



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 697 26 304 T2 2004.04.22**

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 0 931 289 B1**

(21) Deutsches Aktenzeichen: **697 26 304.5**

(86) PCT-Aktenzeichen: **PCT/US97/16765**

(96) Europäisches Aktenzeichen: **97 942 628.5**

(87) PCT-Veröffentlichungs-Nr.: **WO 98/012704**

(86) PCT-Anmeldetag: **18.09.1997**

(87) Veröffentlichungstag

der PCT-Anmeldung: **26.03.1998**

(97) Erstveröffentlichung durch das EPA: **28.07.1999**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **19.11.2003**

(47) Veröffentlichungstag im Patentblatt: **22.04.2004**

(51) Int Cl.⁷: **G11C 16/26**
G11C 16/10

(30) Unionspriorität:

718216 20.09.1996 US

(73) Patentinhaber:

Intel Corporation, Santa Clara, Calif., US

(74) Vertreter:

**Patentanwälte Hauck, Graalfs, Wehnert, Döring,
Siemons, 80336 München**

(84) Benannte Vertragsstaaten:

DE, FR, GB, IT

(72) Erfinder:

**LEAK, A., David, Rancho Cordova, US; BEKELE,
G., Fasil, San Francisco, US; PRICE, C., Thomas,
Fair Oaks, US; BROWN, W., Charles, Folsom, US;
HAZEN, K., Peter, Auburn, US; DALVI, Prakash,
Vishram, Fair Oaks, US; ROZMAN, R., Rodney,
Placerville, US; HAID, John, Christopher, Folsom,
US; KREIFELS, Jerry, El Dorado Hills, US; BAKER,
E., Alan, Granite Bay, US**

(54) Bezeichnung: **NICHTFLÜCHTIGER SCHREIBBARER SPEICHER MIT PROGRAMMIERUNGS-AUFHEBUNGSBEFEHL**

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

GEBIET DER ERFINDUNG

[0001] Die vorliegende Erfindung betrifft das Gebiet der Speichervorrichtungen. Im Besonderen betrifft die vorliegende Erfindung das Unterbrechen eines Programmablaufs in einem nichtflüchtigen, beschreibbaren Speicher zur Ausführung anderer Operationen in dem nichtflüchtigen, beschreibbaren Speicher.

STAND DER TECHNIK

[0002] Bei einer Art eines nichtflüchtigen, beschreibbaren Speichers handelt es sich um einen durch ein einfaches Steuersignal löschraren, sofort neu programmierbaren Festwertspeicher ("Flash-EPROM" oder "Flash-Speicher"). Ein typisches Flash-EPROM weist die gleiche Array-Konfiguration wie ein standardmäßiges EPROM (engl. Abkürzung von "Electrically Programmable Read-Only Memory") und kann auf ähnliche Weise wie ein EPROM programmiert werden. Nach der Programmierung kann entweder der gesamte Inhalt des Flash-EPROM oder ein Block des Flash-EPROM durch elektrische Löschung verhältnismäßig schnell gelöscht werden. Eine Löschrspannung wird den Quellen (Source) aller Zellen in dem Flash-EPROM oder in einem Block des Flash-EPROM zur Verfügung gestellt. Dies führt zu einer vollständigen Array-Löschung oder einer Blocklöschung. Das Flash-EPROM oder der gelöschte Block des Flash-EPROM kann danach mit neuen Daten programmiert werden.

[0003] Flash-EPROMs unterscheiden sich von dem Standard der EEPROMs (engl. Abkürzung von "Electrically Erasable Programmable Read-Only Memory") in Bezug auf den Löschrvorgang. Herkömmliche EEPROMs verwenden für gewöhnlich einen ausgesuchten Transistor zur einzelnen Steuerung des Löschrns von Zellen. Flash-EPROMs erreichen im Gegensatz dazu eine deutlich höhere Dichte als einzelne Transistorzellen.

[0004] Bei einem dem Stand der Technik entsprechenden Flash-EPROM mit einem einzelnen Bit bedeutet eine logische "Eins", dass soweit überhaupt, dann nur wenige Elektronen an dem schwebenden Gate gespeichert werden, das einer Bitzelle zugeordnet ist. Eine logische "Null" bedeutet, dass viele Elektronen an dem der Bitzelle zugeordneten schwebenden Gate gespeichert werden. Das Löschrn des Flash-EPROM bedeutet, dass eine logische Eins in jeder Bitzelle gespeichert wird. Jede einzelne Bitzelle des Flash-EPROM kann nicht von einer logischen Null durch eine logische Eins überschrieben werden, ohne dass vorher ein Löschrvorgang vorgenommen wird. Jede einzelne Bitzelle des Flash-EPROM kann jedoch von einer logischen Eins auf eine logische Null überschrieben werden, da dies die einfache Ad-

dition von Elektronen zu einem dem gelöschten Zustand zugeordneten schwebenden Gate umfasst.

[0005] Flash-EPROMs können gelesen, programmiert (oder beschrieben) und gelöscht werden. Bei einem dem Stand der Technik entsprechenden Flash-EPROM dauert eine Programmoperation zum Schreiben eines Datenbytes für gewöhnlich etwa 10 Mikrosekunden. Da jedoch ein gewisser Spielraum erforderlich ist, um die ordnungsgemäße vollständige Ausführung der Programmoperation zu gewährleisten, wird durch den Hersteller des Flash-EPROM eine maximale Programmzeit spezifiziert. Während eine kennzeichnende Programmoperation somit 10 Mikrosekunden dauern kann, muss das System somit unter Umständen über eine maximale Programmoperationszeit von 100 Mikrosekunden warten, um die korrekte Ausführung der Programmoperation zu gewährleisten.

[0006] Bei einem dem Stand der Technik entsprechenden Flash-EPROM dauert ferner ein Löschrvorgang zum Löschrn eines Datenblocks von 8 Kilobyte zwischen 300 und 600 Millisekunden. Das Flash-EPROM kann jedoch eine maximale Löschrvorgangsdauer von bis zu 3 Sekunden benötigen, um zu garantieren, dass der Löschrvorgang des ganzen Datenblocks korrekt ausgeführt worden ist.

[0007] Da der Löschrvorgang eine derart lange Latenzzeit aufweist, weist ein dem Stand der Technik entsprechendes Flash-EPROM einen Löschrvorgangs-Unterbrechungsbeefehl auf. Wenn ein Löschrvorgangs-Unterbrechungsbeefehl in das Flash-EPROM geschrieben wird, unterbricht das Flash-EPROM den gerade ausgeführten Löschrvorgang. Danach können in dem Flash-EPROM andere Operationen ausgeführt werden. Wenn später ein Löschrvorgangs-Wiederaufnahmebeefehl in das Flash-EPROM geschrieben wird, nimmt das Flash-EPROM den Löschrvorgang wieder an der Stelle auf, an welcher der Vorgang vorher durch den Löschrvorgangs-Unterbrechungsbeefehl unterbrochen worden ist. Eine Implementierung der Löschrvorgangs-Unterbrechungsschaltkreisordnung wird in dem U.S. Patent US-A-5.355.464 mit dem Titel "Circuitry And Method For Suspending The Automated Erasure Of A Non-Volatile Semiconductor Memory" an Fandrich et al. beschrieben, das auf den gleichen Anmelder wie die vorliegende Erfindung übertragen worden ist.

[0008] Die Abbildung aus **Fig. 1** zeigt eine Darstellung eines dem Stand der Technik entsprechenden Flash-EPROM **10**. Das Flash-EPROM weist ein Befehlsregister **20**, eine Speicherfeld-Steuerschaltkreisordnung **40** und ein Speicherfeld **50** auf.

[0009] Eine Mehrzahl von Dateneingabe-/Ausgabe-Anschlussstiften (E/A) **12** ist von dem Anschlussstiften des Flash-EPROM mit einem Befehlsregister **20** gekoppelt. Die Anzahl der Daten-E/A-Anschlussstifte **12** entspricht für gewöhnlich **8** Anschlussstiften oder **16** Anschlussstiften, wobei die Anzahl der Größe der in dem Flash-EPROM zu speichernden Daten

entspricht. Die Daten-E/A-Anschlussstifte **12** ermöglichen es, dass Befehle in das Befehlsregister **20** geschrieben werden. Bei einem dem Stand der Technik entsprechenden Flash-EPROM weist der Befehlsdecoder eine Schaltkreisanordnung zur Decodierung der folgenden Befehle auf (1) Löschen, (2) Löscherunterbrechung, (3) Löschwiederaufnahme, (4) Programmieren, (5) Lesen und (6) Status lesen. Ein Schreibfreigabe-Anschlussstift **30** (WE#) ist so gekoppelt, dass er eine Eingabe in das Befehlsregister **20** vorsieht.

[0010] Das Befehlsregister **20** ist über die Signalleitungen **78a-n** mit einer Speicherfeld-Steuerschaltkreisanordnung **40** gekoppelt. Die Speicherfeld-Steuerschaltkreisanordnung **40** weist ein Statusregister **42** auf. Die Speicherfeld-Steuerschaltkreisanordnung **40** weist ferner eine Leseschaltkreisanordnung, eine Zeilen- und Spalten-Decodiererschaltkreisanordnung für einen Zugriff auf Zellen und das Vorsehen von Daten an Zellen in dem Speicherfeld **50** sowie eine Schreibzustandsvorrichtung auf, die eine Programmierungs- und Löschschriftkreisanordnung aufweist. Die Speicherfeld-Steuerschaltkreisanordnung **40** sieht die entsprechenden Signale für einen Zugriff auf das Speicherfeld **50** auf, um die durch das Befehlsregister **20** vorgesehenen Befehle auszuführen. Die Speicherfeld-Steuerschaltkreisanordnung **40** empfängt eine Adresseingabe von den Adressanschlussstiften **44** des Flash-EPROM. Ein Befehlsrücksetzsignal **48** ist von der Speicherfeld-Steuerschaltkreisanordnung **40** zu dem Befehlsregister **20** gekoppelt.

[0011] Das Speicherfeld ist so gekoppelt, dass es Daten an einen Ausgabe-Multiplexer **60** vorsieht, um Daten an die Daten-E/A-Anschlussstifte **12** des Flash-EPROM als Reaktion auf einen Lesevorgang vorzusehen. Das Statusregister **42** ist ferner so gekoppelt, dass es Daten an den Ausgabe-Multiplexer **60** vorsieht, um Statusdaten als Reaktion auf einen Statuslesevorgang an die Daten-E/A-Anschlussstifte **12** des Flash-EPROM vorzusehen. Das Statusregister **42** sieht Informationen über den aktuell von dem Flash-EPROM ausgeführten Vorgang vor. Die Speicherfeld-Steuerschaltkreisanordnung **40** steuert den Ausgabe-Multiplexer **60** auf der Basis der diesem über das Befehlsregister **20** vorgesehenen Befehle. Die Speicherfeld-Steuerschaltkreisanordnung **40** wählt die Statusregisterausgabe so aus, dass diese als Reaktion auf einen Statuslesevorgang durch den Ausgabe-Multiplexer **60** verläuft, und wobei die Speicherfeld-Steuerschaltkreisanordnung als Reaktion auf einen Lesevorgang den Verlauf der Speicherfeldausgabe durch den Ausgabe-Multiplexer **60** auswählt.

[0012] Bei einem dem Stand der Technik entsprechenden Flash-EPROM sieht ein Anschlussstift **62** Ready/Busy (RY/BY#) des Flash-EPROM eine Statusanzeige vor, die anzeigt, ob das Flash-EPROM gerade arbeitet bzw. belegt ist oder nicht. Ein "niedriger" Zustand des RY/BY#-Anschlussstifts zeigt einen

arbeitenden Zustand an, der bedeutet, dass das Flash-EPROM eine Blocklöschvorgang oder einen Byteschreibvorgang ausführt. Ein "hoher" Zustand des RY/BY#-Anschlussstifts zeigt einen Bereitschaftszustand an, wobei dies bedeutet, dass das Flash-EPROM für neue Befehle bereit ist, dass ein Blocklöschvorgang unterbrochen ist oder dass sich die Vorrichtung in einem Abschaltmodus befindet. Das Statusregister **42** ist so gekoppelt, dass es eine Ausgabe an den RY/BY#-Anschlussstift **62** vorsieht. [0013] Zusätzlich werden an das Flash-EPROM eine Speisespannung Vcc, ein Erdpotenzial Vss und eine Programmierspannung Vpp vorgesehen.

[0014] Die Abbildung aus **Fig. 2** zeigt ein dem Stand der Technik entsprechendes Blockdiagramm des Befehlsregisters **20** und der Speicherfeld-Steuerschaltkreisanordnung **40**. Das Befehlsregister **20** weist einen Befehlsdecoder **70** und die Befehlssignalspeicher **76a-n** auf. Die Befehlssignalspeicher weisen einen Löschsichalspeicher **76a**, einen Löscherunterbrechungs-Sichalspeicher **76b**, einen Löscherunterbrechungs-Wiederaufnahmesichalspeicher **76c**, einen Programmsichalspeicher **76d**, einen Lesesichalspeicher **76m** und einen Statuslesesichalspeicher **76n** auf.

[0015] Der Befehlsdecoder decodiert die von den Daten-E/A-Anschlussstiften **12** empfangenen Befehle. Jeder der Befehle wird über die Signalleitungen **72a-n** an einen zugeordneten Befehlssichalspeicher **76a-n** vorgesehen. Die Befehlssichalspeicher **76a-n** speichern einen Befehl nach der Geltendmachung des Schreibfreigabe-Anschlussstiftes **30** (WE#) zwischen. Die Befehlssichalspeicher **76a-n** sehen den decodierten Befehl über die Signalleitungen **78a-n** an die Speicherfeld-Steuerschaltkreisanordnung **40** vor.

[0016] Die Speicherfeld-Steuerschaltkreisanordnung weist eine Löscherunterbrechungs-schaltkreisanordnung **90**, eine Programmschaltkreisanordnung **94**, eine Leseschaltkreisanordnung **96** und eine Statusleseschaltkreisanordnung **98** auf. Die Löscherunterbrechungs-schaltkreisanordnung **90** weist eine Löscherunterbrechungs-schaltkreisanordnung **92** auf. Die Statusleseschaltkreisanordnung **98** ist mit dem Statusregister **42** gekoppelt.

[0017] Der Löschsichalspeicher **76a**, der Löscherunterbrechungs-sichalspeicher **76b** und der Löscherunterbrechungs-Wiederaufnahmesichalspeicher **76c** sind mit der Löscherunterbrechungs-schaltkreisanordnung **90** gekoppelt. Der Löscherunterbrechungs-sichalspeicher **76b** und der Löscherunterbrechungs-Wiederaufnahmesichalspeicher **76c** sind mit der Löscherunterbrechungs-schaltkreisanordnung **92** in der Löscherunterbrechungs-schaltkreisanordnung **90** gekoppelt.

[0018] Der Programmsichalspeicher **76d** ist mit der Programmschaltkreisanordnung **94** gekoppelt. Der Lesesichalspeicher **76m** ist mit der Leseschaltkreisanordnung **96** gekoppelt, und der Statuslesesichalspeicher **76n** ist mit der Statusleseschaltkreisanordnung **98** gekoppelt.

[0019] Die Speicherfeld-Steuerschaltkreisanord-

nung **40** ist so gekoppelt, dass sie eines oder mehrere Befehlsrücksetzsignale **48** an den Befehlsdecodierer vorsieht, um die Befehlssignalspeicher **76a-n** zu löschen. Der Befehlsdecodierer verwendet die Befehlsrücksetzsignale **48** zum Löschen der Befehlssignalspeicher **76a-n** über Befehlssignalspeicher-Rücksetzsignale **74a-n**. Bei einer Implementierung sind einzelne Befehlssignalspeicher-Rücksetzsignale **74a-n** mit jedem Befehlssignalspeicher **76a-n** gekoppelt. Bei einer anderen Implementierung ist ein Befehlsrücksetzsignal mit allen Befehlssignalspeichern gekoppelt.

[0020] Flash-EPROMs können sowohl zum Speichern von Code als auch Daten verwendet werden. Bei einem Einsatz gemäß dem Stand der Technik wird Code in bestimmten Blöcken des Flash-EPROM gespeichert, während Daten in anderen Blöcken des Flash-EPROM gespeichert werden. Dies ermöglicht das Löschen eines Blocks ohne den Inhalt eines anderen Blocks zu beeinträchtigen. Darüber hinaus sind einige Flash-EPROMs auf Datenblöcke und Codeblöcke verschiedener Größen vorbereitet.

[0021] Zwar ist es möglich, sowohl Code als auch Daten in einem Flash-EPROM zu speichern und den direkt von dem Flash-EPROM an einen Prozessor vorgesehenen Code auszuführen, wobei jedoch ein Problem entsteht, wenn ein Flash-EPROM in einem System eingesetzt wird, das eine Behandlung der Codeerfassungen erfordert. Dies ist durch die lange Latenzzeit für Programmoperationen und Löschvorgänge begründet, wie dies bereits vorstehend im Text beschrieben worden ist. Wenn ein Prozessor zum Beispiel einen Programmvorgang zum Schreiben eines Datenbytes in das Flash-EPROM ausführt und der Prozessor in der Folge anfordert, dass das Flash-EPROM einen Lesevorgang ausführt, um eine Codeerfassung auszuführen, d. h. das Lesen von Code zur Erfassung neuer Anweisungen, die von dem Prozessor auszuführen sind, kann der Lesevorgang um bis zu 100 Mikrosekunden verzögert werden, wobei darauf gewartet wird, dass die Programmoperation abgeschlossen wird. Dies bewirkt ein Anhalten des Prozessors, wobei der Prozessor bis zum Empfang neuer Befehle in dem angehaltenen Zustand verbleibt. Eine derartige Verzögerung beim Lesen von Code wäre in einem System unzulässig, das es voraussetzt, dass Codeerfassungen innerhalb eines kürzeren Zeitraums ausgeführt werden als dem maximalen Zeitraums des Programmablaufs. Die Abbildung aus **Fig. 3** zeigt eine dem Stand der Technik entsprechende Darstellung eines Systems, das einen Prozessor **100**, einen flüchtigen Speicher **102** und ein Flash-EPROM **104** umfasst, die über einen Bus **108** gekoppelt sind. Der flüchtige Speicher **102** und das Flash-EPROM **104** können jedoch auch über separate Busse mit dem Prozessor **100** gekoppelt werden. Das Flash-EPROM weist Code und Daten auf, wobei der Code durch den Prozessor ausgeführt werden kann. Der Code des Flash-EPROM wird in den flüchtigen Speicher übertragen oder kopiert, bei

dem es sich um einen dynamischen Direktzugriffsspeicher (DRAM) oder einen statischen Direktzugriffsspeicher (SRAM) handeln kann. Nach dem der Code in den flüchtigen Speicher übertragen worden ist, wenn das Flash-EPROM eine Programmoperation ausführt und der Prozessor eine Codeerfassungsanforderung erzeugt, kann der Prozessor die Codeerfassungsanforderung durch Lesen des angeforderten Codes aus dem flüchtigen Speicher erfüllen. Der Prozessor wartet nicht darauf, dass das Flash-EPROM dessen Programmablauf abschließt, um eine Codeerfassung auszuführen.

[0022] Dieses System kann jedoch teuer sein, wenn die Größe des in dem Flash-EPROM gespeicherten Codes groß ist, da der DRAM/SRAM ausreichend groß sein müsste, um den gesamten Codeblock zu speichern, um die Latenzzeit des Programmablaufs zu überwinden. Ein Beispiel eines Systems, das die Konfiguration aus der Abbildung aus **Fig. 3** verwenden könnte, ist ein Personalcomputer (PC).

[0023] Die Abbildung aus **Fig. 45** zeigt ein dem Stand der Technik entsprechendes System mit einem Prozessor **100**, einem flüchtigen Speicher **102**, einem Flash-EPROM **104** und einem EEPROM **106**. In diesem dem Stand der Technik entsprechenden System speichert das EEPROM **106** Daten und das Flash-EPROM **104** speichert Code. Der SRAM/DRAM **102** wird zum vorübergehenden Speichern von Daten eingesetzt, bevor die Daten an das EEPROM **106** vorgesehen werden, d. h. der SRAM/DRAM-Speicher dient als Puffer zwischen dem Prozessor und dem EEPROM.

[0024] Die Abbildung aus **Fig. 5** zeigt ein weiteres System gemäß der Stand der Technik, wobei das EEPROM **106** Code speichert und das Flash-EPROM **104** Daten speichert. Der SRAM/DRAM-Speicher **102** dient als temporärer Puffer zwischen dem Prozessor und dem Flash-EPROM.

[0025] Die Abbildungen der **Fig. 3, 4** und **5** zeigen somit dem Stand der Technik entsprechende Flash-EPROM-Systeme, denen die Fähigkeit fehlt, schnell und leicht die Codeerfassungsanforderungen eines Prozessors zu behandeln, während der Flash-Speicher einen Programmablauf ausführt.

[0026] Das U.S. Patent US-A-5.034.922 offenbart eine Einrichtung, die es verhindert, dass programmierte Zellen unvollständig gelesen werden, wenn eine Leseanforderung einen unvollständigen Schreibvorgang unterbrochen hat. In diesem Bezugsdokument wird ein Schreibprozess in einem EEPROM als Reaktion auf ein Lesesteuersignal und eine Leseadresse angehalten. EP-A-0376285 offenbart einen mit einem EEPROM verbundenen Prozessor, wobei der Prozessor eine Datenschreibenanforderung und eine Datenleseanforderung an das EEPROM ausgibt, und wobei eine Steuerschaltung mit dem EEPROM und dem Prozessor gekoppelt ist, wodurch ein Timer in der Steuerschaltung dazu verwendet wird, einen Zeitraum zu takten, um ein Signal für das Operationsende zu erzeugen. Durch eine Daten-

schreibenanforderung werden Daten in einer ausgesuchten Speicherzelle gespeichert, und als Reaktion auf das Beendigungssignal wird der Datenschreibprozess beendet. Offenbart wird eine Einrichtung, die auf eine Datenleseanforderung anspricht, die während einem Datenschreibprozess zum Unterbrechen des Datenschreibprozesses ausgegeben worden ist, um Daten aus einer ausgesuchten Speicherzelle zu lesen und um danach den unterbrochenen Datenschreibprozess wiederaufzunehmen. Ferner werden Einrichtungen offenbart, um einen Zeitzählvorgang des Timers während einem Zeitraum im Verhältnis zu einem Unterbrechungszeitraum des Datenschreibvorgangs zu unterbrechen.

ZUSAMMENFASSUNG DER ERFINDUNG

[0027] Der vorliegenden Erfindung liegt die Aufgabe zugrunde, die Fähigkeit zur Unterbrechung einer Programmoperation in einem nichtflüchtigen, beschreibbaren Speicher vorzusehen, um Code aus dem nichtflüchtigen, beschreibbaren Speicher zu lesen. Vorzugsweise verwendet die vorliegende Erfindung einen Unterbrechungsbefehl zur Unterbrechung einer Programmoperation als Reaktion auf ein Unterbrechungssignal.

[0028] Vorgesehen ist gemäß einem ersten Aspekt der vorliegenden Erfindung ein Verfahren gemäß dem gegenständlichen Anspruch 1 oder 5.

[0029] Vorgesehen ist gemäß einem zweiten Aspekt der vorliegenden Erfindung ein System gemäß dem gegenständlichen Anspruch 3.

[0030] Beschrieben werden ein Verfahren und eine Vorrichtung zur Unterbrechung von Programmoperationen in einem Flash-Speicher. Der Flash-Speicher weist ein Speicherfeld, ein Befehlsregister und eine Speicherfeld-Steuerschaltkreisanordnung auf. Das Befehlsregister decodiert einen Programmunterbrechungsbefehl und sieht als Ausgabe ein Unterbrechungssignal vor. Die Speicherfeld-Steuerschaltkreisanordnung ist so gekoppelt, dass sie das Unterbrechungssignal von dem Befehlsregister empfängt. Die Speicherfeld-Steuerschaltkreisanordnung führt eine Programmoperation aus, wobei Daten in das Speicherfeld geschrieben werden. Die Speicherfeld-Steuerschaltkreisanordnung unterbricht die Programmoperation beim Empfang des Unterbrechungssignals.

[0031] Weitere Aufgaben, Merkmale und Vorteile der vorliegenden Erfindung werden aus den beigelegten Zeichnungen und der nachstehend folgenden genauen Beschreibung deutlich.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0032] Es zeigen:

[0033] **Fig. 1** eine dem Stand der Technik entsprechende Darstellung eines Flash-EPROM;

[0034] **Fig. 2** ein dem Stand der Technik entsprechendes Blockdiagramm des Befehlsregisters und

der Speicherfeld-Steuerschaltkreisanordnung des Flash-EPROM;

[0035] **Fig. 3** eine dem Stand der Technik entsprechende Darstellung eines Systems, das ein Flash-EPROM, einen Prozessor und einen flüchtigen Speicher umfasst, die über einen Bus miteinander verbunden sind;

[0036] **Fig. 4** ein dem Stand der Technik entsprechendes System, das ein EEPROM zum Speichern von Daten verwendet;

[0037] **Fig. 5** ein weiteres dem Stand der Technik entsprechendes System, das ein EEPROM zum Speichern von Code verwendet;

[0038] **Fig. 6** ein Blockdiagramm des Befehlsregisters und der Speicherfeld-Steuerschaltkreisanordnung im Einsatz in einem Ausführungsbeispiel der vorliegenden Erfindung;

[0039] **Fig. 7** ein Blockdiagramm eines weiteren Ausführungsbeispiels des Befehlsregisters und der Speicherfeld-Steuerschaltkreisanordnung;

[0040] **Fig. 8a** ein Blockdiagramm eines Systems, das einen Prozessor, einen SRAM/DRAM-Speicher und ein Flash-EPROM aufweist, die über einen Bus miteinander verbunden sind;

[0041] **Fig. 8b** den Inhalt des SRRM/DRAM-Speichers aus **Fig. 8a**;

[0042] **Fig. 8c** den Inhalt des Flash-EPROM aus **Fig. 8a**;

[0043] **Fig. 9** ein Flussdiagramm mit beispielhaften Schritten eines Systems mit der Konfiguration gemäß den Abbildungen aus den **Fig. 8a, 8b und 8c**; und

[0044] **Fig. 10** ein Flussdiagramm, das veranschaulicht, dass unter Verwendung einer Verschachtelungstechnik mehrere Operationen unterbrochen werden können.

GENAUE BESCHREIBUNG

[0045] Beschrieben werden ein Verfahren und eine Vorrichtung zur Unterbrechung einer Programmoperation bzw. eines Programmablaufs in einem nichtflüchtigen, beschreibbaren Speicher zum Lesen von Code aus dem nichtflüchtigen, beschreibbaren Speicher. In der genauen Beschreibung werden zwar Ausführungsbeispiele beschrieben, bei denen ein Flash-EPROM verwendet wird, jedoch kann die vorliegende Erfindung auch in Verbindung mit jedem anderen nichtflüchtigen, beschreibbaren Speicher verwendet werden, wozu unter anderem die folgenden Speicher zählen: EPROMs, EEPROMs und Flash-Speicher, einschließlich Technologien wie NOR, NAND, AND, Divided bit-line NOR (DINOR) und ferroelektrischer Direktzugriffsspeicher (FRAM).

[0046] Die Abbildung aus **Fig. 6** zeigt ein Blockdiagramm des Befehlsregisters **120** und der Speicherfeld-Steuerschaltkreisanordnung **140** gemäß einem Ausführungsbeispiel der vorliegenden Erfindung.

[0047] Der Befehlsdecoder **170** decodiert die folgenden Befehle: (1) Löschen, (2) Löschenunterbrechung, (3) Löschiwiederaufnahme, (4) Programmie-

ren, (5) Programmunterbrechung, (6) Programmwiederaufnahme, (7) Lesen und (8) Status lesen. Der Befehlsdecoder sieht den decodierten Befehl über Signalleitungen **172a-n** an einen entsprechenden Befehlssignalspeicher **176a-n** vor. Die Befehlssignalspeicher werden unter Verwendung des Schreibfreigabe-Anschlussstifts **130 (WE#)** gesperrt.

[0048] Der Löschsichalspeicher **176a** ist über die Signalleitung **178a** mit der Löschsichaltschaltkreisanordnung **190** gekoppelt. Der Löschkvorgangs-Unterbrechungssichalspeicher **176b** ist über die Signalleitung **178b** mit der Löschkvorgangs-Unterbrechungssichaltschaltkreisanordnung **192** der Löschsichaltschaltkreisanordnung **190** gekoppelt.

[0049] Der Programmsichalspeicher **176d** ist über die Signalleitung **178d** mit der Programmsichaltschaltkreisanordnung **194** gekoppelt. Der Programmunterbrechungs-Sichalspeicher **176e** ist über die Signalleitung **178e** mit der Programmunterbrechungs-Sichaltschaltkreisanordnung **195** der Programmsichaltschaltkreisanordnung **194** gekoppelt.

[0050] Der Lesesichalspeicher **176m** ist über die Signalleitung **178m** mit der Lesesichaltschaltkreisanordnung **196** gekoppelt, und der Statuslese-Sichalspeicher **176n** ist über die Signalleitung **178n** mit der Statuslese-Sichaltschaltkreisanordnung **198** gekoppelt. Die Statuslese-Sichaltschaltkreisanordnung **198** ist mit dem Statusregister **142** gekoppelt, das so gekoppelt ist, dass es eine Statusausgabe an die Daten-E/A- und RY/BY#-Anschlussstifte vorsieht.

[0051] Die Speicherfeld-Steuerschaltkreisanordnung **140** ist so gekoppelt, dass sie ein oder mehrere Befehlsrücksetzsichalsignale **148** an den Befehlsdecoder **170** zum Löschen der Befehlssichalspeicher **176a-n** vorsieht. Der Befehlsdecoder verwendet die Befehlsrücksetzsichalsignale **148** zum Löschen der Befehlssichalspeicher **176a-n** über BefehlssichalspeicherRücksetzsichalsignale **174a-n**. In einem Ausführungsbeispiel sind einzelne Befehlssichalspeicher-Rücksetzsichalsignale mit jedem Befehlssichalspeicher **176a-n** gekoppelt, so dass jeder Befehlssichalspeicher **176a-n** einzeln gelöscht werden kann. In einem anderen Ausführungsbeispiel ist ein Befehlssichalspeicher-Rücksetzsichalsignal mit allen Befehlssichalspeichern gekoppelt.

[0052] Wenn ein Programmunterbrechungsbefehl in den Befehlsdecoder geschrieben wird, sieht der Befehlsdecoder einen Programmunterbrechungsbefehl an den Programmunterbrechungs-Sichalspeicher **176e** vor. Wenn ein Programmwiederaufnahmebefehl an den Befehlsdecoder geschrieben wird, löscht der Befehlsdecoder **170** den Programmunterbrechungs-Sichalspeicher **176e** durch Annahme des Befehlssichalspeicher-Rücksetzsichalsignals **174e**.

[0053] In einem Ausführungsbeispiel handelt es sich bei dem Programmunterbrechungsbefehl und dem Programmwiederaufnahmebefehl um den gleichen Befehl, wobei die Befehle abhängig von dem zeitlichen Auftreten derselben unterschieden werden können. Jedes Mal, wenn der Programmunterbre-

chungs-/Programmwiederaufnahmebefehl an den Befehlsdecoder geschrieben wird, wechselt der Befehlsdecoder zwischen dem Vorsehen eines Programmunterbrechungsbefehls **176e** oder dem Löschen des Programmunterbrechungs-Sichalspeichers **176e**.

[0054] In einem anderen Ausführungsbeispiel wird ein einziger Unterbrechungsbefehl sowohl für Programmunterbrechungen als auch für Löschkvorgangsunterbrechungen verwendet. Ein Programmunterbrechungsbefehl und ein Löschkvorgangs-Unterbrechungsbefehl unterscheiden sich voneinander durch ihr zeitliches Auftreten. Wenn ein Löschkvorgang ausgeführt wird, während der Unterbrechungsbefehl an den Befehlsdecoder geschrieben wird, so wird eine Löschkvorgangsunterbrechung ausgeführt. Wenn ein Programmbetrieb ausgeführt wird, wenn der Unterbrechungsbefehl an den Befehlsdecoder geschrieben wird, so wird eine Programmunterbrechung ausgeführt. Ferner können die Befehle zur Programmwiederaufnahme sowie zur Wiederaufnahme des Löschkvorgangs ebenso gleich sein wie die Programmunterbrechungs-/Löschkvorgangsunterbrechungsbefehle. Der Befehlsdecoder **122** speichert die letzte unterbrochene Operation. Wenn der Befehl "Unterbrechungs-Wiederaufnahme" an den Befehlsdecoder geschrieben wird, während eine angehaltene oder nicht-unterbrechungsfähige Operation ausgeführt wird, wird die letzte unterbrochene Operation wiederaufgenommen. Unterbrochene Operationen können verschachtelt sein, wie dies nachstehend in Bezug auf die Abbildung aus **Fig. 10** näher beschrieben wird.

[0055] Die Speicherfeld-Steuerschaltkreisanordnung **140** interpretiert die an die Schaltkreisanordnung vorgesehenen Befehlssichalsignale **178a-n** und führt eine entsprechende Operation als Reaktion auf die Befehlssichalsignale aus. Die Speicherfeld-Steuerschaltkreisanordnung **140** weist eine Programmunterbrechungs-Sichaltschaltkreisanordnung **195** zur Unterbrechung einer Programmoperation des Speicherfelds **150** auf. Die Speicherfeld-Steuerschaltkreisanordnung **140** weist ferner eine dem Stand der Technik entsprechende Löschkvorgangs-Unterbrechungssichaltschaltkreisanordnung **192** zur Unterbrechung eines Löschkvorgangs des Speicherfelds **150** auf. Die Speicherfeld-Steuerschaltkreisanordnung weist eine Einrichtung zum Speichern des Zustands des unterbrochenen Nicht-Lesevorgangs auf, so dass der Nicht-Lesevorgang später wiederaufgenommen werden kann. Zu den Nicht-Leseoperationen zählen in der folgenden Erläuterung Programmoperationen und Löschooperationen. In einem alternativen Ausführungsbeispiel können jedoch auch andere Arten von Operationen unterbrochen werden, wie etwa Befehlsoperationen und Statusabrufoperationen.

[0056] Eine Programmunterbrechungsoperation beendet ihre Unterbrechung einer Programmoperation innerhalb eines vorbestimmten Zeitraums, so dass ein Lesevorgang mit einer spezifizierten Latenzzeit

ausgeführt werden kann. Der Programmunterbrechungsvorgang wird durch das Schreiben eines Programmunterbrechungsbefehls an den Befehlsdecodierer **170** eingeleitet. In einem Ausführungsbeispiel ist die Programmunterbrechung innerhalb von 7 Mikrosekunden abgeschlossen. Nach diesem vorbestimmten Zeitraum können andere Operationen an dem Flash-EPROM ausgeführt werden.

[0057] In ähnlicher Weise wird die Löschvorgangs-Unterbrechungsoperation innerhalb eines vorbestimmten Zeitraums abgeschlossen. Die Löschvorgangs-Unterbrechungsoperation kann durch das dem Stand der Technik entsprechende Verfahren des Schreibens eines Löschvorgangs-Unterbrechungsbefehls an den Befehlsdecodierer **122** eingeleitet werden. In einem Ausführungsbeispiel erfolgt die Unterbrechung des Löschvorgangs vollständig innerhalb von 20 Mikrosekunden. Nach diesem vorbestimmten Zeitraum können andere Operationen ausgeführt werden.

[0058] In einem Ausführungsbeispiel ist es möglich zu bestimmen, ob eine Programmoperation oder eine Löschoption ausgeführt wird, indem auf den Status des Flash-EPROM unter Verwendung des Statuslesebefehls zugegriffen wird. Der Status kann auch durch den RY/BY#-Anschlussstift **162** bestimmt werden, wie dies vorstehend im Text beschrieben worden ist. Alternativ kann ein separater Anschlussstift verwendet werden, um anzuzeigen, ob eine Programmoperation oder ein Löschvorgang ausgeführt wird.

[0059] Die Abbildung aus **Fig. 7** zeigt ein Blockdiagramm eines weiteren Ausführungsbeispiels des Befehlsregisters **120** und der Speicherfeld-Steuer-schaltkreisanordnung **140**. Das Befehlsregister **120** aus der Abbildung aus **Fig. 7** ist dem Register aus **Fig. 6** ähnlich, mit der Ausnahme, dass der Befehlsdecodierer aus **Fig. 7** einen Löschvorgangswiederaufnahme-Signalspeicher **176c** und einen Programmwiederaufnahme-Signalspeicher **176e** aufweist. Der Löschvorgangswiederaufnahme-Signalspeicher **176** entspricht dem Stand der Technik.

[0060] Der Löschvorgangswiederaufnahme-Signalspeicher **176c** empfängt ein decodiertes Befehlssignal von dem Befehlsdecodierer **170** über die Signalleitung **172c**. Der Löschvorgangswiederaufnahme-Signalspeicher **176c** wird bei Geltendmachung des WE#-Anschlussstifts **130** gesperrt. Der Löschvorgangswiederaufnahme-Signalspeicher **176c** ist über die Signalleitung **178c** mit der Löschvorgangs-Unterbrechungs-Schaltkreisanordnung **192** der Löschschtaltkreisanordnung **190** gekoppelt. Ein Befehlssignalspeicher-Rücksetzsignal **174c** wird von dem Befehlsdecodierer **170** an den Löschvorgangswiederaufnahme-Signalspeicher **176c** vorgesehen.

[0061] Der Programmwiederaufnahme-Signalspeicher **176f** empfängt ein decodiertes Befehlssignal von dem Befehlsdecodierer **170** über die Signalleitung **172f**. Der Programmwiederaufnahme-Signalspeicher **176f** wird bei Geltendmachung des

WE#-Anschlussstifts **130** gesperrt. Der Programmwiederaufnahme-Signalspeicher **176f** ist über die Signalleitung **178f** mit der Programmunterbrechungs-Schaltkreisanordnung **195** der Programmschaltkreisanordnung **194** gekoppelt. Ein Befehlssignalspeicher-Rücksetzsignal **174f** wird von dem Befehlsdecodierer **170** an den Programmwiederaufnahme-Signalspeicher **176f** vorgesehen.

[0062] Der Programmunterbrechungs-Signalspeicher **176e** sieht ein Signal an die Programmunterbrechungs-Schaltkreisanordnung **195** vor, um einen Programmablauf zu unterbrechen, und der Programmwiederaufnahme-Signalspeicher **176f** sieht ein Signal an die Programmunterbrechungs-Schaltkreisanordnung **195** vor, um den unterbrochenen Programmablauf wiederaufzunehmen.

[0063] Die Abbildung aus **Fig. 8a** zeigt ein Blockdiagramm eines Systems mit einem Prozessor **400**, einem SRAM/DRAM-Speicher **405** und einem Flash-EPROM **410**, die über einen Bus **108** miteinander gekoppelt sind. Eine Systemunterbrechung **200** ist so gekoppelt, dass sie eine Eingabe in den Prozessor **400** vorsieht. In einem anderen Ausführungsbeispiel sind der SRAM/DRAM-Speicher **405** und das Flash-EPROM **410** über verschiedene Busse mit dem Prozessor **400** gekoppelt. Die Abbildung aus **Fig. 8b** zeigt den Inhalt des SRAM/DRAM-Speichers **405** aus **Fig. 8a**, und die Abbildung aus **Fig. 8c** zeigt den Inhalt des Flash-EPROM **410** aus **Fig. 8a**.

[0064] Der SRAM/DRAM-Speicher **405** weisen einen Datenpuffer-Speicherbereich **420** auf. Der Datenpuffer-Speicherbereich wird zum vorübergehenden Speichern von Daten verwendet, bevor diese in einem Datenbereich **430** des Flash-EPROM **410** gespeichert werden. Der temporäre Puffer ermöglicht es, dass Daten verhältnismäßig schnell in den SRAM/DRAM-Speicher **405** geschrieben und danach zu dem Flash-EPROM **410** geschrieben werden, wenn der Zeitpunkt dafür gegeben ist.

[0065] Der SRAM/DRAM-Speicher **405** weist ferner eine Unterbrechungsbehandlungsroutine **422** auf. Die Unterbrechungsbehandlungsroutine **422** sieht für den Prozessor **400** als Reaktion auf eine Systemunterbrechung **200** Code vor.

[0066] Das Flash-EPROM **410** weist einen Datenbereich **430**, einen Reserveblockbereich **432** und einen Codebereich **434** auf. Der Reserveblockbereich wird für ein dem Stand der Technik entsprechende Blockverwaltung eingesetzt. Der Codebereich wird zum Speichern verschiedener Routinen verwendet, die durch den Prozessor **400** ausgeführt werden können.

[0067] In einem Ausführungsbeispiel beträgt die Größe des SRAM/DRAM 1 Mbit oder 128 KB, wobei die Größe des Flash-EPROM 8 Mbit entspricht.

[0068] Die Abbildung aus **Fig. 9** zeigt ein Flussdiagramm beispielhafter Schritte eines Systems mit der Konfiguration gemäß den Abbildungen der **Fig. 8a**, **8b** und **8c**. Das Flussdiagramm beginnt mit dem Block **500**, wobei der Prozessor eine Programmope-

ration ausführt. Auf Block **500** folgt im Ablauf der Block **502**, in dem der Prozessor ein Systemunterbrechungssignal **200** empfängt. In dem Block **504** führt der Prozessor eine Codeerfassung von der Unterbrechungsbehandlungsroutine **422** aus. In einem Ausführungsbeispiel wird die Unterbrechungsbehandlungsroutine in dem SRAM/DRAM-Speicher **405** gespeichert. In einem anderen Ausführungsbeispiel wird die Unterbrechungsbehandlungsroutine in dem Prozessor gespeichert, wie etwa in dem Cache-Speicher oder in einem internen ROM-Speicher. Die Unterbrechungsbehandlungsroutine weist Anweisungen zur Erzeugung eines Programmunterbrechungsbefehls an das Flash-EPROM **410** auf. Der Prozessor **400** führt die Unterbrechungsbehandlungsroutine **422** aus und erzeugt in dem Block **506** den Programmunterbrechungsbefehl an das Flash-EPROM.

[0069] Der Ablauf fährt mit Block **508** fort, in dem der Programmablauf unterbrochen wird, der durch das Flash-EPROM ausgeführt worden ist. Eine oder mehrere Codeerfassungen werden in dem Block **510** aus dem Flash-EPROM-Codebereich **434** ausgeführt. Andere Operationen in dem Flash-EPROM werden optional ebenfalls ausgeführt, während der Programmablauf unterbrochen ist. Der Ablauf fährt mit dem Block **512** fort, wobei der Prozessor einen Programmwiederaufnahmebefehl an das Flash-EPROM erzeugt. Der Prozessor **400** erzeugt den Programmwiederaufnahmebefehl und in dem Block **514** wird der Programmablauf wiederaufgenommen. Das Flussdiagramm endet mit dem Block **520**.

[0070] Somit wird in Bezug auf die Abbildungen der Fig. 8a, 8b, 8c und 9 ein Verfahren zur Unterbrechung eines Programmablaufs unter Verwendung einer Unterbrechungsbehandlungsroutine und eines Programmunterbrechungsbefehls beschrieben.

[0071] Bei der Abbildung aus Fig. 10 handelt es sich um ein Flussdiagramm, das zeigt, dass unter Verwendung einer Verschachtelungstechnik mehrere Operationen unterbrochen werden können. Das Flussdiagramm beginnt mit Block **600**. Der Ablauf fährt mit Block **602** fort, wobei der erste Nicht-Lesevorgang beginnt. In einem Ausführungsbeispiel kann als erste (oder äußere) verschachtelte unterbrochene Operation nur der Löschvorgang unterbrochen werden. In dem vorliegenden Ausführungsbeispiel weist der Löschvorgang eine verhältnismäßig niedrige Priorität auf, sodass alle anderen bevorrechtigt behandelt werden können. Der Programmablauf weist eine höhere Priorität auf, sodass er nur von bestimmten Operationen bevorrechtigt werden kann. In einem anderen Ausführungsbeispiel kann es sich bei dem äußeren verschachtelten, unterbrochenen Vorgang um einen Löschvorgang, eine Programmoperation, eine Statusleseoperation oder eine Befehlsoperation handeln.

[0072] In dem Block **604** wird der erste Nicht-Lesevorgang unterbrochen. Eingeleitet wird die Unterbrechung durch Schreiben eines Unterbrechungsbefehls

an den Befehlsdecodierer **170**. Der unterbrochene Vorgang wird erst wiederaufgenommen, wenn während dem unterbrochenen Vorgang eingeleitete Operationen abgeschlossen worden sind. In einem Ausführungsbeispiel geht das RY/BY#-Signal in einen hohen Zustand über, um anzuzeigen, dass der Flash-Speicher bereit ist. Ein Bit in dem Statuswort, auf das über den Statuslesebefehl zugegriffen wird, zeigt dabei an, dass eine Operation unterbrochen worden ist. In einem Ausführungsbeispiel weist das Statuswort ein Bit auf, das anzeigt, dass ein Programmablauf unterbrochen worden ist, und wobei ein anderes Bit anzeigt, ob die Schreibzustandsvorrichtung beschäftigt ist bzw. arbeitet.

[0073] In dem Block **606** können eine oder mehrere andere Operationen ausgeführt werden. In einem Ausführungsbeispiel sind nach der Unterbrechung des ersten Nicht-Lesevorgangs nur bestimmte Operationen zulässig. Wenn zum Beispiel ein Löschvorgang unterbrochen worden ist, dann sind nur die folgenden Operationen zulässig: Lesen, Programmieren, Programmierungsunterbrechung, Programmierungswiederaufnahme, Status lesen und Löschvorgang wiederaufnehmen.

[0074] In dem Block **608** beginnt ein zweiter Nicht-Lesevorgang. In einem Ausführungsbeispiel handelt es sich bei dem zweiten Nicht-Lesevorgang um eine Programmoperation. Das RY/BY#-Signal wechselt auf einen niedrigen Wert, um anzuzeigen, dass der Flash-Speicher belegt ist.

[0075] Der Ablauf fährt mit dem Block **610** fort, wobei der zweite Nicht-Lesevorgang unterbrochen wird. Das RY/BY#-Signal wechselt auf einen hohen Wert, um anzuzeigen, dass der Flash-Speicher bereit ist.

[0076] In dem Block **612** können eine oder mehrere Operationen ausgeführt werden. In einem Ausführungsbeispiel sind nur bestimmte Operationen zulässig, während ein Programmablauf unterbrochen ist. In einem Ausführungsbeispiel handelt es sich bei den einzig zulässigen Operationen um die Vorgänge Lesen, Status lesen und Programmwiederaufnahme.

[0077] In dem Block **614** wird der zweite Nicht-Lesevorgang wiederaufgenommen. Dies wird durch Schreiben eines Wiederaufnahmebefehls an den Befehlsdecodierer **170** erreicht. Die den Blöcken **610–614** entsprechenden Schritte können während dem zweiten Nicht-Lesevorgang mehrfach ausgeführt werden.

[0078] In dem Block **616** wird der zweite Nicht-Lesevorgang abgeschlossen. Der Ablauf fährt mit dem Block **618** fort, in dem eine oder mehrere Operationen ausgeführt werden können.

[0079] In dem Block **620** wird der erste Nicht-Lesevorgang durch Schreiben eines Wiederaufnahmebefehls an den Befehlsdecodierer **170** wiederaufgenommen. Die den Blöcken **604–620** entsprechenden Schritte können während dem ersten Nicht-Lesevorgang mehrfach ausgeführt werden.

[0080] In dem Block **622** wird der erste Nicht-Lesevorgang abgeschlossen. Das Flussdiagramm endet

mit dem Block **630**.

[0081] Neben den in der genauen Beschreibung beschriebenen Ausführungsbeispielen unter Verwendung eines Flash-EPROM kann die Erfindung auch in Verbindung mit jedem anderen nichtflüchtigen, beschreibbaren Speicher verwendet werden, wozu unter anderem die folgenden Speicher zählen: EPROMs, EEPROMs und Flash-Speicher, einschließlich Technologien wie NOR, NAND, AND, Divided bit-line NOR (DINOR) und ferroelektrischer Direktzugriffsspeicher (FRAM).

[0082] In der vorstehenden Beschreibung wurde die Erfindung in Bezug auf bestimmte Ausführungsbeispiele der vorliegenden Erfindung beschrieben. Diesbezüglich ist es jedoch offensichtlich, dass verschiedene Modifikationen und Änderungen möglich sind, ohne dabei vom weiteren Umfang der vorliegenden Erfindung gemäß den Ausführungen in den anhängigen Ansprüchen abzuweichen. Die Beschreibung und die Zeichnungen dienen somit den Zwecken der Veranschaulichung und schränken die Erfindung nicht ein.

Patentansprüche

1. Verfahren zum Lesen von Code aus einem nichtflüchtigen Speicher (**410**) in einem System, das einen Prozessor (**400**) umfasst, der mit einem nichtflüchtigen Speicher (**410**) und einem flüchtigen Speicher (**405**) gekoppelt ist, wobei eine Unterbrechungsbehandlungseinrichtung (**422**) in dem flüchtigen Speicher (**405**) gespeichert ist, wobei das genannte Verfahren gekennzeichnet ist durch:

- (a) Empfangen eines Unterbrechungssignals;
- (b) Abrufen eines Unterbrechungscode von der Unterbrechungsbehandlungseinrichtung (**422**) als Reaktion auf den Empfang des Unterbrechungssignals, wobei der Unterbrechungscode einen oder mehrere Befehle zum Vorsehen eines Programmunterbrechungsbefehls an den nichtflüchtigen Speicher (**410**) aufweist;
- (c) Vorsehen eines Programmunterbrechungsbefehls an den nichtflüchtigen Speicher (**410**) als Reaktion auf das Unterbrechungssignal;
- (d) Unterbrechen eines Programmablaufs, der in dem nichtflüchtigen Speicher (**410**) ausgeführt wird;
- (e) Lesen von Code aus dem nichtflüchtigen Speicher (**410**);
- (f) Vorsehen eines Programmwiederaufnahmebefehls an den nichtflüchtigen Speicher (**410**); und
- (g) Wiederaufnehmen des Programmablaufs.

2. Verfahren nach Anspruch 1, wobei das Verfahren ferner den folgenden Schritt aufweist:

- (h) Ausführen des Codelesevorgangs aus dem nichtflüchtigen Speicher (**410**).

3. System zum Lesen von Code aus einem nichtflüchtigen Speicher (**410**), der mit einem Prozessor gekoppelt ist und einem flüchtigen Speicher, wobei

eine Unterbrechungsbehandlungseinrichtung (**422**) in einem flüchtigen Speicher (**405**) gespeichert ist, wobei das System gekennzeichnet ist durch:

- eine Einrichtung zum Empfangen eines Unterbrechungssignals;
- eine Einrichtung zum Abrufen eines Unterbrechungscode von der Unterbrechungsbehandlungseinrichtung (**422**) als Reaktion auf den Empfang des Unterbrechungssignals, wobei der Unterbrechungscode einen oder mehrere Befehle zum Vorsehen eines Programmunterbrechungsbefehls an den nichtflüchtigen Speicher (**410**) aufweist;
- eine Einrichtung zum Vorsehen eines Programmunterbrechungsbefehls an den nichtflüchtigen Speicher (**410**) als Reaktion auf das Unterbrechungssignal;
- eine Einrichtung zum Unterbrechen eines Programmablaufs, der in dem nichtflüchtigen Speicher ausgeführt wird;
- eine Einrichtung zum Lesen von Code aus dem nichtflüchtigen Speicher (**410**);
- eine Einrichtung zum Vorsehen eines Programmwiederaufnahmebefehls an den nichtflüchtigen Speicher (**410**); und
- eine Einrichtung zur Wiederaufnahme des Programmablaufs.

4. System nach Anspruch 3, wobei ferner eine Einrichtung zum Ausführen des Codelesevorgangs aus dem nichtflüchtigen Speicher (**410**) vorgesehen ist.

5. Verfahren zum Lesen von Code aus einem nichtflüchtigen Speicher (**410**), der mit einem Prozessor (**400**) gekoppelt ist und mit einem flüchtigen Speicher (**405**), der eine Unterbrechungsbehandlungseinrichtung (**422**) speichert, wobei das genannte Verfahren gekennzeichnet ist durch:

- (a) Empfangen eines Unterbrechungssignals;
- (b) Abrufen eines Unterbrechungscode von der Unterbrechungsbehandlungseinrichtung (**422**) als Reaktion auf den Empfang des Unterbrechungssignals, wobei der Unterbrechungscode einen oder mehrere Befehle zum Vorsehen eines Programmunterbrechungsbefehls an den nichtflüchtigen Speicher (**410**) aufweist;
- (c) Vorsehen eines Programmunterbrechungsbefehls an den nichtflüchtigen Speicher (**410**) als Reaktion auf das Unterbrechungssignal;
- (d) Unterbrechen eines Programmablaufs, der in dem nichtflüchtigen Speicher (**410**) ausgeführt wird, als Reaktion auf den Unterbrechungsbefehl;
- (e) Unterbrechen eines Löschvorgangs, der in dem nichtflüchtigen Speicher (**410**) ausgeführt wird, als Reaktion auf den Unterbrechungsbefehl;
- (f) Lesen von Code aus dem nichtflüchtigen Speicher (**410**);
- (g) Vorsehen eines Programmwiederaufnahmebefehls an den nichtflüchtigen Speicher (**410**);
- (h) Vorsehen eines Löschvorgangs-Wiederaufnahmebefehls an den nichtflüchtigen Speicher (**410**);

- (i) Wiederaufnahmen des Löschvorgangs; und
- (j) Wiederaufnahmen des Programmablaufs.

Es folgen 8 Blatt Zeichnungen

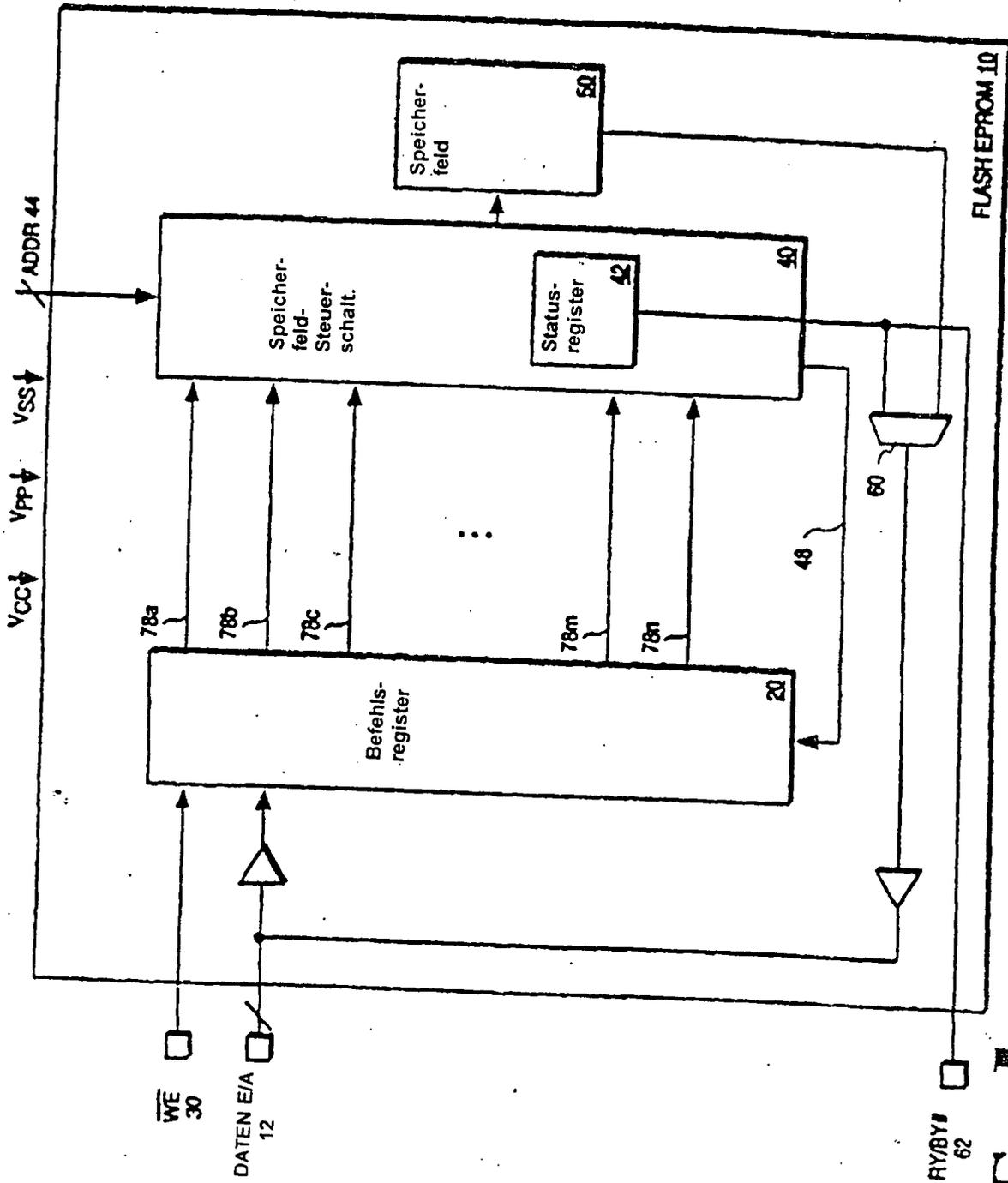


FIG. 1 (STAND DER TECHNIK)

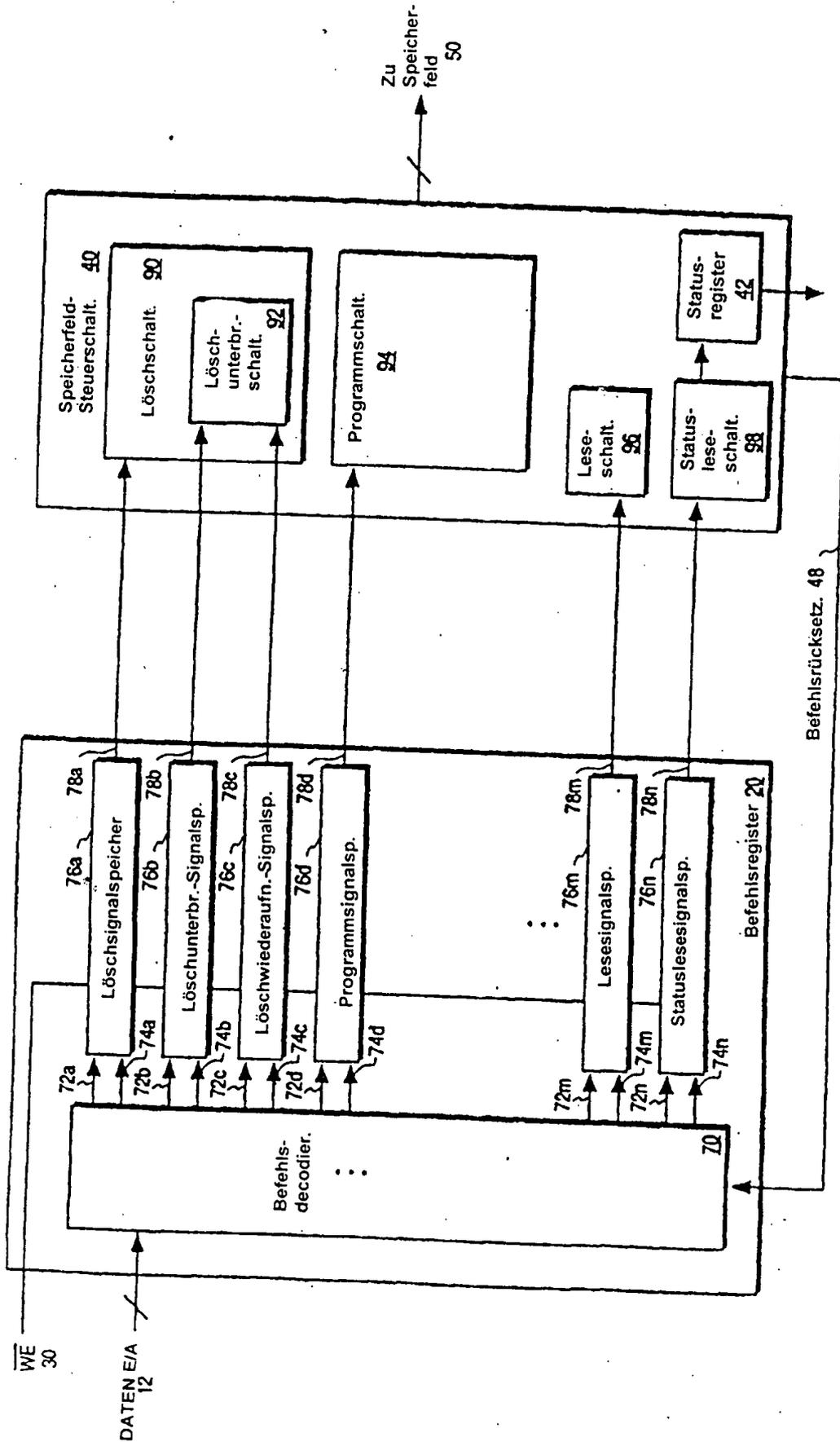


FIG. 2 (Stand der Technik)

FIG. 3

(Stand der Technik)

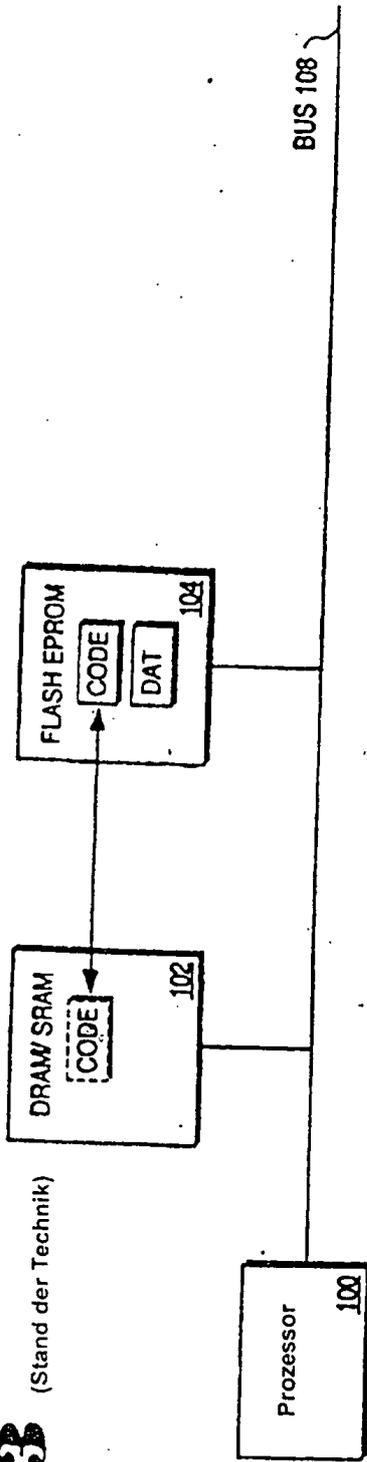


FIG. 4

(Stand der Technik)

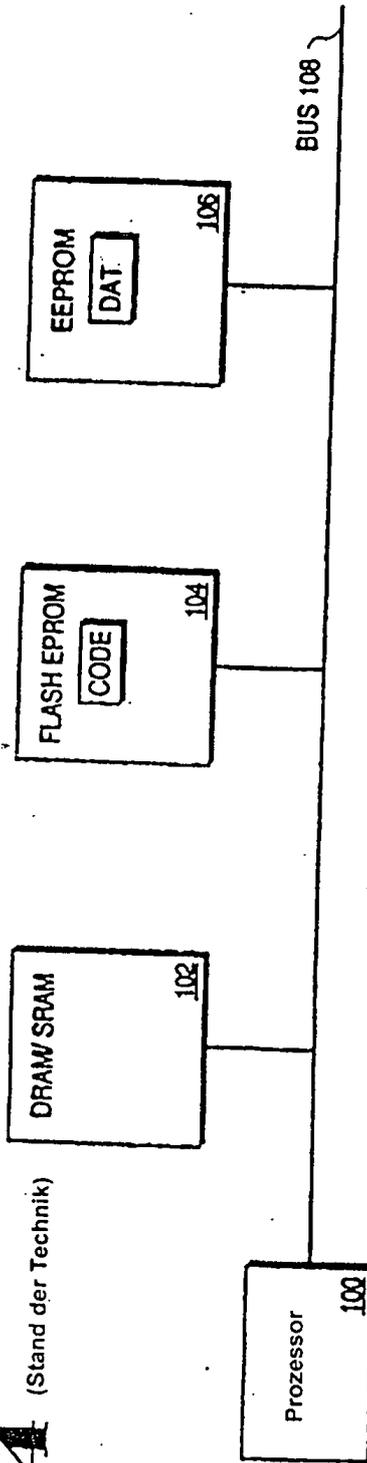
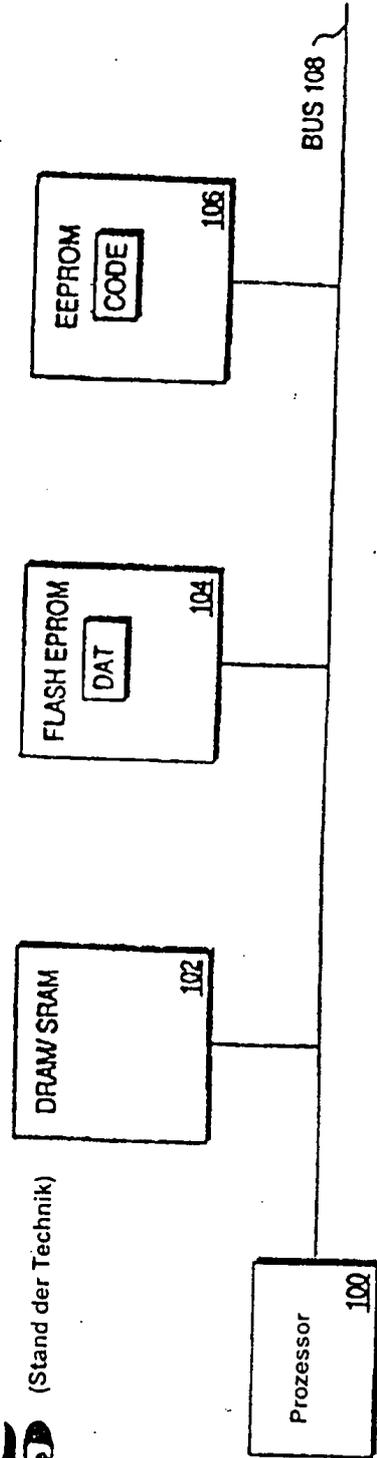


FIG. 5

(Stand der Technik)



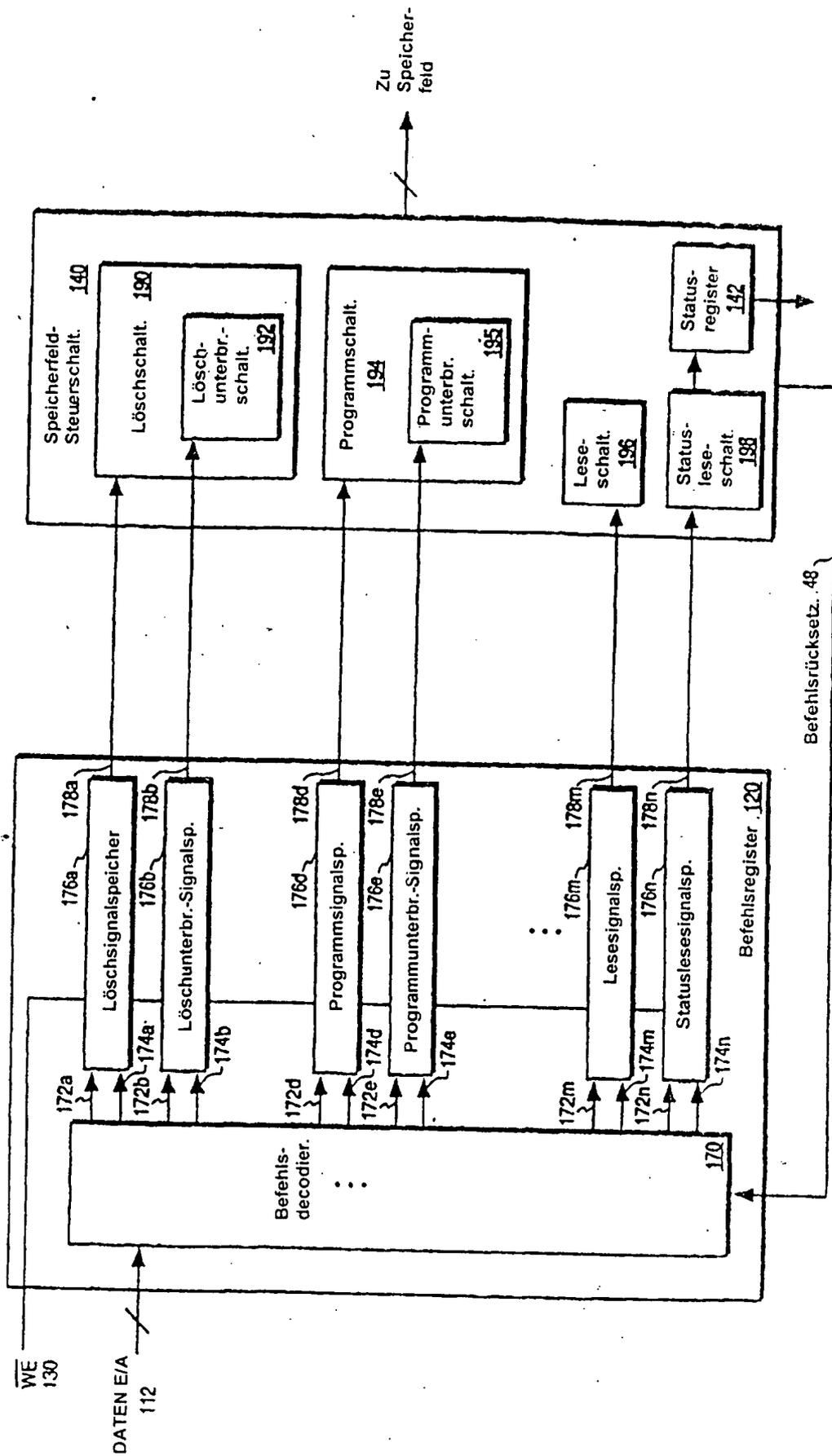


FIG. 6

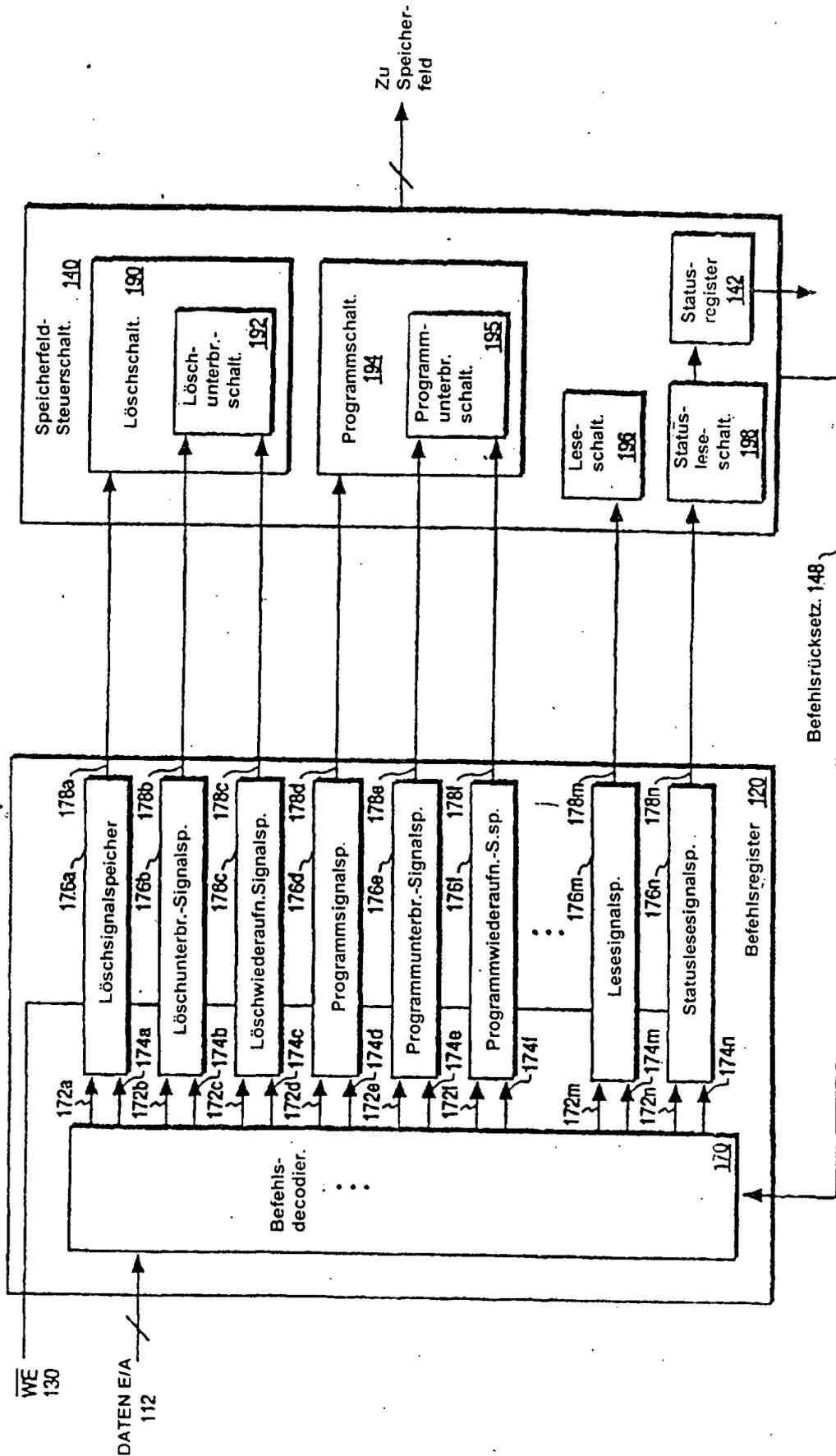


FIG. 7

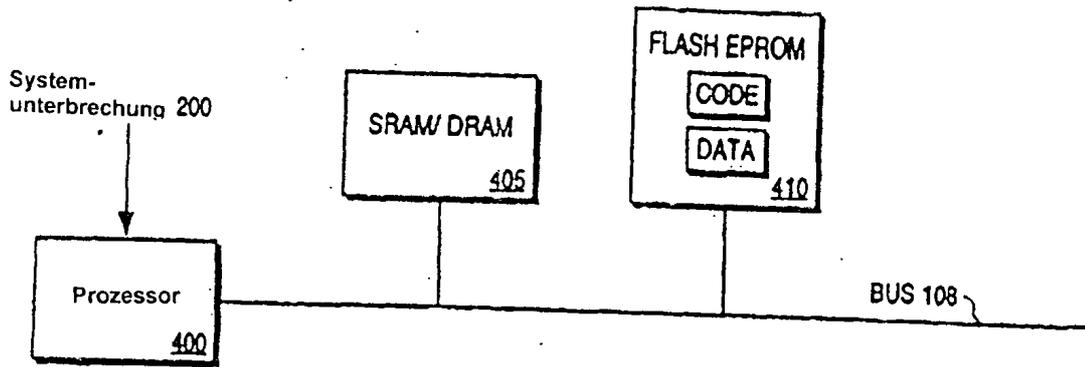


FIG. 8A

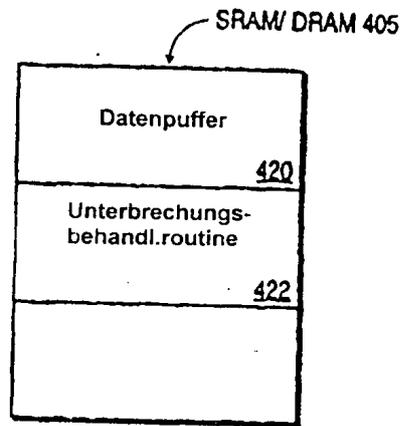


FIG. 8B

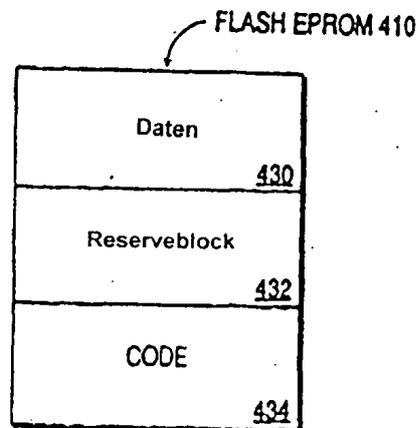


FIG. 8C

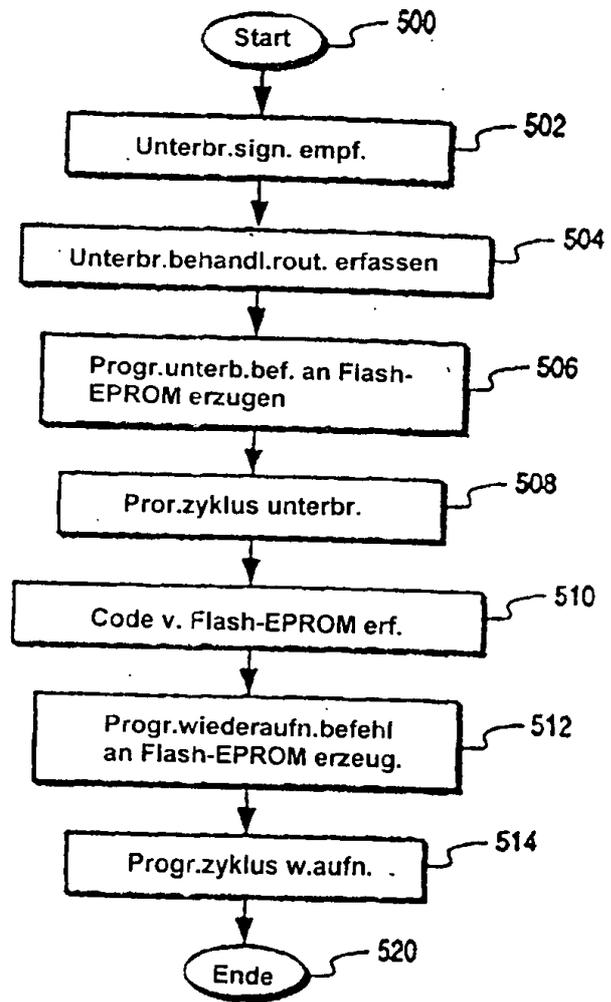


FIG. 9

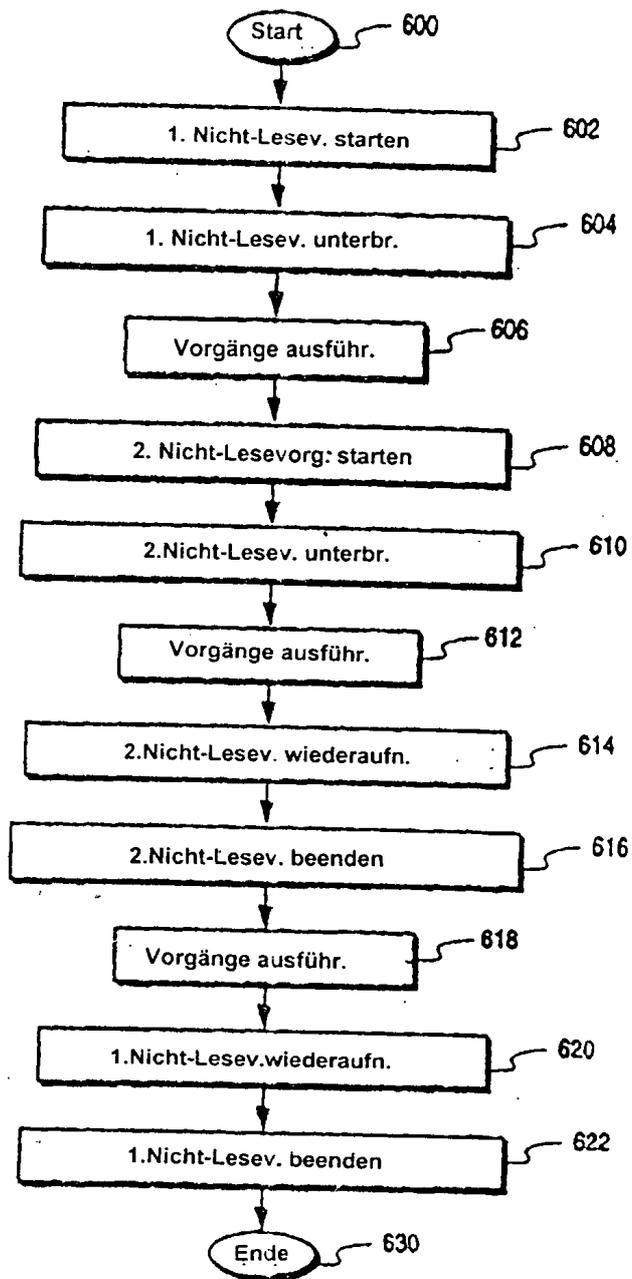


FIG. 10