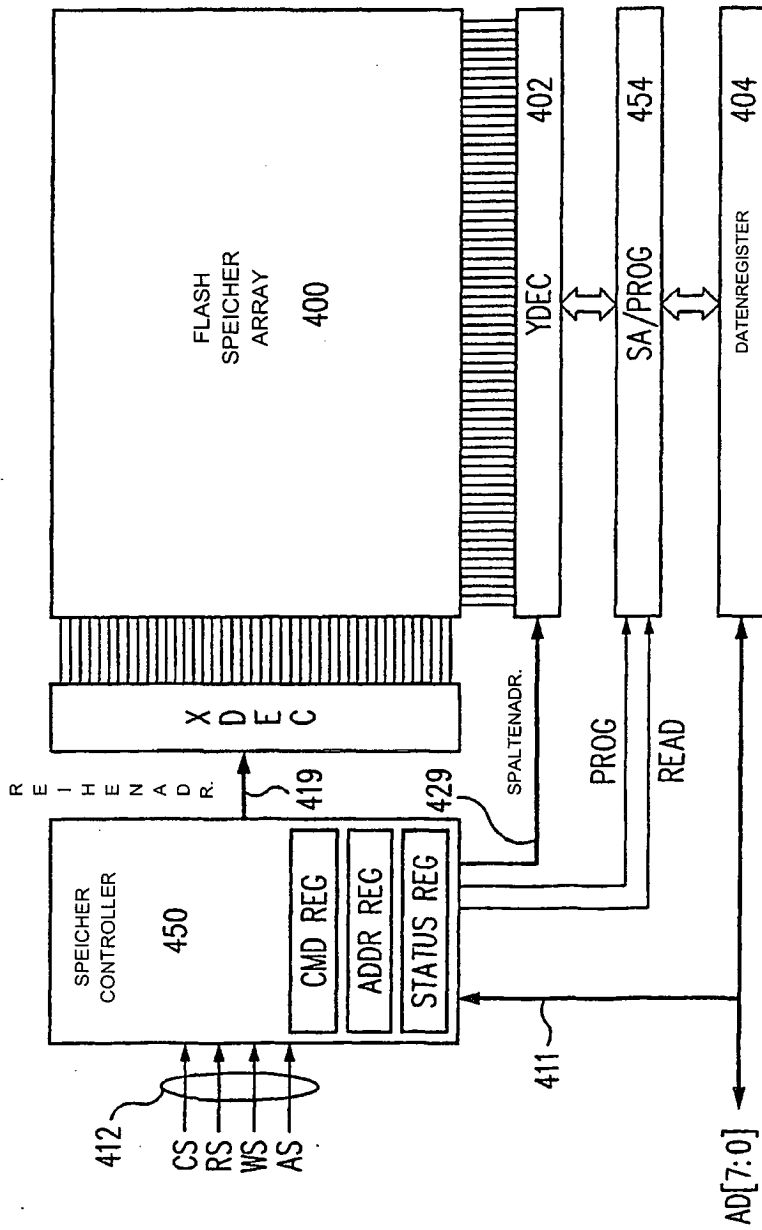


FIG. 2

(Stand der Technik)

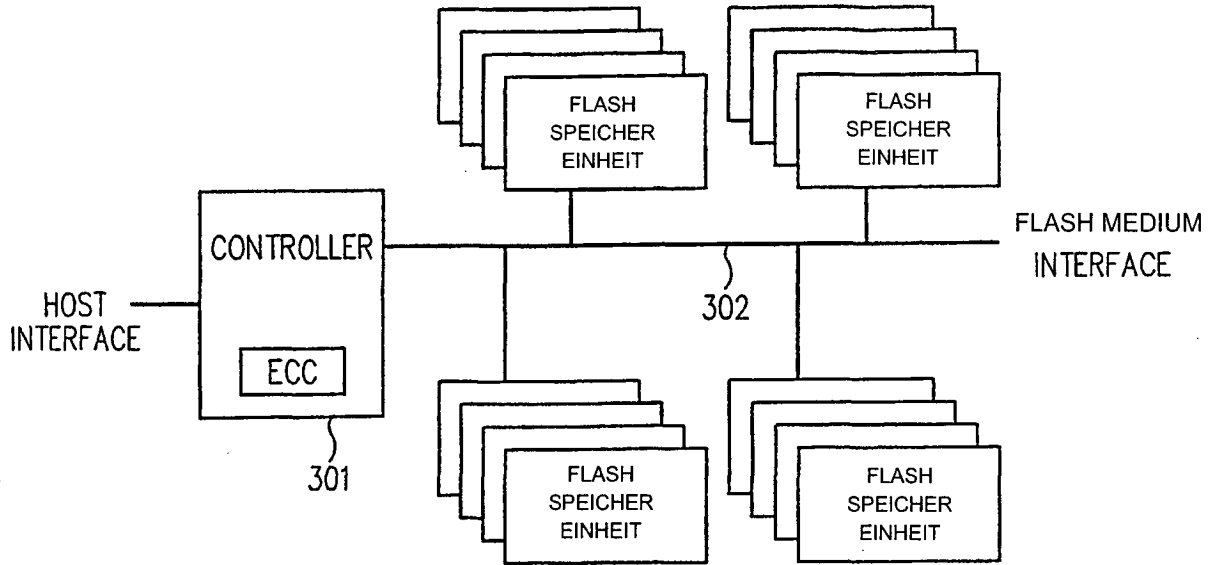


FIG. 3
(Stand der Technik)

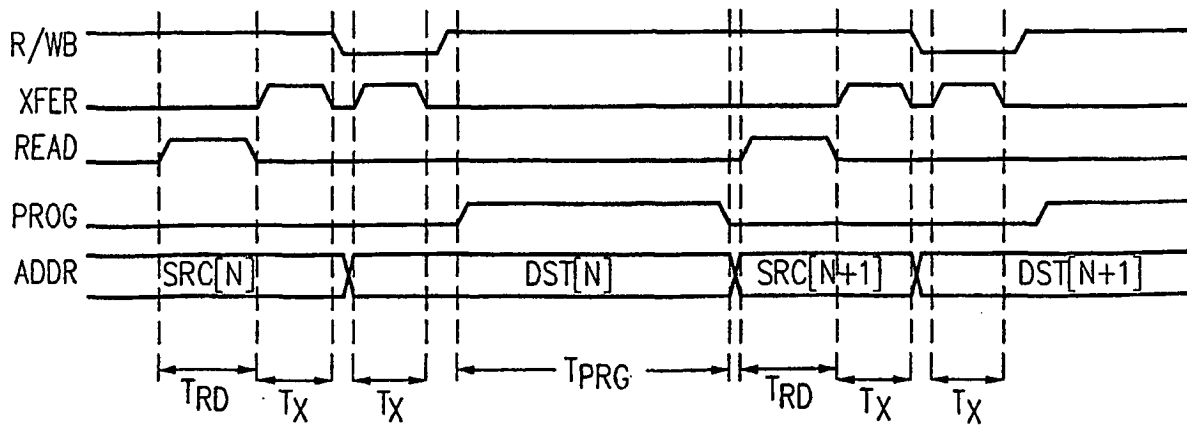


FIG. 4
(Stand der Technik)

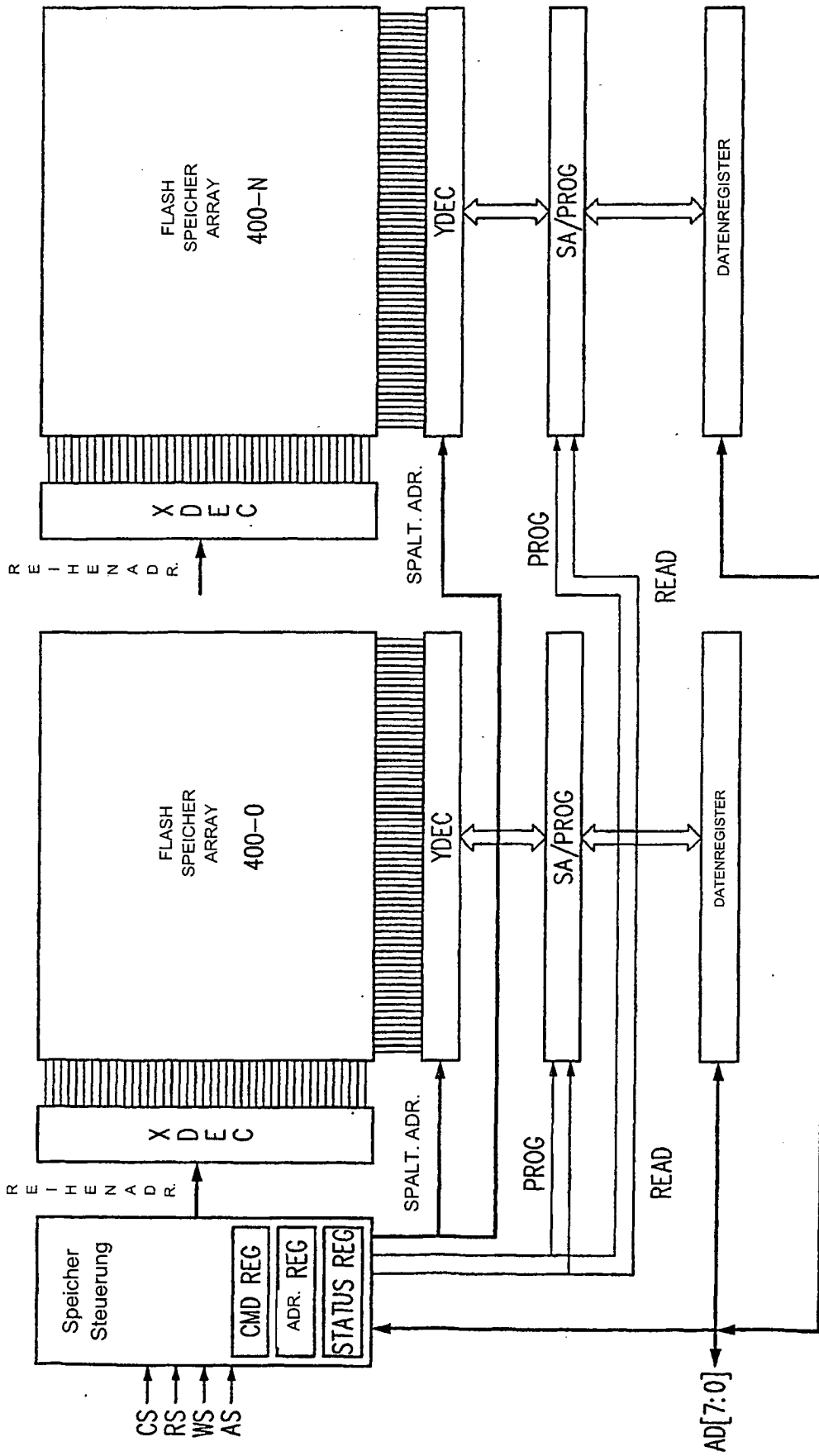


FIG. 5

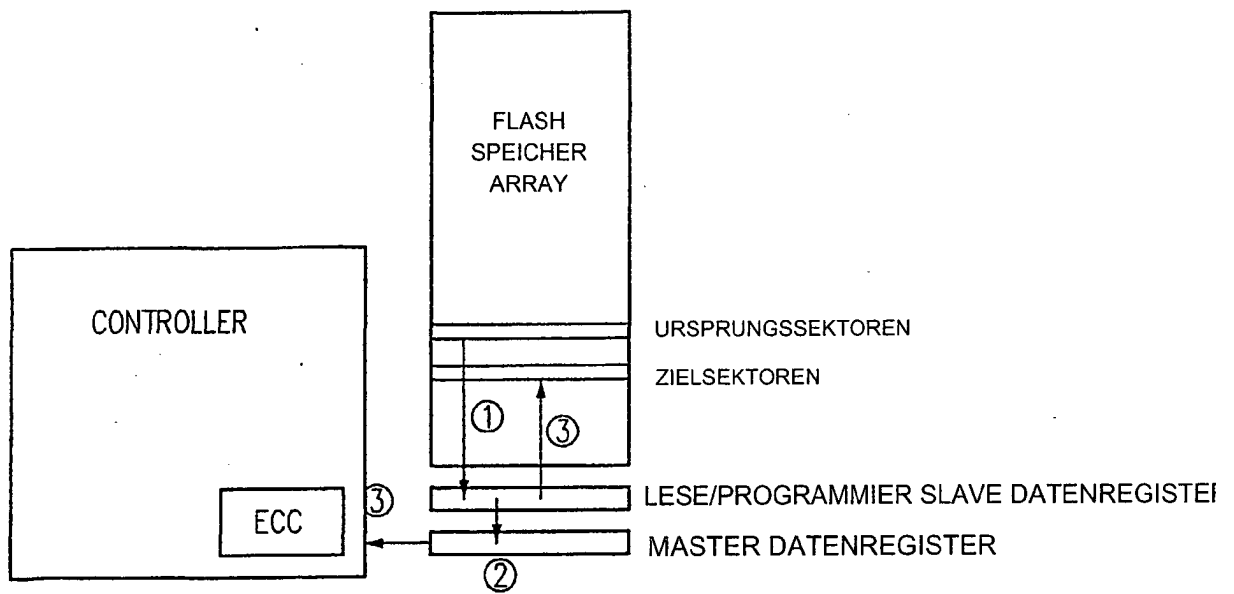


FIG. 6