(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 601 00 779 T2** 2004.07.15

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 199 724 B1**
(21) Deutsches Aktenzeichen: **601 00 779.4**
(96) Europäisches Aktenzeichen: **01 118 067.6**
(96) Europäischer Anmeldetag: **25.07.2001**
(97) Erstveröffentlichung durch das EPA: **24.04.2002**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **17.09.2003**
(47) Veröffentlichungstag im Patentblatt: **15.07.2004**

(51) Int Cl.[7]: **G11C 7/10**
G11C 7/22, G11C 11/4093, G11C 11/4096

(30) Unionspriorität:

2000299384	**29.09.2000**	**JP**
2000374153	**08.12.2000**	**JP**

(84) Benannte Vertragsstaaten:
DE, FR

(73) Patentinhaber:
Mitsubishi Denki K.K., Tokio/Tokyo, JP

(74) Vertreter:
PRÜFER & PARTNER GbR, 81545 München

(72) Erfinder:
Yamauchi, Tadaaki, Tokyo 100-8310, JP; Kozaru, Kunihiko, Tokyo 100-8310, JP

(54) Bezeichnung: **Halbleiteranordnung mit einer unkomplizierten Schnittstelle sowie einer logischen Schaltung und einer eingebauten Speicheranordnung**

Beschreibung

HINTERGRUND DER ERFINDUNG

Gebiet der Erfindung

[0001] Die vorliegende Erfindung bezieht sich auf eine Halbleiterspeichereinrichtung und insbesondere auf eine Halbleiterspeichereinrichtung mit einer Logikschaltungsanordnung, die darin integriert ist, und ein Steuerverfahren dafür.

Beschreibung der Hintergrundstechnik

[0002] **Fig.** 101 ist eine Draufsicht, die eine Stiftkonfiguration eines 64 Mbit synchronen dynamischen Direkzugriffsspeicher (SDRAM) mit einer 16 Bit Wortkonfiguration darstellt.

[0003] **Fig.** 102 ist eine Tabelle, die Anschlußnamen des SDRAM und ihre Funktionen darstellt.

[0004] Bezug nehmend auf **Fig.** 101 und 102 ist ein SDRAM des Standes der Technik zum Beispiel in einem Gehäuse mit 54 Stiftanschlüssen aufgenommen, die einen Anschluß CLK, an den ein Mastertakt eingegeben wird, einen Anschluß CKE, an den ein Taktfreigabesignal eingegeben wird, einen Anschluß /CS, an den ein Chipauswahlsignal eingegeben wird, einen Anschluß /RAS, an den ein Zeilenadreßstrobesignal eingegeben wird, einen Anschluß /CAS, an den ein Spaltenadreßstrobesignal eingegeben wird, und einen Anschluß /WE, an den ein Schreibfreigabesignal eingegeben wird, enthalten.

[0005] Ein SDRAM des Standes der Technik weist weiter Anschlüsse DQ0 bis DQ15, die ein Daten-I/O-Signal liefert/empfangen, einen Anschluß DQM (U/L), durch den ein Ausgangssperrsignal/Schreibmaskierungssignal eingegeben/ausgegeben wird, Anschlüsse A0 bis A11, an die eine Adresse eingegeben wird, Anschlüsse BA0 und BA1, an die eine Bankadresse eingegeben wird, einen Anschluß VDD, der mit einer Leistungsquelle beliefert wird, einen Anschluß VDDQ, der mit einer Ausgangsleistungsquelle beliefert wird, einen Anschluß VSS, der mit einem Massepotential versehen ist, und einen Anschluß VSSQ, der mit einem Ausgangsmassepotential versehen ist, auf.

[0006] Die Anschlüsse sind derart aufgebaut, wie in **Fig.** 101 gezeigt ist, die Daten-I/O-Anschlüsse und die Leistungsquellen sind zwischen dem ersten und dreizehnten Stift und zwischen dem zweiundvierzigsten und vierundfünfzigsten Anschluß vorgesehen; die Steuersignale und das Taktsignal sind zwischen dem fünfzehnten und dem neunzehnten Stift und zwischen dem siebenunddreißigsten und dem neununddreißigsten Stift vorgesehen, und die Adreßstifte sind zwischen dem zwanzigsten und dem fünfunddreißigsten Stift vorgesehen. Solch eine Anschlußkonfiguration ist auf einem Niveau allgemeiner Vielseitigkeit und auch gut benutzt in einem Substrat, auf dem ein System mit einem Speicher angebracht ist.

[0007] **Fig.** 103 ist ein Blockschaltbild, das eine Konfiguration einer Logikschaltung integriert mit einem dynamischen Direktzugriffsspeicher (hier im folgenden als DRAM bezeichnet) des Standes der Technik darstellt.

[0008] Bezug nehmend auf **Fig.** 103 sind ein DRAM **504** und eine Logikschaltung **508** auf einem Chip **501** integriert und mit Anschlüssen zum Eingeben oder Ausgeben von Steuersignalen /RAS, /CAS, ..., /CS zum Zugriff auf den DRAM, eines Adreßssignales ADD und eines Datensignales DATA versehen.

[0009] In dem Chip **501** sind weiter Steuerstifte CTR0 und CTR1 eindeutig für die Logik, ein Anschluß, an dem ein Anforderungssignal REQ eingegeben wird, das Zugriff zu der Logikschaltung anfordert, und ein Anschluß zum Ausgeben eines Strobesignales STRB zum Benachrichtigen der Außenseite, daß die Logikschaltung eine Verarbeitung beendet, enthalten.

[0010] Da im Stand der Technik Stifte eindeutig für die Logikschaltung 508 zum Steuern der Logikschaltung **508** vorgesehen waren, nahm die Zahl der Stifte im Vergleich mit dem in **Fig.** 101 gezeigten Allgemeinzweck-DRAM zu; zum Zusammensetzen eines Systems auf einer Baugruppe mußte eine speziell zugeschnittene Steuerung zum Steuern eines integrierten Logik-DRAM vorgesehen werden. Folglich ging eine allgemeine Vielfalt, wie eine Verbindung mit einem gewöhnlichen Mikrocomputer, verloren, oder spezielle Befehle wurden in einem Mikrocomputer zum Steuern des Systems benötigt.

[0011] Aus der US 5,862,396 A kann eine integrierte Halbleiterschaltungsvorrichtung entnommen werden mit einem Hauptspeicher mit einer arithmetischen Logikverarbeitungsfähigkeit mit ersten Speichern, die mit einem Speicherbus verbunden sind, zum Speichern von Daten von zweiten Speichern mit einer arithmetischen Logikverarbeitungsfähigkeit, die ebenfalls mit dem Bus verbunden sind. Jeder der zweiten Speicher enthält einen Speicherabschnitt zum Speichern von Daten und einen arithmetischen Logikverarbeitungsabschnitt. Der arithmetische Logikverarbeitungsabschnitt führt eine erste Verarbeitung auf mindestens einem Teil der in dem Speicherabschnitt gespeicherten Daten als Reaktion auf einen ersten Befehl, der über den Speicherbus eingegeben ist, durch und ermöglicht, daß ein Resultat der ersten Verarbeitung auf den Speicherbus als Reaktion auf einen zweiten Befehl ausgegeben wird, der über den Speicherbus eingegeben wird.

[0012] Aus der US 4,835,733 kann eine integrierte Halbleiterschaltungsvorrichtung entnommen werden, die Verarbeitungsfähigkeit auf dem gleichen Chip enthält, auf einem oder auf beiden eines Adreßpfades und eines Datenpfades zwischen einem Satz von Zugriffsregistern und einem Speicherfeld so, daß eine Adresse erzeugt, geprüft und manipuliert werden kann, und/oder Daten manipuliert werden können oder mit Referenzmustern von Daten verglichen werden können.

[0013] Aus der US 5,953,738 kann eine integrierte Halbleiterschaltungsvorrichtung entnommen werden mit einem Speicher, der als ein einzelner integrierter Schaltungschip hergestellt ist und ein Feld von Speicherzellen und Schaltungsanordnung zum Zugreifen auf ausgewählte Speicherzellen in dem Feld enthält. Mindestens eine ALU ist zum Empfangen eines Datenzugriffs von ausgewählten Zellen des Feldes und Durchführen einer ausgewählten Operation darauf enthalten.

ZUSAMMENFASSUNG DER ERFINDUNG

[0014] Es ist folglich eine Aufgabe, eine integrierte Halbleiterschaltungsvorrichtung vorzusehen, auf der eine Speicherschaltungsanordnung und eine Logikschaltungsanordnung integriert sind, die durch ein Steuerverfahren ähnlich zu dem eines Allzweck-DRAM gesteuert werden können und die in der Lage sind, ein Resultat einer vorgeschriebenen Logikoperation zu liefern/zu empfangen, die auf Daten durchgeführt wird, die in der Speicherschaltung gespeichert sind, durch eine leicht zu handhabende Schnittstelle.

[0015] Solch eine Aufgabe wird gelöst durch eine integrierte Halbleiterschaltungsvorrichtung mit den Merkmalen des Anspruchs 1.

[0016] Das heißt, die vorliegende Erfindung ist eine integrierte Halbleiterschaltungsvorrichtung und enthält eine Anschlußgruppe, ein Speicherzellenfeld und Logikschaltungsanordnung.

[0017] Bevorzugte Ausgestaltungen der Erfindung sind in den abhängigen Ansprüchen definiert.

[0018] Folglich ist es ein Vorteil der vorliegenden Erfindung, daß die integrierte Logikschaltungsanordnung gemäß einer Sequenz ähnlich zu der gesteuert werden kann, gemäß der Daten, eine Adresse und ein Steuersignal an einen Allzweckspeicher gegeben werden, und das System kann ohne Änderung eines vorhandenen Systems erhalten werden, das gut und leicht zu steuern ist. Weiterhin ist ein Vorteil der vorliegenden Erfindung der, daß eine integrierte Halbleiterschaltung, auf der eine Speicherschaltungsanordnung und eine Logikschaltungsanordnung integriert sind, ein Resultat einer vorgeschriebenen Logikoperation auf Daten, die in der Speicherschaltungsanordnung gespeichert sind, mit einer hohen Geschwindigkeit durch eine Schnittstelle liefern/empfangen kann, die leicht extern zu handhaben ist.

[0019] Die vorangehenden und weiteren Aufgaben, Merkmale, Aspekte und Vorteile der folgenden Erfindung werden ersichtlicher aus der folgenden detaillierten Beschreibung der vorliegenden Erfindung, wenn sie in Zusammenhang mit den begleitenden Zeichnungen genommen wird.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0020] **Fig.** 1 ist ein Blockschaltbild, das eine Konfiguration einer integrierten Halbleiterschaltungsvorrichtung **1** eines ersten Beispieles der vorliegenden Erfindung darstellt;

[0021] **Fig.** 2 ist eine Zeichnung, die ein Speicherabbild einer integrierten Logikhalbleiterspeichereinrichtung des ersten Beispieles darstellt;

[0022] **Fig.** 3 ist ein Blockschaltbild zum Beschreiben eines Weges, wie ein extern eingegebenes Signal zu einer Logikschaltung übertragen wird;

[0023] **Fig.** 4 ist ein Blockschaltbild zum Beschreiben eines Betriebes in einem normalen Modus, einem von Betriebsmodi der integrierten Halbleiterschaltungsvorrichtung **1**;

[0024] **Fig.** 5 ist ein Blockschaltbild zum Beschreiben eines Betriebes eines Lesezugriffes in dem normalen Modus;

[0025] **Fig.** 6 ist ein Blockschaltbild zum Beschreiben eines Betriebes in einem Blockmodus, einem anderen der Betriebsmodi der integrierten Halbleiterschaltungsvorrichtung **1**;

[0026] **Fig.** 7 ist ein Blockschaltbild zum Beschreiben eines Betriebes in einem Puffermodus, einem noch anderen der Betriebsmodi der integrierten Halbleiterschaltungsvorrichtung **1**;

[0027] **Fig.** 8 ist ein Flußdiagramm zum Beschreiben von Betrieben in den drei Modi;

[0028] **Fig.** 9 ist eine Tabelle, die Beispiele von Kryptosystemen darstellt, die eine Logikschaltung **8** ausführen kann;

[0029] **Fig.** 10 ist eine konzeptuelle Zeichnung, die eine Fundamentaleinheit für DES-Verschlüsselung darstellt, die als ein Kryptosystem mit geheimem Schlüssel benutzt wird;

[0030] **Fig.** 11 ist eine erste konzeptuelle Zeichnung, die ein Triple-DES-Verarbeitungssystem darstellt;

[0031] **Fig.** 12 ist eine zweite konzeptuelle Zeichnung, die das Triple-DES-Verarbeitungssystem darstellt;

[0032] **Fig.** 13 ist eine konzeptuelle Zeichnung, die Entschlüsselung darstellt, die entsprechend zu **Fig.** 10 durchgeführt wird;

[0033] **Fig.** 14 ist eine konzeptuelle Zeichnung, die Entschlüsselung darstellt, die entsprechend zu **Fig.** 11 durchgeführt wird;

[0034] **Fig.** 15 ist eine konzeptuelle Zeichnung, die Entschlüsselung darstellt, die entsprechend zu **Fig.** 12 durchgeführt wird;

[0035] **Fig.** 16 ist eine konzeptuelle Zeichnung zum Beschreiben einer Verschlüsselung in einem EBC-Modus;

[0036] **Fig.** 17 ist eine konzeptuelle Zeichnung zum Darstellen von Entschlüsselung in dem EBC-Modus;

[0037] **Fig.** 18 ist eine konzeptuelle Zeichnung zum Beschreiben von Verschlüsselung in einem CBC-Modus;

[0038] **Fig.** 19 ist eine konzeptuelle Zeichnung, die Verarbeitung zum Dechiffrieren eines Chiffriertextes darstellt, der auf die in **Fig.** 18 gezeigte Weise chiffriert ist;

[0039] **Fig.** 20 ist ein Zeitablaufdiagramm zum Be-

schreiben der Verarbeitung, die in **Fig.** 8 beschrieben ist, wenn verschiedene Arten von Verschlüsselungsmodi vorhanden sind;

[0040] **Fig.** 21 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, wenn Daten auf die gleiche Seite bei Schreibzugriff des normalen Modus geschrieben werden;

[0041] **Fig.** 22 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, wenn Daten von 64 Bit auf eine andere Seite in dem normalen Modus geschrieben werden;

[0042] **Fig.** 23 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, wenn Lesezugriff auf die gleiche Seite in dem normalen Modus durchgeführt wird;

[0043] **Fig.** 24 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, wenn Lesezugriff auf eine andere Seite durchgeführt wird;

[0044] **Fig.** 25 ist ein Zeitablauf diagram zum Beschreiben eines Betriebes der integrierten Halbleiterschaltungsvorrichtung **1**, wenn ein externes Taktsignal Ext.CLK gleich 50 MHz ist;

[0045] **Fig.** 26 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, wenn Schreibzugriff auf eine andere Seite in dem normalen Modus durchgeführt wird;

[0046] **Fig.** 27 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, wenn Lesezugriff auf die gleiche Seite in dem normalen Modus durchgeführt wird;

[0047] **Fig.** 28 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, wenn Lesezugriff auf eine andere Seite in dem normalen Modus durchgeführt wird;

[0048] **Fig.** 29 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, wenn ein internes Taktsignal clkL, das an die Logikschaltung **8** gegeben wird, zu 50 MHz umgewandelt wird;

[0049] **Fig.** 30 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, wenn Lesezugriff auf die gleiche Seite in dem normalen Modus durchgeführt wird;

[0050] **Fig.** 31 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, wenn Lesezugriff auf eine andere Seite in dem normalen Modus durchgeführt wird;

[0051] **Fig.** 32 ist ein konzeptuelles Blockschaltbild, das einen Weg darstellt, wie eine integrierte Halbleiterschaltungsvorrichtung **1** der vorliegenden Erfindung und ein Mikroprozessor **90** verbunden werden;

[0052] **Fig.** 33 ist ein konzeptuelles Blockschaltbild, das einen anderen Weg darstellt, wie die integrierte Halbleiterschaltungsvorrichtung **1** der vorliegenden Erfindung und ein Mikroprozessor **90** verbunden werden;

[0053] **Fig.** 34 ist ein Flußdiagramm zum Beschreiben der Steuerung der integrierten Halbleiterschaltungsvorrichtung **1**;

[0054] **Fig.** 35 ist ein konzeptuelles Blockschaltbild, das ein Beispiel eines Systems darstellt, das für die Anwendung eines Blockmodus der integrierten Halbleiterschaltungsvorrichtung **1** geeignet ist;

[0055] **Fig.** 36 ist ein konzeptuelles Blockschaltbild, das eine Konfiguration darstellt, wenn eine in dem Blockmodus tätige integrierte Halbleiterschaltungsvorrichtung **1** auf ein System angewendet wird, in dem ein Cache-Speicher **96** vorhanden ist;

[0056] **Fig.** 37 ist ein schematisches Blockschaltbild, das eine Konfiguration eines Systems zeigt, das geeignet ist, wenn ein Puffermodus der integrierten Halbleiterschaltungsvorrichtung **1** angenommen wird;

[0057] **Fig.** 38 ist ein Blockschaltbild, das eine Konfiguration eines integrierten Logik-DRAM **30** eines dritten Beispieles darstellt, das durch Modifizieren der Konfiguration der integrierten Halbleiterschaltungsvorrichtung **1** des ersten Beispieles erhalten ist;

[0058] **Fig.** 39 ist eine Zeichnung, die ein Speicherabbild eines Systems darstellt, das auf den integrierten Logik-DRAM **30** des dritten Beispieles angewendet ist;

[0059] **Fig.** 40 ist Zeichnungen, die Datenschreiben in ein erstes Datenregister **84** darstellen;

[0060] **Fig.** 41 ist Zeichnungen, die Datenlesen von dem ersten Datenregister **84** darstellen;

[0061] **Fig.** 42 ist ein erstes Flußdiagramm zum Beschreiben eines Betriebes des in **Fig.** 38 gezeigten integrierten Logik-DRAM **30**;

[0062] **Fig.** 43 ist ein Flußdiagramm zum Schreiben eines anderen Betriebes des in **Fig.** 38 gezeigten integrierten Logik-DRAM **30**;

[0063] **Fig.** 44 ist ein Blockschaltbild, das eine Konfiguration darstellt, wenn das erste und das zweite Register aus SRAMs dargestellt sind;

[0064] **Fig.** 45 ist ein schematisches Bild zum Beschreiben einer Konfiguration eines integrierten Logik-DRAM **130** eines vierten Beispieles der vorliegenden Erfindung;

[0065] **Fig.** 46 ist ein konzeptuelles Blockschaltbild zum Beschreiben einer Register-Registertätigkeit;

[0066] **Fig.** 47 ist ein Flußdiagramm zum Beschreiben eines Betriebes des integrierten Logik-DRAM **130** auf eine detailliertere Weise;

[0067] **Fig.** 48 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes des integrierten Logik-DRAM **130** in dem Prozeßfluß, wie in **Fig.** 47 gezeigt ist;

[0068] **Fig.** 49 ist ein konzeptuelles Blockschaltbild zum Beschreiben eines anderen Betriebes des in **Fig.** 45 gezeigten integrierten Logik-DRAM **130**;

[0069] **Fig.** 50 ist eine konzeptuelle Zeichnung zum Beschreiben eines Konzeptes von Betrieben des ersten und des zweiten Registers **84** und **86** und Zählern **85** und **87**;

[0070] **Fig.** 51 ist ein Flußdiagramm zum Beschreiben von mehr Einzelheiten des unter Bezugnahme auf **Fig.** 49 beschriebenen Betriebes;

[0071] **Fig.** 52 ist ein Blockschaltbild, das einen Zustand eines externen Busses darstellt;

[0072] **Fig.** 53 ist ein Zeitablaufdiagramm, das ei-

nen Betrieb eines Vollseitenmodus in einem Register-DRAM-Übertragungsmodus darstellt;

[0073] **Fig.** 54 ist eine Zeichnung, die eine Zuordnung von Adressen zu einem Register 0, einem ersten Datenregister **84** und einem zweiten Datenregister **86** darstellt;

[0074] **Fig.** 55 ist eine Zeichnung, die ein Beispiel von den in den Registern gehaltenen Daten darstellt;

[0075] **Fig.** 56 ist ein schematisches Blockschaltbild zum Beschreiben eines DRAM-Registerbetriebsmodus;

[0076] **Fig.** 57 ist ein Flußdiagramm zum Beschreiben von mehr Einzelheiten des in **Fig.** 56 beschriebenen Betriebes;

[0077] **Fig.** 58 ist eine Zeichnung, die ein Konzept eines Datenübertragungszeitpunktes zum Verbessern einer Übertragungseffektivität darstellt, wenn ein Betrieb in dem Register-DRAM-Übertragungsmodus durchgeführt wird;

[0078] **Fig.** 59 ist ein Zeitablaufdiagramm, das einen Betrieb zum Verbessern einer Effektivität von Datenübertragung beschreibt;

[0079] **Fig.** 60 ist ein Blockschaltbild, das ein Beispiel einer Schaltungskonfiguration zum Durchführen von Verschlüsselung oder Entschlüsselung im CBC-Modus darstellt;

[0080] **Fig.** 61 ist ein schematisches Blockschaltbild zum Beschreiben einer Konfiguration eines integrierten Logik-DRAM **132** eines fünften Beispieles der vorliegenden Erfindung;

[0081] **Fig.** 62 ist ein konzeptuelles Blockschaltbild zum Beschreiben einer Register-Registertätigkeit des integrierten Logik-DRAM **132** des vierten Beispieles;

[0082] **Fig.** 63 ist ein Flußdiagramm zum Beschreiben eines Betriebes des integrierten Logik-DRAM **132** auf eine detailliertere Weise;

[0083] **Fig.** 64 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes des integrierten Logik-DRAM **132** in dem Prozeßfluß, wie in **Fig.** 61 gezeigt ist;

[0084] **Fig.** 65 ist ein konzeptuelles Blockschaltbild zum Beschreiben des Register-DRAM-Betriebes des in **Fig.** 61 gezeigten integrierten Logik-DRAM **132**;

[0085] **Fig.** 66 ist ein Flußdiagramm zum Beschreiben von mehr Einzelheiten des in **Fig.** 65 beschriebenen Betriebes;

[0086] **Fig.** 67 ist eine konzeptuelle Zeichnung zum Beschreiben von Datenübertragungsverarbeitung zwischen dem Register **84** und einer Logikschaltung **74** in einer ersten Modifikation des fünften Beispieles;

[0087] **Fig.** 68 ist ein konzeptuelles Blockschaltbild zum Beschreiben einer Route von dem Register **84** zu einem Datenausgang in einer zweiten Modifikation des fünften Beispieles;

[0088] **Fig.** 69 ist ein konzeptuelles Blockschaltbild zum Beschreiben des DRAM-Registerbetriebsmodus;

[0089] **Fig.** 70 ist ein Flußdiagramm zum Beschreiben von mehr Einzelheiten des in **Fig.** 69 beschriebenen Betriebes;

[0090] **Fig.** 71 ist ein Blockschaltbild, das eine Schaltungskonfiguration darstellt, die einen internen Befehl zum Selbstauffrischen erzeugt;

[0091] **Fig.** 72 ist ein Zeitablaufdiagramm, das ein Verfahren zum Eintritt in einen Niedrigleistungsmodus darstellt;

[0092] **Fig.** 73 ist ein Bild, das eine Schaltungskonfiguration darstellt, die Steuerung von Eingangspuffern **40** oder **42** in dem Niedrigleistungsmodus durchführt;

[0093] **Fig.** 74 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes der in **Fig.** 73 gezeigten Schaltung;

[0094] **Fig.** 75 ist eine Darstellung zum einfachen Beschreiben einer Verarbeitung von Sicherheitsdatenkommunikation in dem Internet;

[0095] **Fig.** 76 ist ein schematisches Blockschaltbild zum Beschreiben eines integrierten Logik-DRAM, der sich auf ein neuntes Beispiel der vorliegenden Erfindung bezieht;

[0096] **Fig.** 77 ist ein schematisches Blockschaltbild, das eine Konfiguration eines DRAM-Steuerabschnittes **42b** und eines Spaltendecoders **58.0** eines Leseverstärkers **60.0** und eines I/O-Abschnittes, der für eine Bank #0 vorgesehen ist, die extrahiert werden, darstellt;

[0097] **Fig.** 78 ist ein Zeitablaufdiagramm zum Beschreiben von Betrieben, bei denen Daten auf ein Bitleitungspaar und weiter auf ein I/O-Leitungspaar LI/O und /LI/O ausgelesen werden;

[0098] **Fig.** 79 ist ein Bild, das eine Konfiguration eines spaltenbezogenen Steuerabschnittes **1206** darstellt;

[0099] **Fig.** 80 ist ein Zeitablaufdiagramm, das Zeitpunkte darstellt, zu denen ein Betrieb des Schreibens ohne Selbstvorladen des Standes der Technik durchgeführt wird;

[0100] **Fig.** 81 ist ein schematisches Blockschaltbild, das eine Konfiguration zum Steuern einer Schreibtätigkeit darstellt, die extrahiert ist;

[0101] **Fig.** 82 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes einer in **Fig.** 81 gezeigten schreibbezogenen Steuerschaltung;

[0102] **Fig.** 83 ist ein schematisches Blockschaltbild zum Beschreiben einer Konfiguration, die einen Selbstauffrischbetrieb steuert zum Verhindern einer Fehlfunktion bei einem Auffrischbetrieb;

[0103] **Fig.** 84 ist ein Diagramm zum Beschreiben eines Effektes des Verringerns einer Stromspitze bei dem in **Fig.** 83 beschriebenen Selbstauffrischbetrieb;

[0104] **Fig.** 85 ist ein schematisches Blockschaltbild, das eine Konfiguration darstellt, die mit einer internen Leistungsquellenpotentialerzeugerschaltung **1100** verknüpft ist;

[0105] **Fig.** 86 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes (Eintritt), bei dem ein Leistungsquellenabschnittmodus betreten wird, und eines Betriebes (Ausgang), bei dem der Leistungsquellenabschnittmodus verlassen wird;

[0106] **Fig.** 87 ist ein Schaltbild, das eine Beispielkonfiguration eines Taktpuffers **44** darstellt;
[0107] **Fig.** 88 ist ein Schaltbild, das eine andere Schaltungskonfiguration des Taktpuffers **44** darstellt;
[0108] **Fig.** 89 ist ein konzeptuelles Blockschaltbild, das eine Konfiguration eines Systems darstellt, in dem ein integrierter Logik-DRAM **1000** verwendet ist;
[0109] **Fig.** 90 ist ein Diagramm, das Betriebsfrequenz eines Speichers darstellt, die gemäß Anwendungen benötigt werden;
[0110] **Fig.** 91 ist ein konzeptuelles Blockschaltbild zum Beschreiben einer Konfiguration, die eine Betriebsgeschwindigkeit eines D-Abschnittes gemäß einer Taktfrequenz ändern kann;
[0111] **Fig.** 92 ist ein schematisches Blockschaltbild, das eine andere Konfiguration darstellt, die ein internes Leistungsquellenpotential gemäß einem externen Taktsignal Ext.CLK steuert;
[0112] **Fig.** 93 ist ein Diagramm zum Beschreiben eines Steuerbetriebes für ein internes Leistungsquellenpotential;
[0113] **Fig.** 94 ist ein schematisches Blockschaltbild zum Beschreiben einer Konfiguration einer Frequenzerfassungsschaltung **1800**;
[0114] **Fig.** 95 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes der in **Fig.** 94 gezeigten Frequenzerfassungsschaltung;
[0115] **Fig.** 96 ist ein schematisches Blockschaltbild zum Beschreiben einer Konfiguration einer internen Leistungsquellenschaltung in der internen Leistungsquellenpotentialerzeugerschaltung **1100**;
[0116] **Fig.** 97 ist ein schematisches Blockschaltbild, das eine andere Konfiguration zum Steuern eines internen Leistungsquellenpotentiales darstellt;
[0117] **Fig.** 98 ist ein Speicherabbild zum Beschreiben eines Beispieles einer Zuordnung in einem Speicherraum, wenn mehrere Arten von Betriebsgeschwindigkeitsmodi vorhanden sind;
[0118] **Fig.** 99 ist ein schematisches Blockschaltbild zum Beschreiben einer Konfiguration einer Treiberschaltung in einem I/O-Puffer **52**;
[0119] **Fig.** 100 ist ein schematisches Blockschaltbild zum Beschreiben einer anderen Konfiguration einer Treiberschaltung in dem in **Fig.** 76 gezeigten I/O-Puffer **52**;
[0120] **Fig.** 101 ist eine Draufsicht einer Stiftkonfiguration eines synchronen dynamischen Direktzugriffsspeicher (SDRAM) des Standes der Technik;
[0121] **Fig.** 102 ist eine Tabelle, die Anschlußnamen des SDRAM und ihre Funktionen darstellt; und
[0122] **Fig.** 103 ist ein Blockschaltbild, das eine Konfiguration eines integrierten Logik-DRAM des Standes der Technik darstellt.

BESCHREIBUNG DER BEVORZUGTEN AUSFÜHRUNGSFORMEN

[0123] Eine detaillierte Beschreibung wird von Ausführungsformen der vorliegenden Erfindung unter Bezugnahme auf die begleitenden Zeichnungen gegeben, in denen die gleichen Symbole in den Zeichnungen die gleichen oder ähnliche Bestandteile bezeichnen.

Erstes Beispiel

[0124] **Fig.** 1 ist ein Blockschaltbild, das eine Konfiguration einer integrierten Halbleiterschaltungsvorrichtung **1** eines ersten Beispieles der vorliegenden Erfindung darstellt.
[0125] Bezug nehmend auf **Fig.** 1 enthält eine integrierte Halbleiterschaltungsvorrichtung **1**: einen Anschluß **10**, der ein Steuersignal wie ein Steuersignal /RAS, /CAS, ..., /CS oder /WE empfängt; einen Anschluß **12**, der ein Adreßsignal ADD empfängt; einen Anschluß **14**, der ein Datensignal DATA empfängt; einen Anschluß **16**, der ein externes Taktsignal Ext.CLK empfängt; einen Schnittstellenabschnitt **2**, der Steuersignale an das Innere gemäß den Steuersignalen /RAS, /CAS, ..., /CS oder /WE, dem Adreßsignal ADD und dem Datensignal DATA ausgibt; einen Speicherabschnitt (DRAM) **4**, der eine Ausgabe des Schnittstellenabschnittes **2** zum Tätigwerden empfängt; ein Register **6**, das Daten und einen Befehl, die von dem Schnittstellenabschnitt **2** gegeben werden, oder ein Logikoperationsresultat hält; eine Logikschaltung **8**, die einen Betrieb wie Signalverarbeitung gemäß den Ausgaben von dem Register **6** und dem Schnittstellenabschnitt **2** durchführt; eine interne Takterzeugerschaltung **7** zum Erzeugen eines internen Taktsignals clkM für den Speicherabschnitt (DRAM) **4** und eines internen Taktsignal clkL für die Logikschaltung **8** gemäß dem externen Taktsignal Ext.CLK; und einen Schnittstellenabschnitt **9** zum Freigeben als Zwischenstufe der Datenlieferungs/Empfanges zwischen dem Register **6** und sowohl dem Speicherabschnitt **4** als auch dem Schnittstellenabschnitt **2**. Das Datenliefern/Empfangen zwischen dem Schnittstellenabschnitt **6**, dem Speicherabschnitt **4** und dem Schnittstellenabschnitt **9** wird durch einen internen Bus mbus durchgeführt.
[0126] Anschlüsse eines Chips **1** sind die gleichen wie Anschlüsse, die in einem Allzweck-DRAM benutzt werden. Folglich kann das gleiche Gehäuse wie das, in dem ein Allzweck-DRAM-Chip aufgenommen ist, verwendet werden. Zum Beispiel ist ein Gehäuse, in dem die integrierte Halbleiterspeichervorrichtung **1** aufgenommen ist, eines mit einer Stiftkonfiguration, wie sie in 101 gezeigt ist.
[0127] Aus diesem Grund wird, wenn eine integrierte Halbleiterschaltungsvorrichtung **1** der vorliegenden Erfindung in einer vorhandenen Anwendung angenommen wird, ein vorhandener Allzweck-DRAM einfach durch die integrierte Halbleitervorrichtungsschaltung **1** ersetzt, so daß keine Notwendigkeit auftritt, eine Baugruppe neu zu entwerfen oder einen zugeschnittenen Steuer-LSI zu entwickeln. Das heißt, da die integrierte Halbleiterschaltungsvorrichtung **1** stiftkompatibel mit dem Allzweck-DRAM ist, kann eine neue Funktion nur durch Ändern einer Software

hinzugefügt werden. Als neue Funktionen sind zum Beispiel Hinzufügen einer Schaltung zur Benutzung in einer Hochgeschwindigkeitsbildverarbeitung und eine Logikschaltung, die Verarbeitungen durchführt, die eine lange Zeit auf einem Mikrocomputer dauert, wie Verschlüsselungsverarbeitung denkbar. Weiterhin ist es erlaubt, daß Steuersignale unter Benutzung mehrerer nichtbenutzter Anschlüsse eingegeben werden können, z. B. ein NC (Nicht-verbindungs-)Stift wie der sechsunddreißigste Stift und der vierzigste Stift in **Fig.** 101 in einem Verpackungsgehäuse eines Allzweck-DRAM.

[0128] Als nächstes wird die Beschreibung eines konkreten Steuerverfahrens gegeben. Ein sogenanntes speicherabgebildetes I/O-Verfahren wird auf eine Steuerung einer integrierten Logikschaltung **8** angewendet.

[0129] **Fig.** 2 stellt ein Speicherabbild einer integrierten Logikhalbleiterspeichereinrichtung des ersten Beispieles dar.

[0130] Bezug nehmend auf **Fig.** 2 wird angenommen, daß eine Kapazität eines auf einem Chip hergestellten DRAM gleich 64 Mbit ist und eine Wortkonfiguration von 16 Bit ist. Adressen des DRAM enthalten eine X-Adresse, die von X0 bis X13 reicht, und eine Y-Adresse, die von Y0 bis Y7 reicht. Daher sind die Speicheradressen, die 8 Mbyte steuern, von 0h bis 3FFFFFh.

[0131] In dem Allzweck-DRAM können Daten in dem gesamten Adreßraum geschrieben und ausgelesen werden. Solch ein Raum, in den Daten geschrieben und ausgelesen sein können, wird ein DRAM-Raum als Definition genannt. Bei der vorliegenden Erfindung ist ein spezielles Gebiet in dem Adreßraum einem Logiksteuergebiet für die integrierte Logikschaltung zugeordnet. Zum Beispiel von 0h bis 1Fh in der Adresse dem Logiksteuergebiet zugeordnet. Eine Kapazität des Logiksteuergebietes beträgt z. B. 256 × 2 Byte = 512 Byte. Ein Befehl und ein Modus, die die Logikschaltung steuern, können gemäß den in den Adreßraum geschriebenen Daten ausgewählt werden.

[0132] Während in **Fig.** 2 ein Gebiet in der Seite der niedrigsten Adresse gesichert ist, kann das Logiksteuergebiet der höchsten Seite (3 FFFFFh bis 3FFFE0h) zugeordnet sein. Es ist auch erlaubt, daß, wenn der SDRAM als DRAM gedacht wird, der auf einem Chip integriert ist, ein Gebiet, in dem eine Adresse zugeordnet ist, nach dem Setzen eines Modusregisters ausgewählt werden kann. Weiterhin kann, solange das Logiksteuergebiet einer Einstellung eines Modusregisters zugeordnet ist, der SDRAM auch als ein normaler SDRAM von 64 Mbit benutzt werden.

[0133] **Fig.** 3 ist ein Blockschaltbild zum Beschreiben eines Weges, wie ein extern eingegebenes Signal zu einer Logikschaltung übertragen wird.

[0134] Bezug nehmend auf **Fig.** 3 enthält ein Schnittstellenabschnitt **2**: einen Puffer **3**, der die Steuersignale /RAS, /CAS, ..., /CS und /WE, die Adresse ADD und das Datensignal DATA empfängt; und eine Decodierschaltung **5**, die eine Ausgabe des Puffers **3** empfängt zum Decodieren der Ausgabe, worin das Register **6** Information wie einen Modus und einen Befehl als Reaktion auf eine Ausgabe der Decodierschaltung **5** hält, und die Logikschaltung **8** wird gemäß der in dem Register **6** gehaltenen Information gesteuert.

[0135] Während die Decodierschaltung **5** das Adreßsignal ADD und das Datensignal DATA decodiert, tritt ein Fall auf, in dem Daten, die in das Logiksteuergebiet geschrieben werden, das durch ein Adreßsignal spezifiziert ist, gehalten werden, wie sie ursprünglich in dem Register **6** sind.

[0136] Wenn das Register **6** aus einem SRAM (Statischer Direktzugriffsspeicher) oder ähnlichem aufgebaut ist, tritt ein Fall auf, in dem Daten in einem Gebiet des SRAM gehalten werden, das gemäß dem Adreßsignal ADD spezifiziert ist. Weiter ist es auch erlaubt, daß ein Gebiet, ein Teil des DRAM, als eine Halteschaltung anstelle des Registers **6** benutzt wird, und Daten zur Steuerung der Logikschaltung werden in dem Gebiet gehalten.

[0137] Das heißt, eine Konfiguration kann angenommen werden, bei der ein Adreßgebiet selbst, in dem der oben beschriebene Logiksteuerbereich zugeordnet ist, in dem Speicherabschnitt **4** zugeordnet ist. Alternativ kann eine Konfiguration auch angenommen werden, bei der die höchsten Bit virtuell zu einem Adreßraum des Speicherabschnittes **4** addiert werden, und ein Adreßgebiet, in dem ein Logiksteuergebiet zugeordnet ist, ein Gebiet wird, ein Teil des virtuellen Adreßraumes, der nicht ein Adreßraum eines Speicherzellenfeldes ist.

[0138] In der folgenden Beschreibung wird jedoch zur Vereinfachung der Beschreibung, obwohl es speziell nicht begrenzt ist, angenommen, daß eine solch einem Logiksteuergebiet zugeordnete Adresse dem Register **6** von **Fig.** 1 zuzuordnen ist.

[0139] Weiterhin ist in der folgenden Beschreibung zur Vereinfachung der Beschreibung eine Logikoperation in der Logikschaltung **8** eine Verschlüsselung, obwohl nicht speziell darauf begrenzt.

[0140] Es sei angenommen, daß die in **Fig.** 1 gezeigte integrierte Halbleiterschaltungsvorrichtung **1** drei Arten von Betriebsmodi aufweist, wie unten beschrieben wird, und eine der drei Arten von Betriebsmodi wird gemäß einer Kombination von Steuersignalen ausgewählt, die extern gegeben werden.

[0141] Wie in der unten gegebenen Beschreibung klar ist, wird in der Halbleiterspeichereinrichtung **1** mit den Betriebsmodi, wie sie unten beschrieben werden, indem ein richtiger Verarbeitungsmodus in Übereinstimmung mit einem eingebetteten System ausgewählt wird, das keinen Daten-Cache enthält, ein System, in dem ein Daten-Cache enthalten ist, oder ein System, in dem eine MMU (Memory Management Unit) unterstützt wird und das in einem virtuellen Adreßraum programmiert wird, eine Verarbeitungseffektivität der Verschlüsselung verbessert.

[0142] **Fig.** 4 ist ein Blockschaltbild zum Beschrei-

ben eines Betriebes in einem normalen Modus, einer der Betriebsmodi der integrierten Halbleiterschaltungsvorrichtung **1**.

[0143] In **Fig.** 4 wird eine Beschreibung eines Betriebes von Schreibzugriff in dem normalen Modus gegeben.

[0144] Wenn Schreibzugriff auf die integrierte Halbleiterschaltungsvorrichtung **1** während des Eintritts in einen Verschlüsselungsmodus gemäß einer Kombination von Steuersignalen durchgeführt wird, werden Schreibdaten verschlüsselt, und ein Resultat davon wird automatisch an einer im Schreiben spezifizierten Adresse gehalten. Das heißt, wenn Daten, von denen gewünscht wird, daß sie verschlüsselt werden, eingegeben werden ([1]), wird Verschlüsselung in der Logikschaltung **8** ([2]) durchgeführt, und ein Verschlüsselungsresultat wird zu einer Adresse in dem Schreibzugriff ([3]) übertragen.

[0145] **Fig.** 5 ist ein Blockschaltbild zum Beschreiben eines Betriebes von Lesezugriff in dem normalen Modus.

[0146] Wenn Lesezugriff auf die integrierte Halbleiterschaltungsvorrichtung **1** während des Eintritts in einen Verschlüsselungsmodus gemäß einer Kombination von Steuersignalen durchgeführt wird, werden Daten an einer Adresse, auf die zugegriffen wird, zu der Logikschaltung **8** von dem Speicherabschnitt **4** übertragen ([1]), Verschlüsselung wird in der Logikschaltung **8** durchgeführt ([2]), und ein Verschlüsselungsresultat wird zu der gleichen Adresse wie eine Adresse in dem Lesezugriff übertragen ([3]).

[0147] **Fig.** 6 ist ein Blockschaltbild zum Beschreiben eines Betriebes in einem Blockmodus, einem anderen der Betriebsmodi der integrierten Halbleiterschaltungsvorrichtung **1**.

[0148] In dem Blockmodus wird eine Blocklänge (Verarbeitungseinheit der Verschlüsselung) von Daten gemäß einer Kombination eines Steuersignals, eines Adreßsignals und anderer während des Setzens eines Verschlüsselungsmodus gesetzt.

[0149] Danach wird das Setzen einer Startadresse für die Verarbeitung durch Bewirken eines Blindschreibens (oder Blindlesens) an eine Adresse durchgeführt. Daten der gesetzten Blocklänge werden automatisch zu der Logikschaltung **8** von dem Speicherabschnitt **4** übertragen ([1]), wobei die Startadresse als ein Startpunkt dient, Verschlüsselung wird durchgeführt ([2]), und danach werden die Daten eines verarbeiteten Resultats wieder automatisch zu den Adressen über eine Blocklänge von der Startadresse als ein Startpunkt geschrieben ([3]). Eine Adreßzählerschaltung (nicht gezeigt) ist für die Datenübertragung der Daten wie eine Blocklänge vorgesehen. Die Adreßzählerschaltung kann funktional durch einen Adreßzähler ersetzt werden, der benutzt wird, wenn eine Selbstauffrischtätigkeit in dem DRAM durchgeführt wird.

[0150] **Fig.** 7 ist ein Blockschaltbild zum Beschreiben eines Betriebes in einem Puffermodus, ein noch anderer der Betriebsmodi der integrierten Halbleiterschaltungsvorrichtung **1**.

[0151] In dem Puffermodus wird eine Blocklänge von zu chiffrierenden Daten während des Einstellens des Verschlüsselungsmodus gesetzt. In dem Puffermodus wird ein vorgeschriebenes Adreßgebiet des Speicherabschnitts **4** als ein Pufferadreßgebiet **4b** gesichert. In dem Pufferadreßgebiet **4b** wird eine Mehrzahl von Puffergebieten gesetzt und Puffer-IDs werden den entsprechenden Puffergebieten zur Unterscheidung zugeordnet. Folglich wird in dem Puffermodus das Einstellen eines Puffer-ID einer Übertragungsbezeichnung eines Chiffrierungsresultats nach Setzen einer Blocklänge von zu chiffrierenden Daten durchgeführt.

[0152] Danach wird das Setzen einer Startadresse zum Verarbeiten durch Bewirken eines Blindschreibens (oder Blindlesens) einer Adresse in einem Verarbeitungsadreßgebiet **4a** durchgeführt.

[0153] Die Daten der gesetzten Blocklänge werden automatisch zu der Logikschaltung **8** von dem Speicherabschnitt **4** mit der Startadresse als ein Startpunkt übertragen ([1]), Verschlüsselung wird durchgeführt ([2]), und danach werden Daten eines Verarbeitungsresultats automatisch zu einem Puffergebiet einer spezifizierten Puffer-ID übertragen ([3]).

[0154] Es sei angemerkt, daß in einem Fall, in dem der Speicherabschnitt **4** aus einer Mehrzahl von Bänken aufgebaut ist, kein Konflikt zwischen Lesen und Schreiben auf einer Bank auftritt, wodurch eine effiziente Verarbeitung ermöglicht wird, wenn Puffergebiete gleichförmig den Bänken zugeordnet werden, und eine Bank, zu der das Verarbeitungsadreßgebiet **4a** gehört, und eine Bank, zu der ein Puffergebiet einer Übertragungsbezeichnung gehört, unterscheiden sich voneinander.

[0155] **Fig.** 8 ist ein Flußdiagramm zum Beschreiben von Tätigkeiten in den drei Modi, wie oben beschrieben wurde.

[0156] Bezug nehmend auf **Fig.** 8, wenn die Verarbeitung gestartet wird (Schritt S100), wird die Logikschaltung **8** durch Softwarerücksetzen zurückgesetzt, wenn eine Verschlüsselungsfunktion zum ersten Mal nach dem Einschalten benutzt wird (Schritt S102).

[0157] Aufeinander folgend wird, wie später im einzelnen ausgeführt wird, das Einstellen verschiedener Arten von Modi für die Verschlüsselung durchgeführt (Schritt S104): zum Beispiel Auswahl eines Verschlüsselungssystems mit Geheimschlüssel, Auswahl einer der drei Arten von Verschlüsselungsmodi, die oben beschrieben wurden, Bezeichnung einer Zahl von Puffern, wenn in dem Puffermodus.

[0158] Weiterhin wird ein Geheimschlüssel eingegeben (Schritt S106). Hierin wird, wenn ein Dreifach-DES-(Datenverschlüsselungsstandard)System ausgewählt ist, die Eingabe von zwei Arten von Schlüsseln benötigt.

[0159] Wenn eine Blockschiffrierung in einer Kette benutzt wird, wird ein anfänglicher Vektor eingegeben (Schritt S108).

[0160] In Aufeinanderfolge in einem Zustand, in dem der normale Modus ausgewählt ist, wenn die Dateneingabe durch ein Dateneingabebefehl gestartet wird (Schritt S110), wird Datenlesen über eine Verschlüsselungsblocklänge und ihre Verschlüsselung bei normalem Lesezugriff durchgeführt (Schritt S112), während andererseits Datenverschlüsselung über die Verschlüsselungsblocklänge und Schreiben eines Verarbeitungsresultats davon in normalem Schreibzugriff durchgeführt wird (Schritt S114). Das heißt, in einer Periode während ein Dateneingabebefehl eingegeben wird und Dateneingabe in dem normalen Lesen und normalen Schreiben durchgeführt wird, wird eine Mehrzahl von Speicherzugriffen der Verschlüsselungsblocklänge angenommen, und Verschlüsselung davon wird sequentiell in der Logikschaltung **8** durchgeführt.

[0161] Wenn ein Befehl eines Dateneingabestopps gegeben wird, hört die Dateneingabe auf (Schritt S116), und darauf folgend wird eine Flagprüfung in dem Register **6** extern durchgeführt (Schritt S138). Solange das Flag FL = "1", was "in Verschlüsselung" bezeichnet, kann kein Zugriff zu dem DRAM durchgeführt werden und der Zugriff zu dem DRAM wird zuerst freigegeben, nachdem FL = "0" hergestellt ist (Schritt S140).

[0162] Wenn andererseits der Blockmodus oder der Puffermodus spezifiziert ist, wird die Eingabe einer Blocklänge folgend dem Schritt S108 (Schritt S120) durchgeführt.

[0163] Darauf folgend wird in dem Fall des Blockmodus, wenn ein Dateneingabebefehl gegeben wird (Schritt S122), das Blindlesen oder Blindschreiben zum Spezifieren einer Startadresse durchgeführt (Schritt S124), und eine Verarbeitung in dem Blockmodus wird durchgeführt. Wenn ein Dateneingabestoppbefehl eingegeben wird (Schritt S126), dann bewegt sich die Verarbeitung zu Schritt S138.

[0164] In dem Puffermodus wird zuerst eine Puffer-ID eingegeben (Schritt S130). Wenn ein Dateneingabebefehl gegeben wird (Schritt S132), dann wird Blindlesen oder Blindschreiben zum Spezifizieren einer Startadresse bewirkt (Schritt S134), und die Verarbeitung in dem Puffermodus folgt. Wenn ein Dateneingabestoppbefehl eingegeben wird (Schritt S136), dann bewegt sich die Verarbeitung zu Schritt S138.

[0165] Durch Durchführen der oben beschriebenen Verarbeitung ist es möglich, drei Arten von Betriebsmodi zu steuern.

[0166] Bei der obigen Verarbeitung ist ein spezieller Modus vorhanden während einer Periode von einer Zeit, wenn ein Dateneingabestartbefehl eingegeben wird, bis zu einer Zeit, wenn ein Dateneingabestoppbefehl eingegeben wird. Wenn folglich der Zugriff extern auf einen Speicherraum eines DRAM durchgeführt wird, wird eine der drei Arten von Verarbeitungen, wie oben beschrieben wurde, auf zu verarbeitenden Daten durchgeführt.

[0167] Selbst wenn der Dateneingabestoppbefehl eingegeben wird, wird Information über einen Modus selbst in dem Register **6** gehalten; wenn daher der Dateneingabestartbefehl eingegeben wird, ist es möglich, die Verarbeitung in dem gleichen Verarbeitungsmodus durchzuführen.

[0168] In einem Fall, in dem eine Unterbrechung, die eine Auffrischtätigkeit anweist, von einer Speichersteuerung eines Systems während der Verschlüsselung in der Logikschaltung **8** gegeben wird, kann eine Verarbeitung wie unten beschrieben angenommen werden.

[0169] Zählen der Zahl von Auffrischbefehlen, die während der Verarbeitung in der Logikschaltung **8** eingegeben werden, wird durchgeführt, und die Information wird zum Beispiel in dem Register **6** gehalten. Wenn Verschlüsselung in der Logikschaltung **8** beendet ist, werden Auffrischverarbeitungen auf dem Speicherabschnitt **4** so häufig wie die Zahl der Zählungen durchgeführt. Das Flag FL bleibt in einem Zustand von "1" während der Auffrischperiode ähnlich zu einer Verarbeitungsperiode der Logikschaltung **8** in Hinblick auf die Auffrischperiode als eine von Kein-DRAM-Zugriff. Nachdem die Auffrischtätigkeit beendet ist, geht das Flag FL zu einem Zustand von "0" über.

[0170] Weiterhin nimmt in **Fig.** 8 nach Eingabe eines Dateneingabestartbefehls die integrierte Halbleiterschaltung **1** nur entweder einen Stoppbefehl oder einen Softwarerücksetzbefehl an.

[0171] Es wird angenommen, daß in dem Speicherabschnitt **4** ein Modusregister zum Spezifizieren eines Betriebsmodus davon zum Beispiel einer Burstlänge oder Dauer vorgesehen ist. Es wird weiter angenommen, daß eine Konfiguration angenommen ist, in der die Verarbeitung aus einem Steuermodus der Logikschaltung **8** derart herauskommen kann, daß die integrierte Halbleiterschaltung **1** einen normalen Betrieb fortsetzen kann unabhängig davon, in welchem Zustand die integrierte Halbleiterschaltung **1** bei dem Einschalten ist: nicht nur, wenn ein Softwarerücksetzbefehl gegeben ist, sondern auch wenn ein Modusregistereinstellbefehl an den Speicherabschnitt **4** gegeben ist.

[Inhalte von Verschlüsselung]

[0172] Eine einfache Beschreibung der in der Logikschaltung **8** durchgeführten Verschlüsselung wird im folgenden gegeben.

[0173] **Fig.** 9 ist eine Tabelle, die Beispiele von Kryptosystemen darstellt, die eine Logikschaltung **8** durchführen kann.

[0174] Bezugnehmend auf **Fig.** 9 unterstützt die Logikschaltung **8** RSA-(Rivest-Shamir-Adelman)Verschlüsselung als ein Kryptosystem mit öffentlichem Schlüssel und ein DES-System und ein Dreifach-DES-System als ein Kryptosystem mit geheimem Schlüssel.

[0175] Weiterhin werden in dem Kryptosystem mit geheimem Schlüssel Modi unterstützt wie ECB (Elek-

trisches Codebuch), CBC (Chiffrierungsblockketten), OFB (Ausgaberückkopplung), CFB (Chiffrierungsrückkopplung), die die hauptsächlichen Blockverschlüsselungsmodi sind. Der Logikschaltung **8** ist eine kritische Verarbeitung in der Verschlüsselung zum Verstärken der Geeignetheit der Anwendung zugeordnet, und andere Verarbeitungen werden der Softwarezentrierungstätigkeiten auf der Seite des Mikrocomputers des Systems unterworfen, das die integrierte Halbleiterschaltungsvorrichtung **1** steuert. Das größte Merkmal davon ist, daß Verschlüsselungssteuerung auf eine Weise kompatibel zu einem Allzweck-SDRAM realisiert werden kann.

[0176] Zum Beispiel wird in einem elektronischen Handelsmarkt die Aufstellung solch eine Kryptosystems, wie es oben beschrieben wurde, wichtig. Selbst bei einem drahtlosen Anwendungsprotokoll (WAP), dessen Anwendung auf ein tragbares Telefon erwartet wird, werden die oben beschriebenen Kryptosysteme unterstützt.

[0177] Zum Beispiel werden in Sicherheitsverarbeitung in solch einem Netzwerk Verarbeitungen wie Hash, Datencodierung und -füllen auf der Seite des Mikrocomputers des Systems verarbeitet. Im Gegensatz dazu kann die integrierte Halbleiterschaltungsvorrichtung Verarbeitungen durchführen, von denen gesagt werden kann, daß sie einen Hauptteil von Softwarezentrierungsverarbeitung in einer Praxis des Standes der Technik ist, wie Leistungsrestbetrieb, der bei elektronischer Authentifizierung gemäß RSA durchgeführt wird, Montgomery-Multiplikationsrestbetrieb und andere Restbetriebe. Wenn folglich eine integrierte Halbleiterschaltungsvorrichtung **1** nur kritische Verarbeitungen in dein System durchführt, kann Hochgeschwindigkeitsverarbeitung realisiert werden, während ein Freiheitsgrad auf der Anwendungsseite vergrößert wird.

[Kryptosystem mit geheimem Schlüssel]

[0178] In dem oben beschriebenen elektronischen Handel oder ähnlichem wird elektronische Authentifizierung in einem Kryptosystem mit öffentlichem Schlüssel durchgeführt, während Verschlüsselung von Datenübertragung/Empfang nach der Authentifizierung allgemein unter Benutzung eines Kryptosystems mit geheimem Schlüssel durchgeführt wird: Benutzung eines sogenannten Hybridsystems.

[0179] **Fig.** 10 ist eine konzeptuelle Zeichnung, die eine fundamentale Einheit für DES-Verschlüsselung darstellt, die als ein Kryptosystem mit geheimem Schlüssel benutzt wird.

[0180] Eine Schlüssellänge des DES ist 56 Bit, und 14 Bit eines einfachen Textes wird als 14 Bit eines Chiffretextes ausgegeben.

[0181] Andererseits sind **Fig.** 11 und 12 konzeptuelle Zeichnungen, die ein sogenanntes Dreifach-DES-Verarbeitungssystem darstellen.

[0182] Das Dreifach-DES enthält ein Kryptosystem von 112 Bit in Schlüssellänge, wie in **Fig.** 11 gezeigt ist, und ein Kryptosystem von 168 Bit in Schlüssellänge, wie in **Fig.** 12 gezeigt ist.

[0183] Bei dem in **Fig.** 11 gezeigten Dreifach-DES-112 wird ein einfacher Text von 64 Bit mit einem ersten 56 Bit-Schlüssel chiffriert, danach mit einem zweiten 56 Bit-Schlüssel dechiffriert, dann weiter mit dem ersten 56 Bit-Schlüssel chiffriert, und ein chiffriertes Resultat wird als Chiffretext von 64 Bit ausgegeben.

[0184] Bei dem Dreifach-DES-168, das in **Fig.** 12 gezeigt ist, wird ein einfacher Text von 64 Bit mit einem ersten 56 Bit-Schlüssel chiffriert, darauf folgend wird dieses mit einem zweiten 56 Bit-Schlüssel dechiffriert, dann weiter mit einem dritten 56 Bit-Schlüssel chiffriert, und ein chiffriertes Resultat wird als Chiffretext von 64 Bit ausgegeben.

[0185] **Fig.** 13 bis 15 sind konzeptuelle Zeichnungen, die die Entschlüsselung darstellen, die entsprechend zu den **Fig.** 10 bis 12 durchgeführt wird.

[0186] Wie in **Fig.** 13 bis 15 gezeigt ist, kann die Entschlüsselung eines Chiffretextes in einen einfachen Text in DES durchgeführt werden, indem absolut der gleiche Algorithmus wie die Chiffrierung von einem einfachen Text in ein Chiffretext benutzt wird.

[EBC-Modus]

[0187] **Fig.** 16 ist eine konzeptuelle Zeichnung zum Beschreiben einer Verschlüsselung in dem EBC-Modus. **Fig.** 17 ist eine konzeptuelle Zeichnung zum Darstellen von Verschlüsselung in dem EBC-Modus.

[0188] Bei der Verschlüsselung wird ein gewöhnlicher Text (einfacher Text) in 64 Bitblöcke Mi (M = M1, M2, M3, ...) unterteilt, und die Verschlüsselung wird auf jedem Block mit einem geheimen Datenschlüssel K durchgeführt, der gemeinsam einem Sender und einem Empfänger eigen ist. Indem das getan wird, werden Chiffretexte Ci (C = C, C2, C3, ...) von jeweils 64 Bit erzeugt.

[0189] Wie in **Fig.** 17 gezeigt ist, wird bei der Entschlüsselung ein Chiffretext Ci mit dem gleichen Schlüssel entschlüsselt, wie bei der Verschlüsselung benutzt wurde, wodurch einfache Texte Mi (M = M1, M2, M3, ...) erzeugt werden.

[CBC-Modus]

[0190] Eine einfache Beschreibung wird von dem CBC-Modus als ein Blockmodus in Kette unten gegeben.

[0191] In dem CBC-Modus wird ein Block Mi, der durch Unterteilen eines einfachen Textes in 64 Bit-Blöcke erhalten wird, zum Erhalten eines Chiffretextblockes Ce ähnlich zu dem oben beschriebenen EBC-Modus chiffriert, und weiter wird eine Exklusivlagiksumme zwischen dem Chiffretextblock Ci und dem nächsten Einfachtextblock Mi + 1 als Eingabe bei der nächsten Verschlüsselung benutzt. Solch ein Vorgang wird wiederholt, und eine Kette wird erstreckt, indem eins nach dem andern addiert wird,

wodurch eine Chiffre erzielt werden kann, die schwer zu knacken ist.

[0192] Andererseits wird Entschlüsselung derart durchgeführt, daß ein Chiffretextblock Ci ähnlich dem EBC-Modus entschlüsselt wird, so daß ein Resultat Mi erhalten wird, eine Exklusivlogiksumme zwischen Ci und einem entschlüsselten Resultat des nächsten Chiffretextblockes Ci + 1 wird als ein Ausgangseinfachtextblock Mi + 1 erzeugt, und solch ein Vorgang wird wiederholt, und eine Kette wird erstreckt, indem eins nach dem andern addiert wird, wodurch eine Entschlüsselung durchgeführt werden kann.

[0193] Wenn ein Einfachtextblock durch Mi bezeichnet ist, wird ein Chiffretextblock durch Ci (i = 1, 2, ...) bezeichnet, Verschlüsselung unter Benutzung eines kryptografischen Schlüssels K wird durch Ek bezeichnet, und Entschlüsselung wird durch Dk bezeichnet, durch Definition kann der CBC-Modus durch die unten gezeigten Logikausdrücke durchgeführt werden:

C1 = Ek(M1 + IV)

Ci = Ek(Mi + Ci – 1) (i: 2, 3, ...)

Mi = Dk(C1) + M1

Mi = Dk(Ci) + Ci – 1 (i: 2, 3, ...)

worin IV ein Anfangswert ist und in einem ersten Schritt sowohl bei der Verschlüsselung als auch bei der Entschlüsselung benutzt wird; ein Symbol + bedeutet eine Exklusivlogiksummenoperation; und eine Funktion Ek (...) bezeichnet eine Verschlüsselung, und eine Funktion Dk (...) bezeichnet eine Entschlüsselung.

[0194] Der Anfangswert (anfängliche Vektor) Iv ist der gleiche Wert auf der Seite der Verschlüsselung als auch auf der Seite der Entschlüsselung. Da ein Wert des Anfangswertes Iv einer dritten Partei bekannt sein kann, ist es nicht notwendig, daß der Anfangswert Iv zwischen einem Sender und einem Empfänger geheim gesendet wird. Zu dieser Zeit wird, wenn ein Wert des Anfangswertes Iv geändert wird, ein anderer Chiffretext aus der gleichen Nachricht erzeugt.

[0195] **Fig.** 18 ist eine konzeptuelle Zeichnung zum Beschreiben von Verschlüsselung in dem CBC-Modus.

[0196] Durch Durchführen der Verschlüsselung einer Exklusivlogiksumme zwischen dem Anfangswert IV und einem Einfachtextblock M1 wird ein entschlüsselter Block C1 erzeugt, und solch ein Vorgang wird sequentiell in Kette danach wiederholt.

[0197] Eine Einfachtextblocklänge, die zu einer Zeit an die integrierte Halbleiterschaltungsvorrichtung **1** eingegeben werden kann, wird jedoch durch eine Größe des Registers **6** bestimmt.

[0198] Wenn daher ein einfacher Text mit einer Länge länger als die Größe des Registers **6**, wird ein nächster Einfachtextblock mit dem unmittelbar vorangehenden Chiffretextblock (Ci in **Fig.** 18) als ein Anfangswert chiffriert.

[0199] **Fig.** 19 ist eine konzeptuelle Zeichnung, die Verarbeitung zum Dechiffrieren eines Chiffretextes darstellt, der auf die in **Fig.** 18 gezeigte Weise chiffriert ist.

[0200] Eine Verarbeitung wird in diesem Fall grundsätzlich umgekehrt zu der Verarbeitung von **Fig.** 18 durchgeführt. Weiter wird in einem Fall, in dem ein Chiffretext C länger als eine Größe des Registers **6** ist, eine Verarbeitung gekettet mit einem unmittelbar vorangehenden Chiffretextblock Ci als ein Anfangswert zu einem Zeitpunkt, zu dem der Chiffretext C die Größe des Registers **6** überschreitet.

[0201] **Fig.** 20 ist ein Zeitablaufdiagramm zum Beschreiben der in **Fig.** 8 beschriebenen Verarbeitung, wenn verschiedene Arten von Verschlüsselungsmodi vorhanden sind.

[0202] Ein charakteristischer Punkt in **Fig.** 20, der sich von einem gewöhnlichen DRAM im Zugriff auf das Register **6** unterscheidet, ist der, daß eine Verarbeitung durchgeführt wird, vorausgesetzt, daß eine Buslänge **1** ist, ohne Rücksicht auf das Setzen in dem Modusregister, das in den Speicherabschnitt **4** gesetzt ist.

[0203] In Hinblick auf die anderen Punkte kann ein Zugriff auf ein Register zu dem gleichen Zeitpunkt und in der gleichen Sequenz wie in einem DRAM durchgeführt werden.

[0204] Bei dem in **Fig.** 20 gezeigten Beispiel ist ein Fall gezeigt, in dem ein Adreßraum zum Zugriff auf das Steuerregister **6** gleich X = h3FFF ist.

[0205] Auf solche Weise wird, nachdem eine Adresse, auf die zuzugreifen ist, spezifiziert ist, ein Softwarerücksetzen zuerst durchgeführt. Darauf folgend wird das Setzen durchgeführt, ob DES-56 oder CBC-Modus zum Beispiel ausgewählt ist, als ein Kryptosystem mit geheimem Schlüssel in einer ersten Moduseinstellung.

[0206] Infolge wird eine Tätigkeit als eine zweite Moduseinstellung durchgeführt, wie eine Einstellung eines Verschlüsselungsmodus, Rücksetzen eines Adreßzählers eines Registers oder Bezeichnung, ob oder nicht ein Laden eines Anfangswertes durchgeführt wird, oder andere.

[0207] Weiter in Reihenfolge wird ein kryptografischer Schlüssel Keyl eingegeben, und ein Anfangswert IV wird ausgegeben.

[0208] Noch weiter in Reihenfolge wird ein einfacher Text mit 8 Byte als eine Einheit eingegeben, und wenn die Dateneingabe beendet ist, wird ein Dateiendsignal EOF eingegeben. Danach wird das Flag geprüft zum Bestätigen, ob oder nicht die Logikschaltung **8** während Verarbeitung ist.

[Details des normalen Betriebes]

[0209] **Fig.** 21 ist ein Zeitablaufdiagramm zum Beschreiben des Betriebes, wenn Daten auf die gleiche

Seite im Schreibzugriff des normalen Modus geschrieben werden.

[0210] Es sei angenommen, daß in **Fig.** 21 ein externes Taktsignal Ext.CLK zum Beispiel 100 MHz hat.

[0211] Weiterhin ist ein interner Takt ClkM zur Benutzung bei dem Betrieb des Speicherabschnittes **4** von 100 MHz synchron zu dem externen Taktsignal Ext.CLK.

[0212] Andererseits wird ein an die Logikschaltung **8** geliefertes Taktsignal clkL durch Frequenzteilen des externen Taktsignals Ext.CLK um zwei in der internen Takterzeugerschaltung **7** erzeugt.

[0213] Daher wird angenommen, daß eine Frequenz des internen Taktsignals clkL zum Beispiel 50 MHz beträgt.

[0214] **Fig.** 21 stellt einen Fall dar, in dem kontinuierlich gegebene Schreibbefehle WT an Adressen auf der gleichen Seite in dem normalen Schreiben ausgegeben werden.

[0215] In **Fig.** 21 ist ein Signal Ext.DQ [15:0] Daten von einer 16 Bit-Wortkonfiguration und werden an einen Daten-I/O-Anschluß 14 des Speicherabschnitts **4** gegeben.

[0216] Ein Signal Smbus [15:0] ist Daten auf einem internen Speicherdatenbus, der den Speicherabschnitt **4** und die Logikschaltung **8** dazwischen verbindet, wie in **Fig.** 1 gezeigt ist.

[0217] Ein Signal SifL [15:0] bezeichnet Daten in der internen Schnittstelle **9** und dem Register **6** zum Durchführen von Eingeben/Ausgeben von Daten auf einem internen Bus mbus oder der Logikschaltung **8**.

[0218] Das heißt, 16 Bit von Daten, die von dem Daten-I/O-Anschluß **14** durch das Register **6** und die interne Schnittstelle **9** gegeben sind, werden an die Logikschaltung **8** nach einer Seriell-Parallelumwandlung in 64 Bit-Daten gegeben, während Daten nach Verschlüsselung, die von der Logikschaltung **8** ausgegeben werden, parallel-seriell umwandelt werden in Daten jeweils von 16 Bit und danach an den DRAM **4** gegeben.

[0219] Ein Signal RdL ist ein Signal, das anzeigt, daß die Logikschaltung **8** die Verarbeitung gestartet hat und sich in einer Verarbeitungsperiode befindet, worin es in einer Periode, in der das Signal auf dem H-Pegel ist, anzeigt, daß die Logikschaltung **8** in Betrieb ist.

[0220] Bezug nehmend auf **Fig.** 21 müssen Daten, die extern durch den Daten-I/O-Anschluß **14** gegeben sind, gerade viermal geschrieben werden, da die Verschlüsselung mit 64 Bit als eine Einheit durchgeführt wird. Mit einer gegebenen Burstlänge von 4, wenn ein Schreibbefehl WT an die integrierte Halbleiterschaltungsvorrichtung **1** gegeben wird, werden dann Daten Da0 bis Da3 jeweils mit 16 Bit kontinuierlich von dem Daten-I/O-Anschluß **14** gegeben. Die Daten werden an die interne Schnittstelle **9** und das Register **6** durch den internen Bus Mbus gegeben, Daten von 64 Bit in der Länge werden in dem Register **6** gespeichert und danach werden die Daten an die Logikschaltung **8** als Daten DA zur Verschlüsselung gegeben. Zur gleichen Zeit wie dieses geht das Signal RdL auf den H-Pegel zum Starten des Betriebes der Logikschaltung **8**.

[0221] Wenn Daten Da' als ein Resultat der Verschlüsselung in der Logikschaltung **8** an das Register **6** ausgegeben werden, geht das Signal RdL auf den L-Pegel zum Beenden des Betriebes der Logikschaltung **8**.

[0222] Die an das Register **6** ausgegebenen Daten Da' sind parallelseriell umgewandelt und auf den internen Bus mbus von der internen Schnittstelle **9** als Daten Da'0 bis Da'3 jeweils von 16 Bit ausgegeben, die in den Speicherabschnitt **4** zu schreiben sind.

[0223] Ein Schreibbefehl WT, der danach gegeben wird, ist nötig zum Eingeben mit einem Intervall von 2tCLK × m + 8tCLK oder länger nach dem ersten Schreibbefehl, worin n die Zahl der Zyklen des internen Taktsignals clkL bedeutet, die für die Verschlüsselung von Daten von 64 Bit notwendig sind.

[0224] **Fig.** 22 ist ein Zeitablaufdiagramm zum Beschreiben des Betriebes, wenn 64 Bit-Daten auf eine unterschiedliche Seite in dem normalen Modus geschrieben werden.

[0225] In diesem Fall wird ein Schreiben-mit-Autovorladenbefehl als ein Schreibbefehl gegeben, worin eine Vorladetätigkeit automatisch durchgeführt wird, nachdem verschlüsselte Daten in den Speicherabschnitt **4** (DRAM) geschrieben sind.

[0226] Nachdem die Vorladetätigkeit beendet ist, wird ein Akt-Befehl ACT wiedergegeben, und danach wird eine ähnliche Verarbeitung wiederholt.

[0227] Wenn hierin ein einfacher Vorladebefehl an die integrierte Halbleiterschaltungsvorrichtung **1** anstelle des Schreiben-mit-Autovorladenbefehl gegeben wird, wird Verarbeitung wie unten beschrieben durchgeführt.

[0228] Das heißt, wenn ein Vorladebefehl PRE eingegeben wird, bevor verschlüsselte Daten in den Speicherabschnitt **4** geschrieben werden, wird das Vorladen automatisch nach Beendigung einer Schreibtätigkeit gestartet.

[0229] Wenn andererseits das Schreiben beendet ist, wird die Vorladetätigkeit bald zu irgendeiner Zeit gestartet.

[0230] **Fig.** 23 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, wenn Lesezugriff auf die gleiche Seite in dem normalen Modus durchgeführt wird.

[0231] In **Fig.** 23 wird ebenfalls angenommen, daß das externe Taktsignal Ext.CLK zum Beispiel 100 MHz ist.

[0232] Daher wird angenommen, daß das an den Speicherabschnitt **4** gegebene interne Taktsignal clkM ebenfalls 100 MHz ist, und daß das an die Logikschaltung **8** gegebene Taktsignal clkL gleich 50 MHz ist.

[0233] Bezug nehmend auf **Fig.** 23, wenn ein Lesebefehl RD gegeben wird, werden Lesedaten von dem Speicherabschnitt **4** auf den internen Datenbus mbus als Daten Da0 bis Da3 jeweils in 16 Bit in Reihenfolge

ausgegeben.

[0234] Daten DA von 64 Bit sind seriell-parallel umgewandelt durch die interne Schnittstelle **9** und das Register **6**. Die Daten DA werden in der Logikschaltung **8** chiffriert. Daten DA' nach der Verschlüsselung werden auf den internen Datenbus mbus durch das Register **6** und die interne Schnittstelle **9** als Daten Da'0 bis Da'3 jeweils von 16 Bit nach parallel-serieller Umwandlung ausgegeben. Die Daten Da'0 bis Da'3 nach der Seriell-Parallelwandlung werden in den Speicherabschnitt **4** geschrieben.

[0235] Eine Periode von der Zeit, zu der ein erster Lesebefehl RD gegeben wird, bis zu dem nächsten Lesebefehl RD = 2tCK × n + 8tCLK oder länger ähnlich der Schreibtätigkeit; der nächste Lesebefehl muß an solch einem Intervall oder länger eingegeben werden.

[0236] **Fig.** 24 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, wenn Lesezugriff auf eine andere Seite unter der gleichen Bedingung wie in **Fig.** 22 durchgeführt wird.

[0237] In diesem Fall ist die Verarbeitung ähnlich zu dem Fall der in **Fig.** 22 gezeigten Schreibtätigkeit, und Daten, die mit der Verschlüsselung fertig sind, werden in den Speicherabschnitt **4** geschrieben, was von einem automatischen Vorladen gefolgt wird.

[0238] **Fig.** 25 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes der integrierten Halbleiterschaltungsvorrichtung **1**, wenn ein externes Taktsignal Ext.CLK gleich 50 MHz ist.

[0239] In **Fig.** 25 ist ein Fall gezeigt, in dem Schreibzugriff auf Adressen auf der gleichen Seite in dem normalen Modus durchgeführt wird.

[0240] In diesem Fall ist das interne Taktsignal clkM zur Benutzung in dem Speicherabschnitt **4**, das von der internen Takterzeugerschaltung **7** erzeugt ist, von einer Frequenz von 100 MHz, die durch Multiplizieren des externen Taktsignals Ext.CLK zur Umwandlung erhalten wird.

[0241] Andererseits ist das interne Taktsignal clkK, das an die Logikschaltung **8** gegeben wird, von 50 MHz synchron zu dem externen Taktsignal.

[0242] Bei dem Betrieb von **Fig.** 25 werden ein Schreibbefehl WT und andere von außen mit einer Frequenz von 50 MHz synchron zu einer externen Taktfrequenz eingegeben, während ein Lesebetrieb von dem Speicherabschnitt **4** und ein Schreibbetrieb in den Speicherabschnitt **4** eines Verschlüsselungsresultats mit 100 MHz synchron zu dem internen Taktsignal clkM verarbeitet werden.

[0243] In diesem Fall wird eine Periode von tCLK × n + tCLK benötigt zum Sicherstellen einer Periode vom Ausgeben des ersten Schreibbefehls WT bis zu der Zeit des Ausgebens des nächsten Schreibbefehls WT.

[0244] **Fig.** 26 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebs, wenn Schreibzugriff auf eine andere Seite in dem normalen Modus in einem Fall durchgeführt wird, in dem das externe Taktsignal ext.CLK von 50 MHz ist.

[0245] In diesem Fall wird ebenfalls ein Schreiben-mit-Autovorladenbefehl als Schreibbefehl gegeben.

[0246] Daher wird eine Vorladetätigkeit durchgeführt, nachdem Daten DA', die in der Logikschaltung **8** verschlüsselt worden sind, parallel-seriell gewandelt sind und in den Speicherabschnitt **4** als Daten Da'0 bis Da'3 jeweils von 16 Bit geschrieben sind.

[0247] Eine Periode von der Zeit, zu der ein erster Zeitschreibbefehl WT gegeben wird, bis zu dem nächsten Schreibbefehl WT gegeben wird, wird gleich oder länger als eine Periode von tCLK × n + 8tCLK benötigt.

[0248] **Fig.** 27 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebs, wenn Lesezugriff auf die gleiche Seite in dem normalen Modus unter den gleichen Bedingungen in dem externen Taktsignal ext.CLK und den internen Taktsignalen clkM und clkL wie in **Fig.** 25 und 26 durchgeführt wird.

[0249] Weiterhin ist **Fig.** 28 ein Zeitablaufdiagramm zum Beschreiben eines Betriebs, wenn Lesezugriff auf eine unterschiedliche Seite in dem normalen Modus unter den gleichen Bedingungen wie für **Fig.** 27 durchgeführt wird.

[0250] In **Fig.** 27 und 28 werden ebenfalls mit Ausnahme, daß Befehle und andere synchron zu dem externen Taktsignal Ext.CLK gegeben werden, Tätigkeiten grundsätzlich ähnlich zu jenen in **Fig.** 23 und 24 durchgeführt.

[0251] In **Fig.** 27 muß jedoch eine Periode von der Zeit, wenn ein erster Zeitlesebefehl RD gegeben wird, bis zu der nächsten Zeit, zu der der Lesebefehl RD gegeben wird, gleich oder länger als tCLK × n + 4tCLK sein, und in **Fig.** 28 muß die Periode gleich oder länger als tCLK × n + 6tCLK sein.

[0252] **Fig.** 29 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebs, wenn das externe Taktsignal Ext.CLK von 25 MHz ist und eine Frequenzmultiplikationstätigkeit in der internen Takterzeugerschaltung **7** mit dem Resultat durchgeführt wird, daß das an den Speicherabschnitt **4** gegebene interne Taktsignal clkM zu 100 MHz umgewandelt ist und das an die Logikschaltung **8** gegebene interne Taktsignal clkL zu 50 MHz umgewandelt ist.

[0253] Externe Befehle werden synchron zu dem externen Taktsignal Ext.CLK ausgegeben, und interne Tätigkeiten werden ebenfalls synchron zu dem externen Taktsignal ext.CLK durchgeführt.

[0254] In **Fig.** 29 ist ein Betrieb in einem Fall gezeigt, in dem Schreibzugriff auf der gleichen Seite in dem normalen Modus durchgeführt wird.

[0255] In diesem Fall muß eine Periode von der Zeit, zu der ein erster Zeitschreibbefehl WT gegeben wird, bis dann, wenn der nächste Schreibbefehl gegeben wird, gleich oder länger als tCLK × n/2 + 4tCLK sein.

[0256] **Fig.** 30 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebs, wenn Lesezugriff auf die gleiche Seite in dem normalen Modus unter Bedingungen durchgeführt wird, daß das gleiche Taktsig-

nal wie in **Fig.** 29 benutzt wird.

[0257] Andererseits ist **Fig.** 31 ein Zeitablaufdiagramm zum Beschreiben eines Betriebs, wenn Lesezugriff auf eine andere Seite in dem normalen Modus unter Bedingungen durchgeführt wird, daß das gleiche Taktsignal wie in **Fig.** 30 benutzt wird.

[0258] Während in dem Fall von **Fig.** 30 eine Periode vom Ausgeben des ersten Zeitlesebefehls RD bis zu der Ausgabe des nächsten Lesebefehls RD gleich oder länger als tCLK × n/2 + 2tCLK sein muß, muß in dem Fall von **Fig.** 31 die Periode gleich oder länger als tCLK × n/2 + 3,5tCLK sein.

[0259] In dem oben beschriebenen normalen Modusbetrieb in dem Fall des Zugriffs zu der gleichen Seite, wenn Verschlüsselung beendet ist (einschließlich einer Schreibtätigkeit eines Verschlüsselungsresultats in den Speicherabschnitt **4**) sowohl bei der Lese- als auch Schreibtätigkeit, ist es notwendig, daß ein Stoppbefehl nach Eingeben eines Vorladebefehls eingegeben wird. Wenn ein Stoppbefehl nach Bestätigung eingegeben wird, daß die Verschlüsselung fertig ist, und zusätzlich keine Unterbrechung der Auffrischung während der Verschlüsselung auftritt, dann wird keine Bestätigung eines Flags FL in der in **Fig.** 8 beschriebenen Verarbeitung benötigt.

[0260] Wenn ein Stoppbefehl in einem Zustand einer offenen Seite eingegeben wird, wird Vorladen automatisch nach Beendigung der Verschlüsselung durchgeführt (einschließlich einer Schreibtätigkeit eines Verschlüsselungsresultates in den Speicherabschnitt **4**), worin eine Bestätigung eines Flags benötigt wird.

[0261] In jedem Fall wird die Beendigung der Verschlüsselung zu der Außenseite benachrichtigt durch Setzen von "0" des Flags, wenn die Verschlüsselung fertig ist.

[0262] Wenn eine Unterbrechung befohlen wird und ein Auffrischbefehl während der Verschlüsselung oder während des Schreibens eines Verschlüsselungsresultats in den Speicherabschnitt **4** gegeben wird, werden die Befehle angenommen, und die Zahl der angenommenen Auffrischbefehle wird gezählt. Das heißt, ein getrennter Zähler ist vorgesehen. Dann werden, nachdem eine Tätigkeit der Verschlüsselung beendet ist oder nachdem eine Schreibtätigkeit des Verschlüsselungsresultats in den Speicherabschnitt **4** beendet ist, Auffrischbefehle an den Speicherabschnitt **4** mit der gleichen Anzahl der gezählten Anzahl in Intervallen einer richtigen Zyklusperiode gegeben.

[0263] In diesem Fall müssen die Auffrischtätigkeiten nur durchgeführt werden, bis der Auffrischzähler in dem Speicherabschnitt 4 zum Anzeigen von 0 in dem Zählwert kommt, während der Zähler heruntergezählt wird.

[0264] Im allgemeinen wird ein Auffrischen eines CBR-Befehls durchgeführt, wenn ein Autoauffrischbefehl gegeben wird oder wenn der Speicherabschnitt **4** ein DRAM in dem EDU-Modus ist. Daher ist eine Adresse, an der Auffrischen durchgeführt wird, eine Adresse, die von dem getrennten Auffrischadreßzähler erzeugt wird. Nachdem die Auffrischtätigkeiten beendet sind, wird das in **Fig.** 8 beschriebene Flag FL von in "1" auf "0" geändert, wobei die Beendigung der Auffrischtätigkeiten als Beendigung der Verschlüsselung betrachtet werden.

[0265] Es sei angemerkt, daß nicht nur in dem Blockmodus sondern auch in dem Puffermodus Tätigkeiten grundsätzlich ähnlich zu den in dem Fall des normalen Modus sind, wie oben beschrieben wurde.

[0266] Was sich unterscheidet ist das, daß kontinuierliche Dateneingaben automatisch in der integrierten Halbleitervorrichtung **1** mit einer Startadresse als Referenz erzeugt werden.

[0267] Weiterhin wird in dem Puffermodus eines Falles, in dem der Speicherabschnitt **4** eine Mehrzahl von Bänken enthält, ein Puffer-ID, das einer Bank unterschiedlich von einer Bank zugeordnet ist, auf der das Lesen durchgeführt wird, ausgewählt und ein Verschlüsselungsresultat wird in das Puffergebiet geschrieben, das durch das ID zugeordnet ist.

[0268] Die Eingabe einer Startadresse bedeutet zum Beispiel das Spezifieren einer Zeilenadresse, wenn ein ACT-Befehl eingegeben wird, als ein Zugriff auf einen gewöhnlichen SDRAM, und dann Spezifizieren einer Spaltenadresse durch Eingeben derselben, wenn Blindlesen oder Blindschreiben durchgeführt wird.

[0269] Selbst wenn eine Burstlänge **1** oder länger ist, ist die Eingabe einer Startadresse für eine Adresse, zu der eingegeben wird, zuerst in einem Blindspaltenzugriff durchgeführt.

[0270] In dem Blockmodus und dem Puffermodus wird, nachdem ein Verschlüsselungsresultat vollständig in dem Speicherabschnitt **4** gespeichert ist und somit die Verschlüsselung vollständig ist, eine Bank, die aktiviert worden ist, automatisch vorgeladen.

[0271] In dem Fall werden ein Vorladebefehl und ein Autovorladebefehl, die während der Verschlüsselung eingegeben sind, ignoriert.

[0272] Weiterhin wird in dem Blockmodus und dem Puffermodus, sobald eine Startadresse gegeben ist, eine Spaltenzugriffstätigkeit ignoriert, selbst wenn der Befehl danach extern gegeben wird.

[0273] Weiterhin wird in dem Blockmodus und dem Puffermodus die Verschlüsselung automatisch über eine Blocklänge von der Startadresse durchgeführt. Zu dieser Zeit können die Tätigkeiten von nicht nur der Erhöhung einer Blocklänge sondern auch einer Verringerung einer Blocklänge möglich sein durch Datensetzen in das Register **6**.

[0274] Nebenbei, in dem Blockmodus und dem Puffermodus kann Datenschreiben als solches nicht nur in einem sequentiellen Modus sondern auch in einem verschachtelten Modus wie in einem SDRAM möglich sein.

[Beispiel 2]

[0275] **Fig.** 32 ist ein konzeptuelles Blockschaltbild,

das einen Weg darstellt, wie die integrierte Halbleiterschaltungsvorrichtung **1** der vorliegenden Erfindung und ein Mikroprozessor **90** verbunden sind.

[0276] Der Mikrocomputer **90** enthält: einen CPU-Kern **94**, eine Speichersteuerung **98** und eine externe Busschnittstellenschaltung **100**, worin die Bestandteile miteinander durch einen internen Bus **102** verbunden sind. Der CPU-Kern **94** ist mit einem Flash-Speicher **106** verbunden, in dem Daten, die zu chiffrieren oder zu dechiffrieren sind, durch eine serielle Schnittstelle **104** gespeichert werden.

[0277] Die externe Schnittstellenschaltung **100** gibt ein Steuersignal, ein Adreßsignal und Daten an die integrierte Halbleiterschaltungsvorrichtung **1** gemäß den Befehlen von dem CPU-Kern **94** aus.

[0278] **Fig.** 32 zeigt ein System, das für einen Betrieb im normalen Modus geeignet ist.

[0279] Das heißt, solch eine Konfiguration ist geeignet für ein System, in dem in dem normalen Modus die zu chiffrierenden oder zu dechiffrierenden Daten an den Mikrocomputer **90** von einer externen Einrichtung gegeben werden, die nicht der Speicherabschnitt **4** der integrierten Halbleiterspeichervorrichtung **1** ist, zum Beispiel der Flash-Speicher **106**.

[0280] Wenn Daten zeitweilig in dem Speicherabschnitt **4** gespeichert sind, die durch den Mikrocomputer **90** übertragen werden, werden die Daten in dem Speicherabschnitt **4** gehalten, nachdem sie automatisch zu einem kryptografischen Verarbeitungsresultat umgewandelt sind. Folglich kann die Zahl der Zugriffe auf den Speicherabschnitt **4**, die für die kryptografische Verarbeitung notwendig ist, verringert werden.

[0281] **Fig.** 33 ist ein konzeptuelles Blockschaltbild, das einen anderen Weg darstellt, wie die integrierte Halbleiterschaltungsvorrichtung **1** der vorliegenden Erfindung und der Mikroprozessor **90** verbunden sind.

[0282] Der Mikrocomputer **90** enthält: einen CPU-Kern **94**; einen Cache-Speicher **96**; eine Speichersteuerung **98**; und eine externe Busschnittstellenschaltung **100**, worin die Bestandteile miteinander durch einen internen Bus **102** verbunden sind. Die externe Busschnitttstellenschaltung **100** gibt ein Steuersignal, ein Adreßsignal und Daten an die integrierte Halbleiterschaltungsvorrichtung **1** gemäß einem Befehl von dem CPU-Kern **94** aus. Daher sind die externe Busschnittstellenschaltung **100** und ein integrierter Logik-DRAM **92** miteinander durch einen Steuersignalbus verbunden, der Steuersignale wie Signale /RAS, /CAS, ..., /CS überträgt, durch einen Adreßbus, der eine Adresse ADD überträgt, und durch einen Datenbus, der Daten DATA überträgt.

[0283] Zum Steuern der integrierten Halbleiterschaltungsvorrichtung **1** in solch einem System muß manchmal über Software nachgedacht werden, die auf dem Mikrocomputer **90** läuft.

[0284] **Fig.** 34 ist ein Flußdiagramm zum Beschreiben der Steuerung der integrierten Halbleiterschaltungsvorrichtung **1**.

[0285] Bezug nehmend auf **Fig.** 34 wird zuerst in einem Schritt S1 eine Adresse eines Logiksteuergebietes in einem reservierten Gebiet spezifiziert. Das heißt, dadurch wird ein Programm nicht einem Adreßraum für Befehlssteuerung einer Logikschaltung zugeordnet. Als ein Verfahren zum Verhindern, daß das Gebiet zugeordnet wird, ist eines beispielhaft, bei dem ein Logiksteuergebiet als ein reserviertes Gebiet geschützt wird, indem eine Funktion eines OS (Betriebssystem) benutzt wird.

[0286] Eine besondere Aufmerksamkeit muß selbst auf einen Einschaltzeitpunkt des OS gerichtet werden, so daß Kernroutinen selbst, die ein Herz eines OS sind, die grundlegende Steuerung des Systems wie Speicherverwaltung, Unterbrechungsverwaltung und Zwischenprozeßkommunikation durchführen, nicht dem Logiksteuergebiet zugeordnet werden. Folglich wird ein reserviertes Gebiet auf der Seite des OS mit spezieller Aufmerksamkeit spezifiziert, daß die Kernroutinen selbst nicht einem Logiksteuergebiet zugeordnet werden.

[0287] Dann werden in Schritt S2 mindestens ein Logiksteuergebiet und ein Gebiet, in dem zu chiffrierende oder zu dechiffrierende Daten gespeichert werden, als ein Gebiet ungeeignet für Cache **4a** in einem System mit einem Daten-Cache spezifiziert, wie in **Fig.** 33 gezeigt ist.

[0288] Das heißt, selbst in einem Fall, in dem Daten entsprechend einem Befehl von dem CPU-Kern **94** in **Fig.** 33 durch den internen Bus **102** zu der integrierten Halbleiterschaltungsvorrichtung **1** gesendet werden, die einen vorgeschriebenen Adreßraum spezifizieren, werden die Daten entsprechend dem Befehl in einen Cache-Speicher **96** geschrieben und nicht zu der integrierten Halbleiterschaltungsvorrichtung **1** übertragen, wenn der Cache-Speicher **96** tätig ist. In diesem Fall kann die in die integrierte Halbleiterschaltungsvorrichtung **1** integrierte Logikschaltung **8** nicht gemäß dem Befehlt tätig sein. Folglich ist es notwendig, derart einzustellen, daß das Logiksteuergebiet nicht dem Lache unterworfen wird. Dieses gilt für ein Gebiet, in dem zu chiffrierende Daten gespeichert werden, auf ähnliche Weise. In der Mehrzahl von Mikrocomputern ist die Steuerung möglich, daß ein Teil eines Adreßraumes für ein Gebiet ungeeignet für Cache spezifiziert wird.

[0289] Weiterhin in einem Fall, in dem eine Speicherverwaltungseinheitsfunktion zur Verfügung steht, ist es so eingestellt, daß ein virtueller Adreßraum nicht in dem Logiksteuerbereich benutzt wird.

[0290] Auf solche Weise wird in einem System wie eins, in dem ein Cache-Speicher. zur Verfügung steht, Initialisierung des Systems derart durchgeführt, daß auf mindestens ein Logiksteuergebiet in der integrierten Halbleiterschaltungsvorrichtung **1** ohne Fehler ohne Benutzung des Cache-Speichers zugegriffen wird.

[0291] Dann wird in Schritt S3 ein Befehl zur Logiksteuerung durch normales Schreiben in ein zugeordnetes Gebiet eingegeben, und in Schritt S4 kann eine

Prüfung des Verarbeitungszustandes in einer Logikschaltung und Lesen eines Verarbeitungsresultats durch normales Lesen durchgeführt werden. Weiterhin wird, wenn in Schritt S5 die Bearbeitung noch nicht beendet ist, die Verarbeitungen von Schritt S3 und S4 wiederholt. Um konkret zu sein, durch Prüfen eines Flags, das in ein Bit D1 einer Adresse Y = 0h geschrieben ist, kann ein Verarbeitungszustand beurteilt werden. Nach Prüfen des Flags FL zum Bestätigen der Beendigung der Verarbeitung kann der Mikrocomputer den nächsten Betrieb wie Zugriff auf ein Betriebsresultat starten.

[0292] Daher wird es möglich, während eine Beendigung einer Verarbeitung durch einen zugeordneten Stift zu einer Empfängerseite in der Praxis des Standes der Technik übertragen wird, gemäß der vorliegenden Erfindung, daß ein Flag-Zustand durch Durchführen eines gewöhnlichen normalen Lesens in dem SDRAM durchgeführt wird.

[0293] **Fig.** 35 ist ein konzeptuelles Blockschaltbild, das ein Beispiel eines Systems darstellt, das zur Anwendung auf einen Blockmodus der integrierten Halbleiterschaltungsvorrichtung **1** geeignet ist.

[0294] Das heißt, der Blockmodus ist geeignet für ein System der Art, in dem zu chiffrierende oder zu dechiffrierende Daten in einem Speicherabschnitt **4** (Hauptspeicher) der integrierten Halbleiterschaltungsvorrichtung zuvor gespeichert werden. Da kryptografische Verarbeitung ohne Ausgeben von Daten in einem Hauptspeicher auf einen externen Bus durchgeführt werden kann, ist es möglich, einen Hochgeschwindigkeitsbetrieb und einen niedrigen Leistungsverbrauch in einer verträglichen Weise zu erzielen.

[0295] **Fig.** 36 ist ein konzeptuelles Blockschaltbild, das eine Konfiguration darstellt, wenn die in dem Blockmodus tätige integrierte Halbleiterschaltungsvorrichtung **1** auf ein System angewendet wird, in dem ein Cache-Speicher **96** vorhanden ist.

[0296] In diesem Fall wird ein kryptografisches Verarbeitungsgebiet als ein mindestens nicht dem Cache unterliegenden Gebiet **4a** spezifiziert.

[0297] Wenn ein virtueller Speicheradreßraum benutzt wird, werden alle Datenblöcke, die zu chiffrieren oder zu dechiffrieren sind, innerhalb der gleichen Seite aufgenommen.

[0298] In einem Mikrocomputer jedoch mit einer Funktion des Löschens nur einer Zeile, die eine spezielle Adresse enthält, tritt keine Notwendigkeit auf, daß ein spezielles Gebiet als ein nicht dem Cache unterliegendes Gebiet bezeichnet wird.

[0299] Dann wird eine Beschreibung eines Systems gegeben, bei dem Zurückschreiben mit einem ausgestatteten Daten-Cache durchgeführt wird.

[0300] In einem Fall, in dem Daten, von denen gewünscht wird, daß sie chiffriert oder dechiffriert werden, in einem fache vor dem Eintritt eines kryptografischen Modus vorhanden sind, wird der Lache gelöscht und geräumt. Das heißt, Zurückschreiben wird zum Sperren einer Fahne durchgeführt. Dann geht das System in den kryptografischen Modus, und eine Startadresse wird eingegeben. Da eine Cache-Zeile mit der Startadresse in dem Daten-Cache zugeordnet ist, wird unmittelbares Räumen benötigt.

[0301] Der Ausdruck "Räumen" hierin bedeutet nur das Ungültigmachen einer Fahne ohne zurückzuschreiben.

[0302] **Fig.** 37 ist ein schematisches Blockschaltbild, das eine Konfiguration eines Systems zeigt, das geeignet ist, wenn ein Puffermodus der integrierten Halbleiterschaltungsvorrichtung **1** angenommen ist.

[0303] In dem Puffermodus ist es notwendig, daß ein Puffergebiet in dem nicht dem fache unterliegenden Gebiet **4a** vorhanden ist.

[0304] Andererseits können Daten selbst vor der kryptografischen Verarbeitung in dem Gebiet **4b** nicht dem fache unterliegend gespeichert werden.

[0305] Wenn ein virtueller Speicheradreßraum benutzt wird, müssen alle eines Datenblockes, die zu chiffrieren oder zu dechiffrieren sind, auf der gleichen Seite aufgenommen werden.

[0306] Wenn eine integrierte Halbleiterschaltung **1**, die sich auf die vorliegende Erfindung bezieht, wie oben beschrieben benutzt wird, kann die integrierte Halbleiterschaltung **1** richtig auf verschiedene Systeme angewendet werden zum Ermöglichen einer kryptografischen Hochgeschwindigkeitsverarbeitung mit niedrigem Leistungsverbrauch.

[0307] Weiter wird die Beschreibung eines Falles eines Rückschreibsystems mit einem Daten-Cache in dem Puffermodus gegeben.

[0308] Wenn zu chiffrierende oder zu dechiffrierende Daten in einem Lache vor dem kryptografischen Modus vorhanden sind, werden die Daten gelöscht und geräumt. Das heißt, Zurückschreiben wird zum Durchführen des Sperrens einer Fahne durchgeführt. Dann wird ein kryptografisches Verarbeitungsresultat in einem Puffergebiet gespeichert, das in dem Gebiet **4a** eingestellt ist, das nicht dem fache unterliegt. Während in dem Puffermodus wird eine Speicherbestimmung eines Verarbeitungsresultats durch eine Puffer-ID spezifiziert, in diesem Fall kann ein Verfahren angenommen werden, in dem eine Speicherbestimmung des Gebiets **4a**, das nicht dem Cashe unterliegt, mit einer Adresse spezifiziert wird.

[Drittes Beispiel]

[0309] **Fig.** 38 ist ein Blockschaltbild, das eine Konfiguration eines integrierten Logik-DRAM **30** eines dritten Beispiels darstellt, der durch Modifizieren der Konfiguration der integrierten Halbleiterschaltungsvorrichtung **1** des ersten Beispiels erhalten wird.

[0310] Bezug nehmend auf **Fig.** 38 enthält ein integrierter Logik-DRAM **30**: einen SDRAM-Abschnitt **32** und einen Logikabschnitt **34**.

[0311] Der SDRAM-Abschnitt **32** enthält: einen Schnittstellenabschnitt **36**, der ein externes Signal empfängt, zum Ausgeben eines Steuersignals gemäß dem externen Signal; und einen DRAM-Kern **38**,

der Datenhalten gemäß einer Ausgabe von dem Schnittstellenabschnitt **36** durchführt. Der Schnittstellenabschnitt **36** enthält: eine Steuersignaleingangsschaltung **40**, die Steuersignale /CS, /RAS, /CAS, /W und DQM empfängt; einen Taktpuffer **44**, der ein Taktsignal CLK und ein Taktfreigabesignal CKE empfängt, zum Erzeugen eines internen Taktes; einen Adreßpuffer **46**, der ein Adreßsignal A0 bis An synchron zu einer Ausgabe des Taktpuffers **44** fängt; und eine I/O-Schaltung **52**, die Eingabe/Ausgabe von Datensignalen DQ0 bis DQn synchron zu dem internen Takt durchführt. Es sei angemerkt, daß der Taktpuffer **44** eine Konfiguration mit einer internen Takterzeugerschaltung **7** ähnlich zu dem ersten Beispiel sein kann.

[0312] Der Schnittstellenabschnitt **36** enthält weiter: eine Steuerschaltung **42**, die Befehlssignale ACT und PRE und andere als Reaktion auf Ausgaben der Steuersignaleingangsschaltung **40** ausgibt; und einen Multiplexer **48**, der eine Ausgabe des Adreßpuffers **46** als eine S-Adresse und eine Y-Adresse gemäß einer Ausgabe der Steuerschaltung **42** multiplext.

[0313] Der Multiplexer **48** enthält: ein Modusregister **50**, das einen Modus gemäß einem Signalbit des Adreßsignals A0 bis Am auf einen MRS-Befehl des Modusregistereinstellens setzen kann.

[0314] Der DRAM-Kern **38** enthält: ein Speicherzellenfeld **54** mit Speicherzellen, die in einer Matrix von Zeilen und Spalten angeordnet sind; einen Zeilendecoder **56**, der eine Zeilenauswahl auf dem Speicherzellenfeld **54** gemäß einer Zeilenadresse durchführt, die von dem Multiplexer **48** gegeben wird; einen Spaltendecoder **58**, der eine Spaltenauswahl auf dem Speicherzellenfeld **54** gemäß einer Spaltenadresse durchführt, die von dem Multiplexer **58** gegeben wird; und einen Leseverstärkertreiber/ Schreibtreiber **60**, der Daten aus einer ausgewählten Speicherzelle ausliest und Daten in eine ausgewählte Speicherzelle schreibt.

[0315] Der Logikabschnitt **34** enthält: eine kryptografische Betriebslogik **74** und einen Registerabschnitt **72**, der Modusinformation zum Steuern der kryptografischen Betriebslogik **74**, Daten, die an die kryptografische Betriebslogik **74** eingegeben werden, und ein Betriebsresultat der kryptografischen Betriebslogik **74** als Reaktion auf eine Ausgabe des Schnittstellenabschnitts **36** hält.

[0316] Der Registerabschnitt **72** enthält: einen Selektor **76**, der aktiviert wird, wenn ein durch ein Adreßsignal A0 bis Am spezifiziertes Gebiet ein vorgeschriebener Wert zum Fangen eines Datensignals ist, das extern durch eine I/O-Schaltung **52** eingegeben ist; ein Steuerregister **78**, das Daten schreibt, die von außen durch den Selektor **76** gegeben sind; ein Modusregister **80**; ein Datenregister **84**; und ein Statusregister **82** und ein Datenregister **86**, die Daten halten, die von der kryptografischen Betriebslogik ausgegeben werden, zum Auslesen der gehaltenen Daten zu der Außenseite als Datensignale DQ0 bis DQn durch den Selektor **76** und die I/O-Schaltung **52**.

[0317] **Fig.** 39 ist eine Zeichnung, die ein Speicherabbild eines Systems darstellt, das auf den integrierten Logik-DRAM **30** des dritten Beispiels angewendet wird.

[0318] Bezug nehmend auf **Fig.** 39 entspricht ein externes RAM-Gebiet in einem Systemspeicherabbild einem integrierten Logik-DRAM. Der integrierte Logik-DRAM ist in ein Logiksteuergebiet und ein DRAM-Gebiet unterteilt, und eine enthaltene kryptografische Betriebslogik wird durch Zugriff auf das Logiksteuergebiet gesteuert. Ein Gebiet auf dem Systemspeicherabbild, das dem Logiksteuergebiet entspricht, wird als ein reserviertes Systemgebiet benutzt, und wenn ein Cache einer CPU und einer MMU (Speicherverwaltungseinheit) benutzt werden, wird es als Gebiet, das nicht dem Cache unterliegt, benutzt. Weiter wird Steuerung im voraus durch Firmware des Systems derart durchgeführt, daß ein Betriebssystem nicht in dieses Gebiet geladen wird. Weiterhin wird ebenfalls ein Anwendungsprogramm daran gehindert, dieses Gebiet zu benutzen.

[0319] Das Logiksteuergebiet wird zum Beispiel in einem Gebiet einer Zeilenadresse X = 3FFFh und einer Spaltenadresse Y = 0h bis FFh zugeordnet.

[0320] Das Steuerregister **78** von **Fig.** 38 wird zum Beispiel einer Adresse von X = 3FFFh und Y = 00h zugeordnet. Das Modusregister **80** wird einer Adresse von X = 3FFFh und Y = 01h zugeordnet. Das Statusregister **82** wird einer Adresse von X = 3FFFh und Y = 02h zugeordnet. Das erste Datenregister **84** wird einer Adresse von X = 3FFFh und Y = 03h zugeordnet, und das zweite Datenregister **86** wird einer Adresse von X = 3FFFh und Y = 04h zugeordnet.

[0321] Die kryptografische Betriebslogik **74** von **Fig.** 38 enthält einen Beschleuniger eines Hauptkryptosystems, das zum Herstellen von Sicherheit auf einem Netzwerk benutzt wird. Die kryptografische Betriebslogik **74** unterstützt Funktionen eines Kryptosystems mit öffentlichem Schlüssel, das bei elektronischer Authentifikation benutzt wird und ein Kryptosystem mit geheimem Schlüssel, das in der Datenübertrag/empfang nach der Authentifizierung benutzt wird. Da Verarbeitung in einer Logikschaltung, die einer Kryptografiktätigkeit zugewiesen ist, durchgeführt wird, kann das Verarbeiten mit niedrigerem Leistungsverbrauch und höherer Geschwindigkeit als eine Allzweck-CPU durchgeführt werden, wodurch sie für ein batteriegetriebenes System zum Beispiel geeignet ist.

[0322] Nun wird die Beschreibung davon gegeben, welche Zuordnungen zu entsprechenden Registern des in **Fig.** 39 gezeigten Logiksteuergebietes gemacht sind.

[0323] Dem Steuerregister **78** werden 16 Bit zugeordnet, D0 bis D15 der Y-Adresse 0h. Durch Schreiben eines Bit D0 mit 1 wird eine kryptografische Funktion zurückgesetzt. Das heißt, Verarbeitung wird durchgeführt, bei der ein Rücksetzpuls einer vorgeschriebenen Zeit an die kryptografische Betriebslogik

74 gegeben wird. Wenn ein Bit D1 gleich 1 ist, zeigt es an, daß die kryptografische Betriebslogik **74** in dem Vorgang der Verschlüsselung oder Entschlüsselung ist. Wenn daher auf die kryptografische Betriebslogik von außen zugegriffen wird, muß der Zugriff durchgeführt werden, nachdem ein Flag, das das Bit D1 anzeigt, zu 0 bestätigt ist.

[0324] Das Steuerregister **78** wird gemeinsam in einem Kryptosystem mit öffentlichem Schlüssel und einem Kryptosystem mit geheimem Schlüssel benutzt.

[0325] Als nächstes wird die Beschreibung von einigen Beispielen eines Registers gegeben, das in der Steuerung eines Kryptosystems mit geheimem Schlüssel benutzt wird.

[0326] Dem Modusregister **80** ist eine Adresse Y = 1h zugeordnet, worin Bit D1 und D0 von 16 Bit davon zur Auswahl eines Kryptosystems benutzt werden. Wenn die 2 Bit = "01" sind, dann ist das Kryptosystem DES, und wenn die 2 Bit "10" sind, dann ist das Kryptosystem dreifach-DES. Wenn die 2 Bit "00" sind, dann ist das Kryptosystem in einem Haltezustand.

[0327] Bit D5 bis D2 werden bei der Auswahl des Blockverschlüsselungsmodus benutzt. Wenn die Bit "0001" sind, dann wird ECB als der Blockverschlüsselungsmodus spezifiziert. Wenn "0010", dann wird CBC als der Blockverschlüsselungsmodus spezifiziert. Wenn "0100", dann wird OFB als der Blockverschlüsselungsmodus spezifiziert. Wenn "1000", dann wird CFB64 als der Blockverschlüsselungsmodus spezifiziert. Wenn "0000", dann ist der Blockverschlüsselungsmodus in einem Haltezustand.

[0328] Bit D8 bis D6 werden bei der Auswahl eines Datenverarbeitungsmodus benutzt. Wenn die Bit "001" sind, dann wird der normale Modus spezifiziert, wenn die Bit "010" sind, dann wird der Blockmodus spezifiziert, und wenn die Bit "100" sind, dann wird der Puffermodus spezifiziert, während, wenn die Bit "000" sind, befindet sich der Datenverarbeitungsmodus in dem Haltezustand.

[0329] Auf solch eine Weise kann ein Betriebsmodus, da 16 Bit einer Adresse von 2 Byte von Daten bei Y = 1h entsprechenden mehrfachen Modi zugeordnet werden können, selbst wenn eine Mehrzahl von Modi vorhanden ist, mit einem Einmalzugriff spezifiziert werden, wenn 2^{16}-Kombinationen effektiv benutzt werden.

[0330] Das Statusregister **82** wird einer Adresse von Y = 02h zugeordnet. Wenn zwei Bit D1 und D0 des Statusregisters "01" sind, bezeichnet diese Verschlüsselung, wenn "100", dann bezeichnet dies Entschlüsselung, während wenn "00", dann bezeichnet dieses eine Verarbeitung in dem Haltezustand. Wenn Bit D5 und D4 "01" sind, dann bezeichnet das einen Eingabestart eines einfachen Textes oder eines Chiffretextes, wenn "10", dann bezeichnet es einen Eingabestopp, während wenn "00", bezeichnet es die Verarbeitung in dem Haltezustand.

[0331] Bit D9 bis D6 bezeichnen eine Textlänge in einem Block eines jeden von OFB und CFB. Eine Adresse von Y = 3h bis 6h ist ein Gebiet, in dem ein DES-Schlüssel von 64 Bit und andere gespeichert sind. Ein Gebiet mit einer Adresse von Y = 7h bis Ah ist ein Gebiet, in dem ein in Dreifach-DES benutzter Schlüssel gespeichert ist.

[0332] Das erste Datenregister **84** ist, wie oben beschrieben wurde, ein Register zum Eingeben von Daten zu der Logikschaltung **74**, die zu chiffrieren oder zu dechiffrieren sind.

[0333] Das zweite Datenregister **86** ist, wie oben beschrieben wurde, ein Register zum Auslesen von chiffrierten oder dechiffrierten Daten aus der Logikschaltung **74**.

[0334] Während das erste und das zweite Datenregister **84** und **86** jeweils als ein Register von außen gesehen werden, besteht jedes tatsächlich aus einer Mehrzahl von Registern und ist eine Art von FIFO-Speicher.

[0335] **Fig.** 40 ist Zeichnungen, die Datenschreiben in ein erstes Datenregister **84** darstellen. Daten werden sequentiell zu Teilen (a) bis (c) von **Fig.** 40 in der Reihenfolge geschrieben. Obwohl es nicht in **Fig.** 38 gezeigt ist, ist ein Zähler zum Betreiben des Datenregisters **84** als FIFO vorgesehen, und der Zähler zählt eine Adresse des Registers.

[0336] **Fig.** 41 ist Zeichnungen, die Datenlesen von dem ersten Datenregister **84** darstellen. Daten werden sequentiell aus Teilen (a) bis (c) von **Fig.** 41 in der Reihenfolge ausgelesen.

[0337] Das zweite Datenregister **86** führt ebenfalls einen FIFO-Betrieb ähnlich zu dem des ersten Datenregisters **84** durch. Zusätzlich gibt es einige andere Register, obwohl sie nicht in **Fig.** 38 gezeigt sind: ein Register, das einen Anfangsvektor zur Verschlüsselung setzt, ein Register, das eine Blocklänge spezifiziert, ein Register, das eine Zahl von Puffern spezifiziert, und ein Register, wie es eine Puffer-ID bezeichnet.

[0338] Es sei angemerkt, daß zum Durchführen eines Kryptosystems mit öffentlichem Schlüssel, zum Beispiel eine kryptografische RSA-Verarbeitung ein Gebiet an einer Adresse von Y = 12h bis 1Fh als reserviertes Gebiet benutzt wird. Wie später beschrieben wird, wird in einem Fall, in dem ein Kryptosystem mit öffentlichem Schlüssel angewendet wird, ein kryptografisches Verarbeitungsresultat in einem enthaltenen Register gespeichert; daher kann der Zugriff zu dem DRAM-Gebiet durchgeführt werden selbst währen der kryptografischen Verarbeitung.

[0339] In einem Fall, in dem eine Zeilenadresse X, an der Lesen durch einen ACT-Befehl für SDRAM durchzuführen ist, gleich 3FFFh ist, erfaßt der Multiplexer **48** dieses zum Aktivieren des Selektors **76**. Dann wird eine Spaltenadresse Y durch einen Lesebefehl oder einen Schreibbefehl eingegeben, und dadurch wird die Auswahl durchgeführt, auf welches Register zuzugreifen ist. Danach werden extern eingegebene Daten durch die I/O-Schaltung **52** in ein Register geschrieben.

[0340] Während in dem Fall des dritten Beispiels ein Adreßgebiet, das als ein Logiksteuergebiet gesichert

ist, von 3FFF00h bis 3FFFFh reicht, ist es auch möglich, daß eine zugeordnete Adresse in einem Multiplexer gemäß Speicherinhalten des Registers **40** geändert wird, der durch ein Modusregistersetzbefehl gesetzt werden kann, und dadurch wird ein integrierter Logik-DRAM der vorliegenden Erfindung in verschiedene Arten von Mikrocomputern eingesetzt.

[0341] In einem Fall, in dem eine Adresse nicht durch einen Modusregistersetzbefehl zugeordnet wird, kann ein integrierter Logik-DRAM der vorliegenden Erfindung als ein gewöhnlicher 64 Mbit SDRAM benutzt werden. Ein Bit kann vorgesehen werden zum Spezifizieren, ob oder nicht eine intern eingesetzte Logikschaltung als ein Modusregister zum Anwenden auf einen gewöhnlichen SDRAM benutzt wird.

[0342] **Fig.** 42 ist ein Flußdiagramm zum Betreiben eines Betriebs des in **Fig.** 38 gezeigten integrierten Logik-DRAM **30**.

[0343] Bezug nehmend auf **Fig.** 42 wird zuerst, wenn die Verarbeitung startet (Schritt S200), eine kryptografische Betriebsschaltung zurückgesetzt (Schritt S202). Das heißt, die kryptografische Betriebsschaltung wird zurückgesetzt durch Zuerstschreiben einer logischen "1" in ein Rücksetzbit eines Steuerregisters vor der Benutzung der kryptografischen Betriebsschaltung.

[0344] Darauf folgend wird das Einstellen verschiedener Daten durchgeführt (Schritt S204). Zum Beispiel werden Auswahl von Verschlüsselung oder Entschlüsselung, Auswahl eines kryptografischen Modus, Eingabe eines geheimen Schlüssels und Eingabe eines Anfangsvektors durchgeführt. Wenn die Einstellung zuvor durchgeführt worden ist, kann die oben beschriebene Verarbeitung überschlagen werden.

[0345] Als nächstes wird ein Zähler in einem Register zurückgesetzt (Schritt S206). Das heißt, die Adreßzähler der entsprechenden Datenregister **84** und **86** werden durch Schreiben eines logischen "1" in Rücksetzbits in dem ersten und dem zweiten Register in dem Steuerregister zurückgesetzt.

[0346] Als nächstes wird eine Eingabe von Daten durchgeführt, von denen gewünscht wird, daß sie chiffriert oder dechiffriert werden (Schritt S208). Das heißt, ein Dateneingabestartbit wird gesetzt, und Daten, von denen gewünscht wird, daß sie chiffriert oder dechiffriert werden, werden kontinuierlich in das erste Datenregister **84** geschrieben. Die geschriebenen Daten werden sequentiell in das Datenregister **84** in dem FIFO-Modus gespeichert. Bei jedem Schreiben wird der Adreßzähler des Datenregisters **84** erhöht. Wenn die Dateneingabe fertig ist, wird ein Dateneingabestartbit gelöscht (oder ein Beendigungsbit wird gesetzt). Dadurch werden seriell geschriebene Daten in parallele Daten zum Vorbereiten der kryptografischen Verarbeitung umgewandelt.

[0347] Hierauf folgend startet der Betrieb in der Logikschaltung **84** (Schritt S210). Der Betrieb startet durch Schreiben eines logischen "1" in ein Betriebsstartbit des Steuerregisters **78**. Darauf folgend wird der Betrieb durchgeführt (Schritt S212). Während der Tätigkeit zeigt ein Besetztbit des Statusregisters **82** ein logisches "1" an. Es wird durch Prüfen des Besetztbit bestätigt, ob oder nicht der Betrieb läuft. Der logische Wert kann zum Beispiel als ein Signal RdL ausgelesen werden. Die in der kryptografischen Betriebslogik **74** verarbeiteten Daten werden in das zweite Datenregister **86** in einem FIFO-Modus gespeichert, wann immer die Daten verarbeitet sind. Jedes Mal, wenn Daten gespeichert werden, wird ein Adreßzähler für das zweite Register **86** erhöht.

[0348] Dadurch können Daten der kryptografischen Verarbeitung, die parallel ausgegeben werden, als serielle Daten ausgegeben werden.

[0349] Wenn die Tätigkeit beendet ist (Schritt S214), geht das Besetztbit des Statusregisters auf ein logisches "0". Darauf folgend wird das Auslesen des Betriebsresultats durchgeführt (Schritt S216). Wenn das Betriebsresultat ausgelesen ist, wird ein Adreßzähler für das zweite Datenregister **86** zurückgesetzt. Daten werden kontinuierlich aus dem Datenregister **86** ausgelesen. Jedes Mal, wenn Daten ausgelesen werden, wird der Adreßzähler für das zweite Datenregister **86** erhöht.

[0350] **Fig.** 43 ist ein Flußdiagramm zum Beschreiben eines anderen Betriebs des in **Fig.** 38 gezeigten integrierten Logik-DRAM **30**.

[0351] Bezug nehmend auf **Fig.** 43 ist die Verarbeitung bis zu Schritt S206 ähnlich zu der von **Fig.** 42.

[0352] In **Fig.** 43 startet folgend auf Schritt S206 ein Betrieb in der kryptografischen Betriebslogik **74** (Schritt S209). Der Betrieb startet durch Schreiben eines logischen "1" in ein Betriebsstartbit des Steuerregisters. Die kryptografische Betriebslogik **74** ist jedoch in einem Wartezustand, wenn das erste Datenregister **84** leer ist. Als nächstes wird Dateneingabe durchgeführt (S211). Ein Dateneingabestartbit wird gesetzt, und Daten, von denen gewünscht wird, daß sie chiffriert oder dechiffriert werden, werden kontinuierlich zu dem Datenregister **84** geschrieben. Geschriebene Daten werden sequentiell in das erste Datenregister **84** in einem FIFO-Modus gespeichert. Jedes Mal, wenn Daten geschrieben werden, wird der Adreßzähler für das erste Datenregister **84** erhöht. Wenn die Daten von 8 Byte in dem Datenregister **84** gesammelt sind, startet der Betrieb. Wenn die Dateneingabe beendet ist, wird ein Dateneingabestartbit gelöscht, oder ein Datenbeendigungsbit wird gesetzt.

[0353] Als nächstes fährt der Betrieb fort, der in der kryptografischen Betriebslogik **74** durchzuführen ist (Schritt S212). Während des Betriebs bezeichnet ein Besetztbit des Statusregisters **82** ein logisches "1". Durch Prüfen des Besetztbits kann bestätigt werden, ob der Betrieb läuft oder nicht. Daten, die in der kryptografischen Betriebslogik **74** verarbeitet werden, werden in das zweite Datenregister **86** in einem FIFO-Modus gespeichert, wann immer die Daten verarbeitet sind. Jedes Mal, wenn Daten gespeichert werden, wird der Adreßzähler für das zweite Daten-

register **86** erhöht.

[0354] Die Verarbeitung folgend auf die oben beschriebene Verarbeitung ist ähnlich zu der von **Fig.** 42; so daß die Beschreibung davon weggelassen wird.

[0355] Es sei angemerkt, daß in der oben dargestellten Beschreibung das erste und das zweite Datenregister **84** und **86** einfach aus einer Registerschaltung aufgebaut sind.

[0356] Solch ein Register kann jedoch auch aus einem statischen Direktzugriffsspeicher aufgebaut sein.

[0357] **Fig.** 44 ist ein Blockschaltbild, das eine Konfiguration darstellt, wenn das erste und das zweite Datenregister zur Benutzung zum Durchführen des Eingebens/Ausgebens von Daten auf der kryptografischen Betriebslogik **74** durch SRAMs dargestellt sind.

[0358] Mit einer Konfiguration, wie sie oben ebenfalls beschrieben wurde, kann eine Logikoperation wie Verschlüsselung mit hoher Geschwindigkeit gemäß einer Anforderung des Systems durchgeführt werden.

Vierte Ausführungsform

[0359] **Fig.** 45 ist ein schematisches Diagramm zum Beschreiben einer Konfiguration eines integrierten Logik-DRAM **130** eines vierten Beispiels der vorliegenden Erfindung.

[0360] Eine Konfiguration des integrierten Logik-DRAM **130** einer vierten Ausführungsform der vorliegenden Erfindung, wie er in **Fig.** 45 gezeigt ist, ist grundsätzlich ziemlich ähnlich zu dem integrierten Logik-DRAM **30** des in **Fig.** 38 gezeigten dritten Beispiels.

[0361] Zuerst einmal sind jedoch in dem integrierten Logik-DRAM **130** vier Bänke #0 bis #3 in einem Speicherzellenfeld **38** vorgesehen, und die Bänke sind so aufgebaut, daß sie unabhängig voneinander gelesen und beschrieben werden können.

[0362] Entsprechend solch einer Konfiguration sind Zeilendecoder **56.0** bis **56.3**, Spaltendecoder **58.0** bis **58.3** und Leseverstärker **60.0** bis **60.3** in den entsprechenden Bänken vorgesehen.

[0363] Weiterhin ist in **Fig.** 45 ein Steuersignaleingangsanschluß **11** neu vorgesehen, an welchen Anschluß ein Steuersignal CRYPT zum externen Befehlen einer kryptografischen Tätigkeit gegeben wird.

[0364] Weiterhin ist in dem integrierten Logik-DRAM **130** eine Steuerschaltung **42** explizit gezeigt als unterteilt in einen Adreßzähler zum automatischen Erzeugen einer internen Adresse in einem Auffrischbetrieb, einem Blockbetrieb, einem Puffermodusbetrieb und anderen; einen DRAM-Steuerabschnitt **42b** zum Steuern des Betriebs des DRAM gemäß einem Steuersignal und einem Adreßsignal; und einen Registerlogik-DRAM-Steuerabschnitt **42a** zum Steuern eines Registers, einer Logikschaltung, Liefern/Empfangen von Daten zwischen einer Logikschaltung und dem DRAM und anderen.

[0365] Es sei angemerkt, daß in **Fig.** 45 ein Register 0 (hier im folgenden als Reg0 zur Kürze ebenfalls genannt) gemeinsam für das Steuerregister **78**, das Modusregister **80** und das Statusregister **82** gezeigt ist, die in **Fig.** 38 gezeigt sind; ein Zähler **85** ist explizit für ein erstes Datenregister **84** gezeigt (hier im folgenden als Reg1 zur Kürze ebenfalls genannt) und ein zweiter Adreßzähler **87** ist explizit für ein zweites Datenregister **86** (hier im folgenden Reg2 zur Kürze ebenfalls genannt) gezeigt.

[0366] In dem integrierten Logik-DRAM **130** eines vierten Beispiels, der in **Fig.** 45 ebenfalls gezeigt ist, ist ein Modusregister **50** zum Halten eines Parameters eines Modusregisters gesetzt, der ein Steuerbefehl für den DRAM ist. Das Modusregister **50** kann nicht nur das Modussetzen für den DRAM ausführen, sondern auch das Setzen einer Zugriffsfreigabe oder -sperrung für das Register Reg0, das erste Datenregister **84** und das zweite Datenregister **86**. Wenn ein Modusregistersetzen eingegeben wird, werden das Steuerregister und die kryptografische Betriebsschaltung **74** zurückgesetzt.

[0367] Weiterhin ist in dem in **Fig.** 45 gezeigten integrierten Logik-DRAM **130** ebenfalls das Register Reg0 ein Register zum Steuern eines Befehls zum Steuern der kryptografischen Betriebsschaltung **74** und zum Steuern eines Modus, das erste Datenregister Reg1 ist ein Register zum Halten von Eingabedaten für die kryptografische Betriebslogik, und das zweite Datenregister Reg2 ist ein Register zum Halten eines Ausgangsresultats der kryptografischen Betriebslogik.

[0368] Da die anderen Punkte in dem Aufbau ähnlich zu entsprechenden Punkten im Aufbau des integrierten Logik-DRAM **30** des in **Fig.** 38 gezeigten dritten Beispiels sind; sind die gleichen Bezugszeichen an den gleichen Bauteilen angebracht, und die Beschreibung davon wird weggelassen.

[Register-Registerbetrieb]

[0369] Als nächstes wird die Beschreibung eines Betriebs des integrierten Logik-DRAM **130** des in **Fig.** 45 gezeigten vierten Beispiels gegeben.

[0370] In dem ersten Beispiel und anderen sind Konfigurationen und Betriebe derart, daß Daten, die kryptografisch zu verarbeiten sind, an die Logikschaltung **74** von außen oder von dem Speichezellenfeld gegeben werden, und die Daten nach der kryptografischen Verarbeitung in das Speicherzellenfeld wieder geschrieben werden.

[0371] Wie jedoch in **Fig.** 45 gezeigt ist, wenn eine Konfiguration angenommen wird, in der zwei Register **85** und **86** zum Dateneingeben/Ausgeben auf der Logik **74** vorgesehen sind, kann der folgende Betrieb (ein Register-Registerbetrieb) durchgeführt werden.

[0372] **Fig.** 46 ist ein konzeptuelles Blockschaltbild zum Beschreiben solch eines Register-Registerbetriebs.

[0373] Zuerst wird durch Eingeben eines Steuersignals Daten in das Register 0 zum Durchführen des Setzens des Schreibmodus geschrieben ([1]).

[0374] Darauf folgend werden zu chiffrierende oder zu dechiffrierende Daten in das erste Register **84** durch den Daten-I/O-Anschluß **14** von der Außenseite geschrieben ([2]).

[0375] Wenn Daten einer Datenblocklänge, das heißt Daten von 8 Byte für kryptografische Verarbeitung eingegeben sind, startet die Verarbeitung der kryptografischen Betriebslogik **74** ([3]). Darauf folgend wird jedes Mal, wenn die Verarbeitung für Daten von 8 Byte beendet ist, ein Verarbeitungsresultat in das zweite Register **86** geschrieben ((4]).

[0376] Zugriff von der Außenseite kann auf die Bänke 0 bis 3 durchgeführt werden, während solch eine Verarbeitung der kryptografischen Betriebslogikschaltung **74** durchgeführt wird.

[0377] Darauf folgend wird bestätigt, daß das Flag FL in dem Register 0 gleich 0 ist, Daten werden zu der Außenseite von dem zweiten Register **86** durch den Daten-I/O-Anschluß **14** ausgegeben ([5]).

[0378] **Fig.** 47 ist ein Flußdiagramm zum Beschreiben eines Betriebs eines solchen integrierten Logik-DRAM **130** in einer detaillierteren Weise.

[0379] Zu allererst wird die Leistung eingeschaltet (Schritt S300) und die Initialisierung des DRAM wird durchgeführt (Schritt S302).

[0380] Darauf folgend wird ein Signal CRYPT, das an den Steuersignaleingangsanschluß **11** eingegeben wird, auf den H-Pegel angehoben, und dadurch wird das Datenschreiben zu dem Register 0 möglich.

[0381] Wenn dann eine kryptografische Funktion zum ersten Mal benutzt wird, nachdem eingeschaltet wurde, wird ein Softwarerücksetzen durchgeführt (Schritt S306).

[0382] Weiterhin wird das Setzen verschiedener Modi durchgeführt (Schritt S308). Zum Beispiel werden Auswahl eines Kryptosystems mit geheimem Schlüssel, Auswahl, ob eine Schlüsseleingabe durchgeführt ist oder nicht, Auswahl über einen kryptografischen Bearbeitungsmodus und andere durchgeführt.

[0383] Noch weiterhin wird ein anfänglicher Vektor IV bei Notwendigkeit eingegeben (Schritt S310).

[0384] Dann wird ein geheimer Schlüssel eingegeben (Schritt S312).

[0385] Weiterhin bewegt sich die Verarbeitung zur Auswahl, ob oder nicht eine anfängliche Eingabe durchgeführt ist, und zur Auswahl, ob Verschlüsselung oder Entschlüsselung durchgeführt wird (Schritt S314), und hierauf folgend, ob die anfängliche Eingabe auszuführen ist, dann werden die zu verarbeitenden Daten in das erste Register Regl eingegeben (Schritt S316).

[0386] Wenn die oben beschriebene Initialisierung beendet ist, wird gewöhnlich ein Eingangsstartbefehl zuerst gegeben (Schritt S318). Zu dieser Zeit ist das Flag FL in dem Register 0 auf "1" gesetzt.

[0387] Darauf folgend werden zu verarbeitende Daten zu dem ersten Register **84** eingegeben (Schritt S320). Wenn die Eingabe von Daten von 8 Byte in der Länge beendet ist, wird eine Verschlüsselung/Entschlüsselung gestartet. Ein Verarbeitungsresultat wird in das zweite Register **86** geschrieben, wann immer das Verarbeitungsresultat erhalten ist. Wenn das erste Register leer wird, geht die Verarbeitung in einen Wartezustand.

[0388] Als nächstes wird, wenn ein Eingabestoppbefehl eingegeben wird (Schritt S322), dann die Flagprüfung durchgeführt (Schritt S324). Wenn die Verarbeitung richtig beendet ist, wird das Flag FL zu "0", daher werden, wenn dieses bestätigt ist, Daten zu der Außenseite zu dem zweiten Register **86** durch den Daten-I/O-Anschluß **14** ausgelesen (Schritt S326).

[0389] Die folgende Verarbeitung ist eine Wiederholung der oben beschriebenen Verarbeitung.

[0390] Es sei angemerkt, daß die Tätigkeit in Schritten S314 und S318 auch simultan durchgeführt werden, indem ein Wert des Zählers **85** richtig zurückgesetzt wird.

[0391] **Fig.** 48 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebs des integrierten Logik-DRAM **130** in dem in **Fig.** 47 gezeigten Verarbeitungsfluß.

[0392] Mit der Ausnahme einer Periode, in der Datenschreiben in das erste Register **84** durchgeführt wird, ist der Zugriff zu dem DRAM möglich, selbst wenn die kryptografische Betriebslogikschaltung **74** in Betrieb ist.

[0393] Jedes Mal, wenn die Verarbeitung in der kryptografischen Tätigkeit beendet ist, werden Daten sequentiell in das zweite Register **86** geschrieben.

[0394] Während einer Periode, während der Daten aus dem zweiten Register zu der Außenseite ausgelesen werden, ist der Zugriff zu dem DRAM unmöglich.

[0395] Durch Durchführen der Verarbeitung, wie sie oben beschrieben wurde, ist der Zugriff zu dem DRAM möglich zu jeder Zeit, selbst wenn die kryptografische Betriebslogikschaltung **74** in Betrieb ist, solange kein externer Zugriff auf das Register durchgeführt wird.

[0396] Daher tritt kein Problem auf, selbst wenn eine Unterbrechung während der kryptografischen Verarbeitung auftritt, und Daten schreiben oder lesen kann in dem DRAM während der kryptografischen Verarbeitung durchgeführt werden.

[0397] In diesem Fall funktionieren das erste und das zweite Register **84** und **86** als ein FIFO einer Breite von 8 Bit (512 Stufen). Wenn Datenschreiben die letzte Stufe erreicht, kehrt der Betrieb zu der ersten Stufe zurück und Überschreiben wird dort durchgeführt; daher muß ein Verarbeitungsresultat aus einem Register vor dem Überschreiben ausgelesen werden.

[Register-DRAM-Betriebsmodus]

[0398] **Fig.** 49 ist ein konzeptuelles Blockschaltbild zum Beschreiben eines anderen Betriebs [Register-DRAM-Betriebsmodus] des in **Fig.** 45 gezeigten integrierten Logik-DRAM **130**.

[0399] In **Fig.** 46 ist die Konfiguration derart, daß die in der kryptografischen Betriebslogikschaltung **74** verarbeiteten Daten zu der Außenseite zu den Daten-I/O-Anschluß **14** ausgelesen werden.

[0400] Hier ist eine Konfiguration möglich, bei der in dem kryptografischen Betrieb verarbeitete Daten nicht zu der Außenseite ausgelesen werden sondern in ein Speicherzellenfeld des DRAM-Abschnitts geschrieben werden.

[0401] Bei dieser Konfiguration wird eine spezifische Bank, zum Beispiel die Bank **3**, als eine Bank für solch ein Datenschreiben im voraus ausgewählt.

[0402] Wenn solch eine spezifische Bank ausgewählt ist, kann der DRAM-Abschnitt für eine Unterbrechung von einer Speichersteuerung oder ähnliches zu einer anderen Bank ausgelegt werden.

[0403] Bezug nehmend auf **Fig.** 49 wird zuerst ein Modussetzen durch Schreiben von Daten in das Register 0 durchgeführt ([1]). Darauf werden zu chiffrierende oder zu dechiffrierende Daten in das erste Register **84** geschrieben ([2]).

[0404] Wenn die in das erste Register **84** eingegebenen Daten 8 Byte betragen, startet eine kryptografische Verarbeitung in der kryptografischen Betriebslogikschaltung **74** ([3]).

[0405] Nachdem die Verarbeitung beendet ist, wird das Datenschreiben von 8 Byte als eine Einheit in das zweite Datenregister **86** durchgeführt ([4]).

[0406] In einer Periode von der Dateneingabe zu dem Register **1** zu der Dateneingabe in das Register **2** kann auf die Bänke 0 bis 3 des DRAM-Abschnitts zugegriffen werden.

[0407] Hierauf folgend wird bestätigt, daß das Flag FL "0" ist, und danach wird der Eintritt in den Register-DRAM-Übertragungsmodus durchgeführt ([5]).

[0408] Wenn solch ein Eintritt durchgeführt wird, wird das Datenschreiben zu einer Registerübertragungsbestimmungsbank (z. B. die Bank 3) durchgeführt ([6]).

[0409] Wenn der Zähler für das zweite Datenregister **86** zurückgesetzt wird, wird Schreiben startend an der ersten Stufe durchgeführt, und wenn es nicht zurückgesetzt ist, startet Schreiben wiederum an einer Stufe irgendwo zwischen der ersten und der letzten Stufe.

[0410] In diesem Fall wird die Datenübertragung durch Schreiben durchgeführt, wobei auf eine Adresse zugegriffen wird, zu der gewünscht wird, daß Daten übertragen werden.

[0411] Wenn die oben beschriebene Datenübertragung beendet ist, wird das Verlassen aus dem Register-DRAM-Übertragungsmodus durchgeführt ([7]).

[0412] Während der oben beschriebenen Verarbeitung kann auf eine Bank, die nicht als eine Registerübertragungsbestimmung spezifiziert ist, ähnlich zugegriffen werden, wie unter Bezugnahme auf **Fig.** 46 beschrieben worden ist.

[0413] **Fig.** 50 ist eine konzeptuelle Zeichnung zum Beschreiben eines Konzeptes von Tätigkeiten des ersten und des zweiten Registers **84** und **86** und der Zähler **85** und **87** zum Ermöglichen der oben beschriebenen Konfiguration.

[0414] Der erste Zähler **85** zählt eine Position, an der das Datenschreiben beendet ist, als eine Zählung CT1, während eine Position gezählt wird, an der die Eingabe zu der Logikschaltung **74** vorangeschritten ist, als eine Zählung CT2.

[0415] Andererseits zählt der zweite Zähler eine Position, an der das Schreiben eines Verarbeitungsresultats vorangeschritten ist, als eine Zählung CT3.

[0416] In dem ersten Register **84** wird die Verarbeitung ermöglicht fortzufahren, bis CT2 < CT1 und eine Schreibtätigkeit in dem ersten Register **84** höher in der Priorität als eine Lesetätigkeit darauf ist.

[0417] **Fig.** 51 ist ein Flußdiagramm zum Beschreiben in mehr Einzelheiten der unter Bezugnahme auf **Fig.** 49 beschriebenen Tätigkeit.

[0418] Bezug nehmend auf **Fig.** 51 ist die Verarbeitung bis zu Schritt S324 grundsätzlich ähnlich zu der in **Fig.** 47 gezeigten.

[0419] Wenn danach bestätigt wird, daß das Flag "0" ist, wird der Eintritt in den Register-DRAM-Übertragungsmodus durchgeführt (Schritt S330).

[0420] Hierauf folgend wird der Adreßzähler **87** des zweiten Registers **86** zurückgesetzt, und ein Schreibbefehl für den DRAM wird eingegeben (Schritt S334).

[0421] Als Reaktion darauf wird die Datenübertragung von dem zweiten Register **86** zu der Bank 3 durchgeführt.

[0422] Dann wird ein Befehl gegeben zum Verlassen des Register-DRAM-Übertragungsmodus (Schritt S336).

[0423] Darauf folgend kehrt die Verarbeitung zu einem der Schritte S314, S318 und S330 gemäß der Bezeichnung durch ein Steuersignal zurück.

[0424] Durch Durchführen der oben beschriebenen Verarbeitung kann ein Verarbeitungsresultat von dem zweiten Register **86** zu dem DRAM übertragen werden nach Beendigung des Betriebs zwischen den Registern.

[0425] Zu dieser Zeit werden Daten des Registers **2** durch Geben eines Schreibbefehls an den DRAM zu einer Zugriffsadresse in dem DRAM übertragen.

[0426] In dieser Situation wird die Sperrung von Daten an den externen Daten-I/O-Anschluß **14** parallel zu der Datenübertragung gegeben.

[0427] Wenn weiterhin Daten, die von einer Adresse gelesen sind, auf die zugegriffen ist, zu dem zweiten Register **86** übertragen werden sollen und zu der Außenseite ausgelesen werden sollen, durch den Lesezugriff des DRAM ausgelesen werden, kann eine Konfiguration ebenfalls angenommen werden, bei der Daten aus dem zweiten Register **86** durch den Daten-I/O-Anschluß **14** ausgelesen werden (externer

Bus während der Periode des Register-DRAM-Übertragungsmodus).

[0428] Während einer Periode des Register-DRAM-Übertragungsmodus in dem Register-DRAM-Betriebsmodus, wie oben beschrieben wurde oder einem DRAM-Registerbetriebsmodus, der später beschrieben wird, wird die Datenübertragung zwischen den Registern **84**, **86** und dem DRAM-Abschnitt durch den internen Bus mbus innerhalb des integrierten Logik-DRAM **130** Chip durch Zugriff des integrierten Logik-DRAM **130** durchgeführt.

[0429] **Fig.** 52 ist ein Blockschaltbild, das einen Zustand eines externen Datenbusses darstellt, wenn solche integrierten Logik-DRAMs **130a** und **130b** mit einer Mikrosteuereinheit MCU durch den externen Bus ext.bus verbunden sind.

[0430] Wie in **Fig.** 52 gezeigt ist, gibt es in dem Register-DRAM-Übertragungsmodus keine Chance, daß Daten zu einem Daten-I/O-Anschluß des integrierten Logik-DRAM **130b** eingegeben werden oder entgegengesetzt dazu, daß Daten von dem Daten-I/O-Anschluß davon ausgegeben werden. Daher ist ein Bus, mit dem der Daten-I/O-Anschluß verbunden ist, des externen Bus ext.bus offen zu anderen Chips. Folglich kann der integrierte Logik-DRAM **130a** das Liefern/Empfangen von Daten mit dem externen Bus ext.bus durchführen.

[0431] **Fig.** 53 ist ein Zeitablaufdiagramm, das einen Zustand darstellt, in dem der integrierte Logik-DRAM **130b** in einem vollen Seitenmodus des Register-DRAM-Übertragungsmodus tätig ist.

[0432] Das heißt, ein ACT-Befehl und eine Zeilenadresse Xa werden an einem Zeitpunkt t1 von **Fig.** 53 eingegeben, ein Lesebefehl RD (oder Schreibbefehl WT) und eine Spaltenadresse Ya werden zu einem Zeitpunkt t2 eingegeben, und danach startet der integrierte Logik-DRAM **130b** einen Betrieb in dem vollen Seitenmodus. Während einer Periode TP1 bis ein Vorladebefehl oder ähnliches gegeben wird, ist eine Zeilenadresse bei Xa fixiert, und Daten in dem DRAM-Abschnitt an den Spaltenadressen Y = Ya, Ya + 1, Ya + 2, ... werden zwischen dem DRAM-Abschnitt und dem Register übertragen, während die Spaltenadressen intern erzeugt werden.

[0433] Daher sind während einer Periode TP2 nicht nur ein Bus, mit dem der Daten-I/O-Anschluß verbunden ist, sondern auch ein Adreßbus und ein Befehlsbus zu den anderen Chips offen.

[0434] Das heißt, solch eine Periode TP2 kann effektiv zum Zugreifen auf die anderen Einrichtungen benutzt werden.

[Adreßzuordnung in Register]

[0435] **Fig.** 54 ist eine Zeichnung, die eine Zuordnung von Adressen in dem Register 0, dem erste Datenregister **84** und dem zweite Datenregister **86** darstellt. **Fig.** 55 ist eine Zeichnung, die ein Beispiel von in den Registern gehaltenen Daten darstellt.

[0436] Es sei angemerkt, daß angenommen ist, daß alle Zeilenadresse X gleich #3FFF sind.

[0437] Bezug nehmend auf **Fig.** 54 und 55, ein Softwarezurücksetzen wird durchführt, oder ein Flag wird an eine Spaltenadresse Y = #00 gesetzt.

[0438] Wenn hierin der Wert D0 gleich 1 ist, dann bezeichnet dieses, daß eine kryptografische Funktion zurückgesetzt ist, und der Wert D1 ist ein Flag, das anzeigt, daß ein kryptografischer Vorgang stattfindet.

[0439] Eine Y-Adresse #01 ist ein Gebiet zum Setzen eines Modus oder eines kryptografischen Gebiets.

[0440] In diesem Gebiet werden eine Auswahl eines Kryptosystems, eine Auswahl einer Länge eines Schlüssels und eines Blockverschlüsselungsmodus und zusätzlich eine Bezeichnung einer Bank zwischen einem internen Register durchgeführt, wobei eine direkte Datenübertragung während der Periode des Registerübertragungsmodus durchgeführt werden kann.

[0441] Eine Y-Adresse #02 ist ein Gebiet zum Halten von: Daten, die anzeigen, ob Verschlüsselung oder Entschlüsselung oder ein Eintritt in einen Haltezustand durchgeführt wird; Daten, die anzeigen, ob eine Starteingabe, Stoppeingabe oder Eintrittseingabe in einen Haltezustand eines einfachen Textes oder eines Chiffriertextes spezifiziert ist; eines Signals zum Zurücksetzen des Zählers **85** des ersten Registers **84** und eines Signals zum Zurücksetzen des Adreßzählers **87** des zweiten Registers **86**.

[0442] Zusätzlich zu den oben beschriebenen Tätigkeiten werden im Fall einer Blockchiffrierung in einer Kette ebenfalls Daten gespeichert, die anzeigen, ob ein anfänglicher Vektor eingegeben ist oder alle Daten in der Verarbeitung als anfängliche Daten ausgewählt sind.

[0443] Auf eine Y-Adresse #03 wird zugegriffen, wenn Datenschreiben in das erste Register **84** durchgeführt wird.

[0444] Eine Y-Adresse #04 ist eine Adresse, die einen Zugriff zu dem zweiten Register **86** anzeigt.

[0445] An einer Y-Adresse #05 sind Daten zum Steuern des Register-DR-Übertragungsmodus gespeichert.

[0446] An einer Y-Adresse #06 sind Daten zum Steuern eines Teilauffrischens gespeichert, wobei der Ausdruck "Teilauffrischen" eine Funktion zum Auffrischen nur eines spezifizierten Adreßraums bei der Selbstauffrischung bedeutet.

[0447] An Y-Adressen #10 bis #13 ist ein erster Schlüssel von 64 Bit in der Länge gespeichert, und an Y-Adressen #14 bis #17 ist ein zweiter Schlüssel mit 64 Bit in der Länge gespeichert.

[0448] An Y-Adressen #18 bis 1b ist ein dritter Schlüssel von 64 Bit in der Länge gespeichert.

[0449] An Y-Adressen #1c bis 1F ist ein anfänglicher Wert eines anfänglichen Vektors gespeichert.

[0450] Y-Adressen #20 bis #5F sind ein Gebiet, das für einen öffentlichen Schlüssel reserviert ist.

[DRAM-Registerbetriebsmodus]

[0451] Als nächstes wird eine Beschreibung von anderen Tätigkeiten zwischen dem DRAM und jedem des Registers 0, des ersten und des zweiten Registers **84** und **86** gegeben.

[0452] **Fig.** 56 ist ein schematisches Blockschaltbild zum Beschreiben eines Betriebsmodus, in dem Daten, die zuvor in solch einem DRAM gespeichert sind, in der Logikschaltung **74** chiffriert werden zum Ausgeben eines Resultats zu der Außenseite (der Betriebsmodus wird hier im folgenden als DRAM-Registerbetriebsmodus bezeichnet).

[0453] Bezug nehmend auf **Fig.** 56 wird zuerst ein vorgeschriebenes Signal von den Steuersignaleingangsanschlüssen **10** und **11** eingegeben zum Schreiben von Daten in das Register 0 und Eintreten in den Register-DRAM-Übertragungsmodus ([1]).

[0454] Dann werden Daten, von denen gewünscht wird, daß sie chiffriert oder dechiffriert werden, zu dem ersten Register **86** von dem DRAM-Abschnitt übertragen ([2]). Wenn der Zähler des ersten Registers **84** zurückgesetzt ist, werden Daten übertragen, wobei an der führenden Stelle des ersten Registers **84** gestartet wird, während, wenn es nicht zurückgesetzt ist, die Datenübertragung an einer Stelle neu gestartet, die irgendwo zwischen der führenden Stelle und der letzten Stelle ist. In diesem Fall werden die zu übertragenden Daten durch Durchführen des Lesezugriffs auf eine Adresse in dem DRAM-Abschnitt spezifiziert, von dem die Übertragung gewünscht wird ([3]).

[0455] Wenn Daten von 8 Byte in der Länge in das erste Register **84** eingegeben sind, startet eine Verarbeitung in der kryptografischen Betriebslogikschaltung **74** ([3]). Daten, deren Verarbeitung beendet worden ist, werden in das zweite Register **86** geschrieben, jeweils mit Daten von 8 Byte als eine Einheit ([4]).

[0456] Hierin wird durch Eingeben eines vorgeschriebenen Steuersignals der Austritt aus dem Register-DRAM-Übertragungsmodus durchgeführt ([5]). In diesem Fall ist in dem Register-DRAM-Übertragungsmodus einschließlich der Verarbeitung von [1] bis [5] der Zugriff auf eine Bank, die nicht als eine Übertragungsquelle zu dem Register spezifiziert ist, ermöglicht ohne Rücksicht, ob oder nicht die oben beschriebene Verarbeitung von [1] bis [5] durchgeführt wird.

[0457] Danach wird mit dem integrierten Logik-DRAM **130** bestätigt, daß das Flag FL = "0" ist, das Auslesen von Daten wird aus dem zweiten Register **86** durch den Daten-I/O-Anschluß **14** ausgeführt ([6]).

[0458] **Fig.** 57 ist ein Flußdiagramm zum Beschreiben im einzelnen des in **Fig.** 56 beschriebenen Betriebs.

[0459] Bezug nehmend auf **Fig.** 57 ist die Verarbeitung bis zu Schritt S312 grundsätzlich ähnlich zu der in **Fig.** 51 gezeigten Verarbeitung.

[0460] Hierauf folgend startet, nachdem eine anfängliche Eingabe und Auswahl von Verschlüsselung/Entschlüsselung durchgeführt sind (Schritt S340), die Verarbeitung (Schritt S342), und der Eintritt in den Register-DRAM-Übertragungsmodus wird durchgeführt (Schritt S344).

[0461] Darauf folgend wird der Adreßzähler **85** des ersten Registers **84** zurückgesetzt (Schritt S346), und ein Lesebefehl an den DRAM wird eingegeben (Schritt S348).

[0462] Als Reaktion darauf werden zum Beispiel Daten zu dem ersten Register, zum Beispiel zu der Bank 3 übertragen.

[0463] Wenn die Dateneingabe zu dem ersten Register beendet ist, dann wird ein Befehl zum Verlassen des Register-DRAM-Übertragungsmodus gegeben (Schritt S350).

[0464] Danach wird ein Startbefehl zum Verarbeiten eingegeben (Schritt S352), und dadurch werden die Daten in dem ersten Register chiffriert/dechiffriert, wobei die Verarbeitung bis zu den letzten Datenbit in dem ersten Register zu einer automatischen Beendigung fortfährt.

[0465] Wenn danach bestätigt wird, daß das Flag FL = "0" ist, werden die Inhalte aus dem zweiten Register aus dem Daten-I/O-Anschluß **14** ausgelesen (Schritt S360).

[0466] Hierauf folgend wird eine der Verarbeitungen von Schritt S340 und S344 gemäß einer Bezeichnung durch ein Steuersignal wieder hergestellt.

[0467] Durch Durchführen der Verarbeitung wie beschrieben kann, nachdem eine Zwischenregistertätigkeit beendet ist, ein Verarbeitungsresultat zu der Außenseite von dem zweiten Register **86** übertragen werden.

[0468] In einem Fall, in dem in Schritt S342 ein Verarbeitungsstart eingegeben wird, wird die Verarbeitung gestartet, wenn Daten einer minimalen Betriebseinheit in das erste Register gespeichert sind.

(Verringerung im Wartezustandsstrom unter Benutzung des Register-DRAM-Übertragungsmodus)

[0469] Weiterhin kann durch die Benutzung des Register-DRAM-Übertragungsmodus ein Ruhestrom des integrierten Logik-DRAM **130a** auch um einen großen Spielraum verringert werden, wie unten beschrieben wird.

[0470] Das heißt, es ist angenommen, daß das erste Register **84** aus zum Beispiel einer Schaltung aufgebaut ist, die keine Auffrischung benötigt, wie ein SRAM.

[0471] In diesem Fall werden, wenn eine Größe von Daten, die gehalten werden, gleich oder kleiner als eine Größe des ersten Registers **84** ist, Daten der Größe zu dem ersten Register **84** übertragen, und danach wird kein Auffrischbetrieb in dem DRAM-Abschnitt benötigt. Als Resultat kann ein Ruhestrom unterdrückt werden.

[0472] Zum Durchführen eines solchen Betriebs

wird zum Beispiel die folgende Prozedur angenommen.

i) Daten, die in dem Register-DRAM-Übertragungsmodus benötigt werden, werden direkt zu dem ersten Register **84** auf den DRAM-Abschnitt übertragen;

ii) alle Bänke werden in einen nichtausgewählten Zustand durch Zurücksetzen einer Bank zur Teilselbstauffrischung versetzt;

iii) der Eintritt in den Selbstauffrischmodus wird durchgeführt; und

iv) das System geht in einen Ruhezustand.

[0473] Bei einem gewöhnlichen Selbstauffrischen werden die Eingangspuffer für einen Befehl, eine Adresse, ein Daten-I/O-Anschluß und andere in den inaktiven Zustand gezwungen, und in diesem Zustand wird der DRAM-Abschnitt einem automatischen Auffrischen unterworfen. In dieser Situation wird zum Beispiel ein Ruhestrom auf einem Niveau in der Größenordnung von 300 μA verbraucht. Wenn dagegen die oben beschriebenen Tätigkeiten durchgeführt werden, kann ein Ruhestrom auf zum Beispiel einen Wert gleich oder kleiner als 20 μA verringert werden, da kein Strom, der für eine Auffrischtätigkeit benötigt wird, verbraucht wird.

Erste Modifikation des vierten Beispiels

[0474] **Fig.** 58 ist eine Zeichnung, die ein Konzept eines Datenübertragungszeitpunkts zum Verbessern einer Übertragungseffektivität darstellt, wenn ein Betrieb in dem Register-DRAM-Übertragungsmodus durchgeführt wird.

[0475] Bezug nehmend auf **Fig.** 58 wird in einem Fall, in dem Daten zu dem DRAM-Abschnitt von dem zweiten Register **86** geschrieben werden, folgende Textdaten, von denen gewünscht wird, daß sie eingegeben werden, an den Daten-I/O-Anschluß **14** gegeben, wenn eine DRAM-Gebietsadresse an den integrierten Logik-DRAM **130** in dem Register-DRAM-Übertragungsmodus gegeben wird.

[0476] In diesem Fall werden Daten nicht nur zu dem DRAM-Abschnitt von dem zweiten Register **86** übertragen, sondern die Daten, die an den Daten-I/O-Anschluß **14** gegeben sind, werden ebenfalls an das erste Register **84** eingegeben. Indem das getan wird, kann die Datenübertragungseffektivität verbessert werden.

[0477] Wenn andererseits eine DRAM-Adresse an den integrierten Logik-DRAM 130 zum Übertragen von Daten zu dem ersten Register **84** von dem DRAM-Abschnitt gegeben wird, werden nicht nur die Daten einer vorgeschriebenen DRAM-Adresse zu dem ersten Register **84** übertragen, sondern Daten des zweiten Registers **86** werden an die Außenseite von dem Daten-I/O-Anschluß **14** ausgegeben. Indem das getan wird, kann die Effektivität der Datenübertragung verbessert werden.

Zweite Modifikation des vierten Beispiels

[0478] **Fig.** 59 ist ein Zeitablaufdiagramm, das einen Betrieb zum Verbessern einer Effektivität der Datenübertragung beschreibt, wenn eine Frequenz des externen Taktsignals Ext.CLK niedrig ist.

[0479] Wenn die Frequenz des externen Taktsignals Ext.CLK niedrig ist, ist das interne Taktsignal clkM, das zu dem DRAM-Abschnitt geliefert wird, von einer Frequenz mit einem Wert, der durch Multiplizieren der Frequenz des externen Taktsignals Ext.CLK erhalten wird. In diesem Fall wird auf das Register und den DRAM-Abschnitt abwechselnd in dem ersten und dem zweiten Halbzyklus des Taktes zugegriffen.

[0480] Das heißt, zu einem Zeitpunkt t1 werden, wenn die Datenübertragung (Schreiben) von dem zweiten Register **86** zu dem DRAM-Abschnitt befohlen ist, die Daten des zweiten Registers **86** zuerst zu dem DRAM-Abschnitt übertragen. Darauf folgend werden zu einem Zeitpunkt t2 der nächsten Aktivierungszeit des internen Taktsignals clkM Daten, die an den Daten-I/O-Anschluß **14** gegeben sind, an das erste Register **84** eingegeben. Danach wird die Datenübertragung von dem Register zu dem DRAM-Abschnitt entsprechend durchgeführt.

[0481] Auf der anderen Seite werden zu einem Zeitpunkt t3, wenn Datenübertragung (Auslesen) zu dem ersten Register **84** von dem DRAM-Abschnitt befohlen ist, Daten des DRAM-Abschnitts zuerst zu dem ersten Register **84** übertragen. Dann werden zu einem Zeitpunkt t4 einer noch nächsten Aktivierungszeit des internen Taktsignals clkM Daten des zweiten Registers **86** zu dem DRAM-Abschnitt übertragen. Danach wird die Datenübertragung von dem DRAM-Abschnitt zu der Außenseite auf ähnliche Weise durchgeführt.

[0482] Wenn die oben beschriebenen Tätigkeiten angenommen werden, kann eine Effektivität der Datenübertragung verbessert werden.

[0483] Es sei angemerkt, daß während in der oben dargestellten Beschreibung eine Breite des internen Busses mbus zum Beispiel 16 Bit beträgt, der Zugriff von dem Register und der Zugriff von dem DRAM-Abschnitt simultan durch Annehmen einer Datenübertragung mit einer Busbreite von 32 Bit davon durchgeführt werden kann.

[0484] **Fig.** 60 ist ein Blockschaltbild, das ein Beispiel einer Schaltungskonfiguration zum Durchführen von Verschlüsselung und Entschlüsselungverarbeitung in dem CBC-Modus darstellt.

[0485] An dem ersten Teil in dem Start der Verschlüsselung wird ein anfänglicher Vektor IV durch den Multiplexer **302** ausgewählt und an den Multiplexer **304** gegeben. Wenn andererseits die Verarbeitung des nächsten Datenblocks während der Verschlüsselung durchgeführt wird, wird unmittelbar ein vorangehendes Verarbeitungsresultat in dem Multiplexer **302** an den Multiplexer **304** davon gegeben.

[0486] Der Multiplexer **304** gibt Daten von dem Multiplexer **302** an einen Eingang einer Logikbetriebs-

schaltung **308** der exklusiven Summe bei der Verschlüsselung, während er Eingangsdaten an dem einen Eingang der Logikbetriebsschaltung **308** der logischen Summe während der Entschlüsselung gibt.

[0487] Der Multiplexer **306** gibt Eingangsdaten an den anderen Eingang der logischen Betriebsschaltung **308** der exklusiven Summe bei der Verschlüsselung, während er eine Ausgabe einer Verschlüsselungs/Entschlüsselungsschaltung **312** an den anderen Eingang der logischen Betriebsschaltung **308** der exklusiven Summe bei der Entschlüsselung gibt.

[0488] Die Ausgabe der Logikbetriebsschaltung **308** der exklusiven Summe wird an die Verschlüsselungs/Entschlüsselungsschaltung **312** durch den Multiplexer **310** bei der Verschlüsselung gegeben, während sie andererseits durch den Multiplexer **314** ausgegeben wird.

[0489] Bei der Verschlüsselung werden Eingangsdaten an die Verschlüsselungs/Entschlüsselungsschaltung **312** gegeben, und eine Ausgabe der Verschlüsselungs/Entschlüsselungsschaltung **312** wird an den anderen Eingang der Logikbetriebsschaltung **308** der exklusiven Summe durch den Multiplexer **306** gegeben.

[0490] Bei der Verschlüsselung andererseits wird eine Ausgabe der Verschlüsselungs/Entschlüsselungsschaltung **312** durch den Multiplexer **314** ausgegeben.

[0491] In dem CBC-Modus sind solche Konfigurationen kaskadeverbunden zum Ermöglichen der Verschlüsselung und Entschlüsselung, wie in **Fig.** 18 und 19 beschrieben ist.

Fünftes Beispiel

[0492] **Fig.** 61 ist ein schematisches Blockschaltbild zum Beschreiben einer Konfiguration eines integrierten Logik-DRAM **152** eines fünften Beispiels der vorliegenden Erfindung.

[0493] Eine Konfiguration des integrierten Logik-DRAM **132** des fünften Beispiels, wie in **Fig.** 61 gezeigt ist, ist fast die gleiche wie die des integrierten Logik-DRAM **130** des in **Fig.** 45 gezeigten vierten Beispiels.

[0494] Der integrierte Logik-DRAM **132** unterscheidet sich jedoch von dem integrierten Logik-DRAM **130** des vierten Beispiels darin, daß das zweite Register **86** entfernt ist und Eingabe/Ausgabe von Daten auf der Logikschaltung **74** durch das erste Register **84** durchgeführt wird.

[0495] Die anderen Punkte sind grundsätzlich ähnlich zu entsprechenden Punkten in der Konfiguration des integrierten Logik-DRAM **130** des in **Fig.** 45 gezeigten vierten Beispiels; daher sind die gleichen Bezugszeichen an den gleichen Bestandteilen angebracht, und keine Beschreibung davon wird wiederholt.

[Register-Registerbetrieb]

[0496] Als nächstes wird eine Beschreibung eines Register-Registerbetriebs des integrierten Logik-DRAM **132** des in **Fig.** 61 gezeigten fünften Beispiels gegeben. **Fig.** 62 ist ein konzeptuelles Blockschaltbild zum Beschreiben der Register-Registertätigkeit des integrierten Logik-DRAM **132** des fünften Beispiels.

[0497] Der Betrieb ist ähnlich zu dem Betrieb von **Fig.** 46 mit der Ausnahme, daß die Logikschaltung **74** das Liefern/Empfangen von Daten durch das Register **84** durchführt.

[0498] **Fig.** 63 ist ein Flußdiagramm zum Beschreiben solch eines Betriebs des integrierten Logik-DRAM **132** auf eine detaillierte Weise, die zu **Fig.** 47 vergleichbar ist.

[0499] Im Vergleich mit **Fig.** 47 startet die Verarbeitungstätigkeit in Schritt S319 anstelle in Schritten S318, S320 und S322.

[0500] **Fig.** 64 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebs des integrierten Logik-DRAM **132** in dem Verarbeitungsfluß, wie in **Fig.** 61 gezeigt ist.

[0501] Auf den DRAM-Abschnitt kann mit Ausnahme einer Periode zugegriffen werden, in der das Schreiben und Lesen in dem ersten Register **84** durchgeführt wird, selbst während des Betriebs der kryptografischen Betriebslogikschaltung **74**.

[0502] **Fig.** 65 ist ein konzeptuelles Blockschaltbild zum Beschreiben des Register-DRAM-Betriebs des in **Fig.** 61 gezeigten integrierten Logik-DRAM **132**.

[0503] Der Betrieb ist ähnlich zu dem von **Fig.** 49 mit der Ausnahme, daß die Logikschaltung **74** das Liefern/Empfangen von Daten durch das Register **84** durchführt.

[0504] **Fig.** 66 ist ein Flußdiagramm zum Beschreiben des in **Fig.** 65 beschriebenen Betriebs auf eine detaillierte Weise.

[0505] Im Vergleich mit **Fig.** 51 startet der Verarbeitungsbetrieb in Schritt S319 anstelle in Schritten S318, S320 und S322.

[0506] Es sei angemerkt, daß ein Betrieb ähnlich zu all den in dem vierten Beispiel beschriebenen Betrieben ebenfalls realisiert werden kann, wenn Lesen und Schreiben simultan durchgeführt werden kann und eine Bandbreite durch Annehmen einer Konfiguration des ersten Registers **84** mit Doppelports verdoppelt wird.

[0507] Weiterhin werden in dem Register **84** gespeicherte Daten D in der Logikschaltung **74** verarbeitet, und ein Verarbeitungsresultat davon wird in das Register **84** an seinem ursprünglich gespeicherten Platz geschrieben. Indem das getan wird, kann in dem fünften Beispiel ein Adreßzähler ausreichend sein, während in dem vierten Beispiel eine Konfiguration angenommen ist, in dem getrennte Adreßzähler für das erste bzw. zweite Register vorgesehen sind.

[0508] Noch weiterhin, wenn Daten der minimalen Betriebseinheit in das Register **84** gespeichert wer-

den und die Verarbeitung zu einer Zeit startet, kann eine Konfiguration angenommen werden, indem Daten entsprechend der in **Fig.** 50 gezeigten Zählung CT2 verarbeitet werden und das Verarbeitungsresultat davon in einen ursprünglichen Platz mit der Zählung CT2 als eine Referenz geschrieben werden.

[0509] Es sei angemerkt, daß durch Benutzen des Register-DRAM-Übertragungsmodus ähnlich zu der Konfiguration des vierten Beispiels selbst bei der Konfiguration des fünften Beispiels eine effektive Benutzung des Busses realisiert werden kann, wenn der Bus zu anderen Chips offen ist.

Erste Modifikation des fünften Beispiels

[0510] **Fig.** 67 ist eine konzeptuelle Zeichnung zum Beschreiben der Datenübertragungsverarbeitung zwischen dem Register **84** und einer Logikschaltung **74** in einer ersten Modifikation des fünften Beispiels.

[0511] Bezug nehmend auf **Fig.** 67 liest die Logikschaltung **74** nächste Daten zuvor während einer gegenwärtigen Tätigkeit im voraus. Durch Durchführen solcher Verarbeitung können daher, da die Inhalte der Register **84** während des Betriebs zwischen dem Zeitpunkt T1 bis T2 gelesen werden, Daten eines Verarbeitungsresultats unmittelbar zu dem Register **84** geschrieben werden und zu einem Zeitpunkt t3 ausgegeben werden, wenn der Betrieb beendet ist.

Zweite Modifikation des fünften Beispiels

[0512] **Fig.** 68 ist ein konzeptuelles Blockschaltbild zum Beschreiben einer Route von dem Register **84** zu einem Datenausgang in einer zweiten Modifikation des fünften Beispiels.

[0513] Eine Konfiguration wird angenommen, bei der eine Vorausleseverriegelungsschaltung **88** zwischen dem Register **84** und dem I/O-Puffer **52** vorgesehen ist. Das heißt, wenn die Verarbeitung in der Logikschaltung **74** beendet ist, wird das führende Betriebsresultat zuvor in die Vorausleseverriegelungsschaltung **88** von dem Register **84** gelesen und in der Vorausleseverriegelungsschaltung **88** gehalten.

[0514] Die Zeit der Datenausgabe von dem Register **84** an den Daten-I/O-Anschluß **14** muß zum Beispiel die gleiche sein wie die eines Allzweck-SDRAM. Mit solch einer Konfiguration tritt keine Chance auf, daß die Zeit der Datenausgabe verzögert ist, selbst wenn die Daten an den Daten-I/O-Anschluß **14** ausgegeben werden, nachdem ein externer Befehl angenommen ist.

[DRAM-Registerbetriebsmodus des fünften Beispiels]

[0515] **Fig.** 69 ist ein konzeptuelles Blockschaltbild zum Beschreiben eines Betriebsmodus, bei dem Daten, die zuvor in dem DRAM-Abschnitt gespeichert sind, in der Logikschaltung **74** chiffriert werden und danach nach außen ausgegeben werden, das ist der DRAM-Registerbetriebsmodus in dem fünften Beispiel.

[0516] Der Betrieb ist ähnlich zu dem Betrieb von **Fig.** 56 mit der Ausnahme, daß die Logikschaltung **74** Liefern/Empfangen von Daten durch das Register **84** durchführt.

[0517] **Fig.** 70 ist ein Flußdiagramm zum detaillierteren Beschreiben des in **Fig.** 69 beschriebenen Betriebs.

[0518] Im Vergleich mit **Fig.** 57 sind die Schritte S342 und S356 beseitigt, und das Datenlesen wird in Schritt S360' statt in S360 durchgeführt.

[0519] Durch Durchführen der oben beschriebenen Verarbeitung kann ein Verarbeitungsresultat nach außen von dem ersten Register **84** übertragen werden.

[0520] Es sei angemerkt, daß auch in diesem Fall ein Betrieb ähnlich zu dem in dem vierten Beispiel realisiert werden kann, wenn Schreiben und Lesen simultan durchgeführt werden können und eine Bandbreite durch Annehmen einer Konfiguration des ersten Registers **84** mit Doppelports verdoppelt ist.

[0521] Weiterhin kann durch Benutzen des Register-DRAM-Übertragungsmodus ähnlich zu dem vierten Beispiel ein Ruhestrom um einen großen Spielraum verringert werden.

Sechstes Beispiel

[0522] Eine Konfiguration wird beschrieben, bei der es kein Hindernis für die Verschlüsselung gibt, selbst wenn ein Selbstauffrischbefehl extern an den in **Fig.** 45 gezeigten integrierten Logik-DRAM **130** oder den in **Fig.** 61 gezeigten integrierten Logik-DRAM **132** gegeben wird.

[0523] Zuerst enthält der Adreßzähler **42b** Zeilenadreßzähler, die für die entsprechenden Bänke in der in **Fig.** 45 gezeigten Konfiguration vorgesehen sind. **Fig.** 71 ist in solch einer Konfiguration ein Blockschaltbild, das eine Schaltungskonfiguration darstellt, die einen internen Befehl zum Selbstauffrischen erzeugt.

[0524] Ein extern gegebener Selbstauffrischbefehl wird an ein Eingangsende der AND-Schaltungen **402**, **404**, **406** und **408** gegeben.

[0525] Andererseits wird an die anderen Eingangsenden der AND-Schaltungen **402** bis **408** entsprechende Signale CONT[0] bis CONT[3] gegeben, die bezeichnen, welche der Bänke für eine Bank spezifiziert wird zur Benutzung bei der Verschlüsselung, die Liefern/Empfangen mit einem Register durchführt. Wenn zum Beispiel der DRAM-Abschnitt nicht zur Benutzung bei der Verschlüsselung spezifiziert ist, [CONT[0] bis CONT[3]] = [1, 1, 1,].

[0526] Wenn andererseits eine Bank zur Benutzung bei der Verschlüsselung eine Bank #0 ist, [CONT[0] bis CONT[3]] = [0, 1, 1, 1]. In diesem Fall werden interne Selbstauffrischbefehle AREF1 bis RREF3 für die Bänke #1 bis #3 aktiviert, während ein Signal AREF0 für die Bank #0 nicht aktiviert wird. Daher

empfängt eine zur Benutzung in der Verschlüsselung spezifizierte Bank keinen Einfluß eines externen Selbstauffrischbefehls.

[0527] Wenn es in irgendeinem der Register gespeichert ist, daß ein Selbstauffrischbefehl gegeben worden ist, muß die Bank #0 nur den Selbstauffrisch empfangen, nachdem die Verschlüsselung beendet ist.

Siebtes Beispiel

[0528] Zu den integrierten Logik-DRAMs **130**, **132** usw. kann weiter eine Funktion hinzugefügt werden, die unten beschrieben wird, zum Erzielen einer weiteren Abnahme des Leistungsverbrauchs.

[0529] Das heißt, zuerst kann eine Funktion hinzugefügt werden derart, daß eine Bezeichnung eines Teilauffrischmodus möglich sein kann.

[0530] Das heißt, ein Speicherraum, der in einem Selbstauffrischmodus aufzufrischen ist, kann mit einer Bank als eine Einheit bezeichnet werden. In diesem Fall ist eine Konfiguration grundsätzlich ähnlich zu dem sechsten Beispiel vorgesehen, und es ist nun notwendig, daß ein interner Befehl für das Selbstauffrischen ebenfalls für jede Bank erzeugt wird.

[0531] Weiterhin kann zum Verringern eines Ruhestroms in einem nicht abgeschalteten Modus ein Niedrigleistungsmodus gesetzt werden, wie unten beschrieben wird.

[0532] **Fig.** 72 ist ein Zeitablaufdiagramm zum Beschreiben eines solchen Niedrigleistungsmodus.

[0533] Es sei angemerkt, daß der Eintritt in den Niedrigleistungsmodus gesteuert wird dadurch, ob eine spezifische Adresse, die zugeordnet ist, wenn ein Modusregistersetzbefehl (MRS-Befehl) eingegeben ist, eingegeben ist oder ein Logiksteuerraum, wie er in **Fig.** 2 beschrieben ist, zugeordnet ist.

[0534] Bezug nehmend auf **Fig.** 72 wird zu einem Zeitpunkt t0 nicht nur ein Chipauswahlsignal Ext.CS aktiviert (auf dem "L"-Pegel), sondern auch der Vorladebefehl PRE wird an der Aktivierungskante des externen Taktsignals Ext.CLK gegeben.

[0535] In dem Niedrigleistungsmodus ist während einer Periode, wenn wie in einer Periode TA von **Fig.** 72 das Chipauswahlsignal auf dem "H"-Pegel, eine Bank ist inaktiv, und zusätzlich während einer Periode, während der keine Daten ausgegeben werden, wird der Leistungsverbrauch abgeschnitten wie unten beschrieben wird. Das heißt, in solch einer Periode TA sind die beiden Signale clkM und clkL inaktiv, und die Eingangspuffer der ersten Stufe für die entsprechenden Signale /RAS, /CAS, /WE und ADD sind ebenfalls inaktiv.

[0536] **Fig.** 73 ist ein Bild, das eine Schaltungskonfiguration darstellt, die eine Steuerung eines Eingangspuffer **40** oder **46** in solch einem Niedrigleistungsmodus durchführt.

[0537] Das heißt, nachdem das externe Taktsignal ext.CLK an einen Eingangspuffer **500** erster Stufe eingegeben ist, wird das Signal weiter an eine interne Takterzeugerschaltung **44** durch eine AND-Schaltung **502** gegeben, die durch das Signal EN aktiviert wird. Andererseits wird eine Ausgabe des Eingangspuffers **500** erster Stufe an einen Chipauswahlsignaleingangspuffer als ein Signal CLKcs durch einen Puffer **504** einer unmittelbar folgenden Stufe gegeben.

[0538] Der Chipauswahlsignaleingangspuffer enthält: einen Puffer **510** erster Stufe, der ein Signal ext.CS empfängt; ein Übertragungsgatter **512**, das durch ein Signal CLKcs gesteuert wird; eine Verriegelungsschaltung **514** zum Halten einer Ausgabe des Übertragungsgatters **512** zum Ausgeben eines Signals CSp2; und eine OR-Schaltung **516**, die ein Signal RASOR, ein Resultat einer logischen Summenoperation zwischen dem Signal CSp2 und dem internen RAS-Signal für jede Bank empfängt, zum Ausgeben des Signals EN.

[0539] In den Pufferschaltungen **40** und **46** ist ein Logikgatter **520**, das die Befehlssignale /RAS, /WE oder /CAS empfängt, oder das Adreßsignal ADD, die nicht das Chipauswahlsignal sind, und das Signal CKE, eine NOR-Schaltung, die ein entsprechendes aus den Signalen an einem Eingang davon empfängt und weiter ein invertiertes Signal/EN des Signals EN an dem anderen Eingangsende davon empfängt.

[0540] **Fig.** 74 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebs der in **Fig.** 73 gezeigten Schaltung.

[0541] Wenn ein ACT-Befehl zu einem Zeitpunkt t0 gegeben wird, wird das Signal EN zum Aktivieren des Logikgatters **520** aktiviert.

[0542] Wenn andererseits das Chipauswahlsignal Ext.CS aktiviert wird und der Vorladebefehl PRE zu einem Zeitpunkt t2 gegeben wird, wird das Signal EN an einem Zeitpunkt t3 zum Deaktivieren ebenfalls des Logikgatters **520** deaktiviert. Weiterhin wird das externe Taktsignal Ext.CLK nicht an die interne Takterzeugerschaltung **44** gegeben, und keines der internen Taktsignale clkM und clkL wird erzeugt. Wenn jedoch die Ausgabe von Daten fortfährt, selbst nachdem ein Vorladebefehl eingegeben ist, wird das Signal clkM während der Datenausgabe aktiviert.

[0543] Daher kann der Leistungsverbrauch in dem Niedrigleistungsmodus verringert werden.

Achtes Beispiel

[0544] **Fig.** 75 ist eine Darstellung zum einfachen Beschreiben der Verarbeitung einer Sicherheitsdatenkommunikation in dem Internet.

[1] Eine Sicherheitssite wird auf einer Benutzerseite angeklickt.

[2] Als Reaktion darauf sendet eine Hostseite (ein Server usw.) ein Authentifizierungsdokument eines Servers. Das Authentifizierungsdokument des Servers enthält: eine Seriennummer eines Zertifikates und ein Ablaufdatum davon, einen öffentlichen Schlüssel eines Servers, eine elektronische Signatur, die durch ein Authentifizierungsbüro vorbereitet ist, worin der Ausdruck "elektroni-

sche Signatur" Daten bedeutet, die durch Chiffrieren einer Verarbeitungssammlung (MD) des Authentifizierungsbüros mit einem privaten Schlüssel (oder geheimem Schlüssel) des Authentifizierungsbüros erhalten sind.

[3] Darauf folgend wird die Authentifizierung des Servers auf der Seite des Benutzers durchgeführt.

[0545] Einzelheiten des oben beschriebenen Vorgangs sind wie folgt:

Zuerst 1) eine Nachrichtensammlung wird mit dem öffentlichen Schlüssel des Authentifizierungsbüros chiffriert (*). Der öffentliche Schlüssel des Authentifizierungsbüros ist im allgemeinen in einem Browser enthalten.

2) Eine Nachrichtensammlung des Authentifizierungsdokuments wird berechnet, und es wird geprüft, ob oder nicht die Berechnung mit einer dechiffrierten Nachrichtensammlung übereinstimmt.

3) Ein öffentlicher Schlüssel des Servers wird erhalten.

[0546] Indem dem oben beschriebenen Vorgang gefolgt wird, wird zum Beispiel eine Verbindung in SSL (Sicherheits-Socket-Schicht) gestartet. Dann wird die Verarbeitung von der Benutzerseite zu dem Server durchgeführt zum Empfangen seiner Authentifizierung zum Zugriff auf den Server.

[0547] Zum Beispiel werden eine Benutzer-ID, ein Paßwort und andere zu dem Host (Server) nach Verschlüsselung mit dem öffentlichen Schlüssel des Servers (*) gesendet.

[0548] Bei dieser Verarbeitung tritt ein Fall auf, wenn die Benutzerseite ein Authentifizierungsdokument hat, in dem die Benutzerseite die gleichen Tätigkeiten wie die oben beschriebenen durchführt, die durch den Server durchgeführt werden.

4) Darauf folgend führt die Hostseite die Authentifikation des Benutzers unter Benutzung einer Benutzer-ID und eines Paßwortes des Benutzers durch, die zu der Hostseite gesendet worden sind.

Das heißt, die Authentifikationsverarbeitung wird durchgeführt durch Entschlüsseln der ID und des Paßwortes des Benutzers unter Benutzung eines geheimen Schlüssel des Servers.

5) Die Benutzerseite führt die Auswahl eines Geheimschlüsselkryptosystems durch.

Die Hostseite bestimmt ein Geheimschlüsselkryptosystem und benachrichtigt die Benutzerseite, welches Geheimschlüsselkryptosysten angenommen ist und erhält die Zufallszahl und den gegenwärtigen Zeitpunkt [2].

7) Der Benutzer erzeugt Zufallszahlen, die als eine Basis eines geheimen Schlüssels dienen.

Die erzeugten Zufallszahlen werden mit dem öffentlichen Schlüssel des Servers chiffriert und danach zu dem Server (*) gesendet.

8) Weiterhin verschlüsselt die Benutzerseite einen geheimen Schlüssel in Übertragung von dem Benutzer zu dem Server mit dem öffentlichen Schlüssel des Servers zum Übertragen des geheimen Schlüssels, der bei der Übertragung (*) benutzt wird.

9) Die Hostseite entschlüsselt den geheimen Schlüssel in der Übertragung von dem Benutzer zu dem Server mit dem privaten Schlüssel (geheimen Schlüssel) zum Erhalten des geheimen Schlüssels in der Übertragung.

10) Die Benutzerseite überträgt notwendige Daten nach Verschlüsselung auf ihrer Seite. Zu dieser Zeit wird die Verschüsselung mit dem geheimen Schlüssel durchgeführt (*). Zum Beispiel wird das Dreifach-DES-System angenommen.

11) Die Hostseite führt die Entschlüsselung von Daten mit dem in 9) erhaltenen geheimen Schlüssel durch.

[0549] Andererseits chiffriert die Serverseite ebenfalls einen geheimen Schlüssel (Sitzungsschlüssel) in Übertragung von dem Server zu dem Benutzer mit dem öffentlichen Schlüssel des Benutzers zum Übertragen des geheimen Schlüssels in der Übertragung.

[0550] Der Benutzer entschlüsselt den geheimen Schlüssel in der Übertragung von dem Server zu dem Benutzer mit dem privaten Schüssel (geheimen Schlüssel) der Benutzerseite zum Erhalten des geheimen Schlüssels in der Übertragung (*).

[0551] Die Hostseite überträgt Daten zu der Benutzerseite nach Chiffrierung der Daten mit dem Sitzungsschlüssel in Übertragung von dem Server zu dem Benutzer, der so erhalten ist.

[0552] Die Benutzerseite entschlüsselt die Daten mit dem Sitzungsschlüssel, der in [9'] erhalten ist.

[0553] Bei der Datenkommunikation, wie oben beschrieben wurde, kann die Verarbeitung, die durch das Symbol (*) bezeichnet ist, mit einer integrierten Halbleiterschaltungsvorrichtung **1** oder einem integrierten Logik-DRAM durchgeführt werden, die sich auf die vorliegende Erfindung beziehen.

[0554] Es sei angemerkt, daß, wenn Daten, die durch Verschlüsselung oder ähnliches verarbeitet worden sind, in dem zweiten Register gespeichert sind und dann von dem zweiten Register ausgelesen werden, das heißt, wenn die Daten in dem Register-Registerbetriebsmodus gesetzt werden, keine Steuerung des DRAM-Abschnitts von dem Mikrocomputer **90** benötigt wird, da absolut kein Zugriff zu dem DRAM-Abschnitt auftritt. Selbst wenn weiterhin Zugriff auf den DRAM-Abschnitt oder eine Auffrischanforderung während der Verarbeitung auftritt, kann die Verarbeitung durchgeführt werden, die vor dem Zugriff oder ähnliches stattfand.

[0555] Während in einem Geheimkryptosystem Verschlüsselung mit 64 Bit als eine Einheit durchgeführt wird, tritt eine Notwendigkeit zum Durchführen eines Auffüllens gemäß einer Regel auf, wenn die Eingangsdaten weniger als 64 Bit in Länge sind. In diesem Fall wird die Verarbeitung durch eine integrierte Halbleiterschaltung **1** oder den integrierten Logik-DRAM **30** und **130** bis unmittelbar vor die letzten

Datenbit durchgeführt, und der Mikrocomputer **90** folgt der Verarbeitung zum Chiffrieren der letzten Daten.

[0556] Dieses ist so, da richtig gemäß den Daten geändert wird, wieviel Auffüllen durchgeführt wird, kann dieser Teil besser durch die Seite des Mikrocomputers **90** zum Erhalten einer einfacheren und leichteren Konfiguration als ein System durchgeführt werden.

[0557] Weiterhin kann in der integrierten Halbleiterschaltungsvorrichtung **1** oder der integrierten Logikschaltung eine Konfiguration angenommen werden, bei der zwei oder mehr Arten von Schlüsseln auf der Hand gehalten werden. Der Grund ist der, daß, da zwei Arten von Schlüsseln mit einem Authentifikationsbüro bzw. einem Server gehalten werden, wenn zwei Arten von Schlüsseln gleichzeitig auf der Hand gehalten werden, keine Notwendigkeit auftritt, daß ein Schlüssel neu jedes Mal geladen wird, wenn Verschlüsselung alternativ mit dem Authentifikationsbüro oder dem Server durchgeführt wird, da Schlüssel durchgehend für die entsprechenden zwei Benutzungen benutzt werden.

[0558] Es sei angemerkt, daß in der oben dargestellten Beschreibung Verschlüsselung hauptsächlich auf Beispiele fokussiert ist, die als Verarbeitung in einer Logikschaltung genommen sind, die in einer Halbleiterschaltungsvorrichtung zusammen mit einem Speicherabschnitt enthalten ist, es speziell hier bestätigt wird, daß die vorliegende Erfindung nicht auf die in den Beispielen beschriebenen Fälle begrenzt ist. Die Verarbeitung, die durch eine Logikschaltung durchgeführt wird, kann zum Beispiel Verschlüsselung oder weiterhin Bildverarbeitung oder ähnliches sein.

Neuntes Beispiel

[0559] Wie in **Fig.** 21 bis 31 des ersten Beispiels beschrieben wurde, treten in einem integrierten Logik-DRAM, der sich auf die vorliegende Erfindung bezieht, Fälle auf, in denen ein Taktsignal clkM zur Benutzung der Steuerung des Betriebes des DRAM-Abschnitts eine Betriebsfrequenz aufweist, die von einem relativ niedrigen Wert zu einem relativ hohen Wert reicht.

[0560] In den Fällen weist eine Leistung des DRAM-Abschnitts eine Möglichkeit auf, obwohl das Taktsignal clkM mit niedriger Geschwindigkeit tätig ist, wenn der DRAM-Abschnitt zu Zeiten gemäß der Zahl der Takte ähnlich zu einem Fall mit hoher Geschwindigkeit gesteuert wird, daß er auf ein unnötig niedriges Niveau im Vergleich mit der Erwartung für das Taktsignal bei niedriger Geschwindigkeit verschlechtert wird aufgrund des Vorhandenseins eines unnötigen Betriebsspielraums.

[0561] In der folgenden Beschreibung werden eine Konfiguration und ein Betrieb eines integrierten Logik-DRAM dargestellt, in denen, selbst wenn das Taktsigna clkM bei einer vergleichsweise niedrigen Frequenz tätig ist, ein Betrieb des DRAM-Abschnitts durchgeführt wird, wobei eine hohe Leistung behalten wird.

[Verriegelung des Spaltenbetriebs]

[0562] **Fig.** 76 ist ein schematisches Blockschaltbild zum Beschreiben eines integrierten Logik-DRAM, der sich auf ein neuntes Beispiel der vorliegenden Erfindung bezieht, welches ein Bild it, das zu **Fig.** 45 des vierten Beispiels vergleichbar ist.

[0563] Da eine Konfiguration eines integrierten Logik-DRAM **1000**, der sich auf das neunte Beispiel bezieht, eine Steuerung des DRAM-Abschnitts als ein Merkmal aufweist, kann eine Konfiguration, die mit der Steuerung des DRAM-Abschnitts verknüpft ist, auch auf den integrierten Logik-DRAM **132** angewendet werden, der in **Fig.** 61 des fünften Beispiels gezeigt ist.

[0564] Da weiterhin die Konfiguration mit hoher Leistung bei einem Betrieb niedriger Geschwindigkeit des DRAM-Abschnitts zu verknüpfen ist, kann eine unten beschriebene Konfiguration auch auf eine Konfiguration angewendet werden, in der nur ein DRAM auf einem Chip integriert ist.

[0565] Zuerst Bezug nehmend auf **Fig.** 76 ist eine Konfiguration eines integrierten Logik-DRAM **1000** grundsätzlich ähnlich zu der Konfiguration des in **Fig.** 45 gezeigten integrierten Logik-DRAM **130**.

[0566] In **Fig.** 76 enthält die Konfiguration jedoch explizit: eine Erzeugerschaltung **1100** eines internen Leistungsquellenpotentials zum Liefern eines internen Leistungsquellenpotentials; einen Leistungsquellenanschluß **17** zum Liefern eines externen Leistungsquellenpotentials Ext.Vdd an die Erzeugerschaltung **1100** des internen Leistungsquellenpotentials; und einen Leistungsquellenanschluß **18** zum Liefern eines Massepotentials Vss an die Erzeugerschaltung **1100** des internen Leistungsquellenpotentials.

[0567] Wie aus der folgenden Beschreibung klar wird, da sich die Konfiguration des integrierten Logik-DRAM **1000** des neunten Beispiels sich von der Konfiguration des integrierten Logik-DRAM **130** nur in den Konfigurationen eines DRAM-Steuerabschnitts und eines Adreßzählerabschnitts **42b** und einer Konfiguration der Erzeugerschaltung **100** des internen Leistungsquellenpotentials unterscheidet; sind daher die gleichen Bezugszeichen an die gleichen Bestandteile in der Konfiguration von **Fig.** 76 angebracht, und keine Beschreibung davon wird wiederholt.

[0568] **Fig.** 77 ist ein schematisches Blockschaltbild, das-eine Konfiguration eines DRAM-Steuerabschnitts **42b** und einen Spaltendecoder **58.0**, einen Leseverstärker **60.0** und einen I/O-Abschnitt darstellt, der für eine Bank #0 vorgesehen ist, die herausgezogen sind.

[0569] Bezug nehmend auf **Fig.** 77 enthält der DRAM-Steuerabschnitt **42b**: einen Befehlsdeco-

dier/Steuerabschnitt **1200**, der eine Eingabe von dem Eingangspuffer **40** empfängt, zum Erzeugen eines Befehlssignals; einen Lesesteuerabschnitt **1202**, der eine Ausgabe von dem Befehlsdecodier/Steuerabschnitt empfängt, zum Ausgeben von Signalen SON und SOP zum Steuern eines Betriebs des Leseverstärkers **60.0**; einen Spaltenverriegelungssignalerzeugerabschnitt **1204** zum Ausgeben eines Spaltenverriegelungssignals ZC1, wie beschrieben wird, gemäß einer Ausgabe von dem Lesesteuerabschnitt **1202**; und einen spaltenbezogenen Steuerabschnitt **1206** zum Ausgeben eines Spaltendecodieraktivierungssignals CDE gemäß der Steuerung von dem Befehlsdecodier/Steuerabschnitt **1200** und des Signals ZCE.

[0570] Der Spaltendecoder **58.0** enthält: einen Decoder **1210**, der ein Spaltenadreßdecodiersignal CAD, das gemäß einem Adreßsignal erzeugt ist, und ein Signal CDE von dem DRAM-Steuerabschnitt **42b** empfängt, zum Ausgeben eines Signals CSL zum Auswählen einer Speicherzellenspalte, die in der Bank #0 ausgewählt ist.

[0571] In der Bank #0 ist ein Bitleitungspaar BL und /BL vorgesehen, und an einem Schnittpunkt einer Wortleitung WL, die durch einen Zeilendecoder **56.0** ausgewählt ist, und einer Bitleitung BL ist eine Speicherzelle MC vorgesehen, die einen Speicherzellentransistor MTR und einen Speicherzellenkondensator C enthält.

[0572] Daten, die in einer Speicherzelle MC gespeichert sind, werden auf eine Bitleitung BL als Reaktion auf die Aktivierung einer Wortleitung WL ausgelesen und durch den Leseverstärker **60.0** verstärkt, wenn er durch die Aktivierungssignale SON und SOP aktiviert ist. Die durch den Leseverstärker **60.0** verstärkten Daten werden auf ein I/O-Leitungspaar LI/O und /LI/O von dem Bitleitungspaar BL und /BL dadurch ausgelesen, daß Transistoren TR1 und TR2, die in einem I/O-Gatter **1220** enthalten sind, das durch das Signal CSL ausgewählt ist, durch das Signal SCL leitend gemacht sind.

[0573] Weiterhin ist eine Vorladungsschaltung **1230** entsprechend zu den Bitleitungen BL und /BL zum Ausgleichen und Vorladen der Bitleitungen BL und /BL des Bitleitungspaares auf ein vorgeschriebenes Potential als Reaktion auf das Signal SPR von dem DRAM-Steuerabschnitt **42b** vorgesehen.

[0574] **Fig.** 78 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebes, bei dem Daten von einer Speicherzelle MC auf ein Bitleitungspaar ausgelesen werden und weiter auf ein I/O-Leitungspaar LI/O und /LI/O in den in **Fig.** 77 gezeigten Konfigurationen.

[0575] Hauptbefehle sind ein ACT-Befehl und ein Lese- oder Schreibbefehl in einem synchronen DRAM (hier im folgenden als SDRAM bezeichnet). Von dem Lese- und Schreibbefehl wird verlangt, daß sie voneinander in der Zeit um einen Wert getrennt sind, der mit einer Zeit tRCD definiert ist.

[0576] Zum Beispiel in einem Fall, in dem ein Spezifikationswert der Zeit tRCD = 20 ns mit einem Taktsignal clkM von 100 MHz in der Betriebsfrequenz ist, ist die Zeit tRCD so definiert, daß ein Wert tRCD = 2 × clkM als eine Funktion einer Frequenz des Taktsignals clkM angenommen wird.

[0577] In einem Fall, in dem tRCD auf solche Weise definiert ist, wird jedoch verlangt, wenn ein Taktzyklus irgendwie kleiner als 10 ns, zum Beispiel von tRCD = 9,5 ns ist, daß die Zeit tRCD auf tRCD = 3 clkM anstelle der Regel gesetzt wird, damit der Spezifikationswert von 20 ns erfüllt wird.

[0578] Nachdem jedoch ein Lese/Schreibbefehl eingegeben ist, wenn eine Reserve verfügbar in einem Zeitspielraum für einen Lese- oder Schreibbetrieb auf dem DRAM-Abschnitt verfügbar ist, wird eine Zeit eines Betriebes, der mit der Zeit tRCD verknüpft ist, das heißt eine Zeit eines Betriebes des Leseverstärkers und eine Zeit, die zwischen dem Auslesen der Daten durch den Leseverstärker **60.0** und des weiteren Auslesens der Daten auf das I/O-Leitungspaar LI/O und /LI/O benötigt wird, richtig eingestellt, und dadurch kann die Zeit tRCD zwei Takte unverändert halten, selbst wenn ein Zyklus des Taktsignals clkM gleich 9,5 ns wird, wie oben angegeben wurde.

[0579] Das heißt, wenn eine Zeit des in **Fig.** 77 beschriebenen Signals ZCE eingestellt wird, kann der Betrieb des DRAM-Abschnitts mit hoher Leistung sein, wie unten beschrieben wird.

[0580] Es wird Bezug genommen auf **Fig.** 78, der ACT-Befehl wird zu einer Zeit der Aktivierung des Taktsignals clkM an einen Zeitpunkt t0 eingegeben.

[0581] Als Reaktion geht zu einem Zeitpunkt t1 ein Pegel einer Wortleitung WL in einen aktiven Zustand zum Auslesen von Daten aus der Speicherzelle MC auf die Bitleitung BL. Darauf folgend geht als Reaktion auf die Aktivierung des ACT-Befehls zu dem Zeitpunkt t0 das Leseverstärkeraktivierungssignal SON, das von dem Lesesteuerabschnitt **1202** ausgegeben ist, auf den H-Pegel zu einem Zeitpunkt t2, und dadurch wird die Verstärkung der Spannungen, die auf das Bitleitungspaar BL und /BL ausgelesen sind, durchgeführt.

[0582] Dann wird zu einem Zeitpunkt t3, wenn eine Zeit von zwei Takten abläuft, nachdem das Taktsignal clkM in einen aktiven Zustand zu dem Zeitpunkt t0 geht, ein Lesebefehl (hier im folgenden als RD ebenfalls zur Kürze genannt) als Reaktion auf die Aktivierung des Taktsignals clkM ausgegeben. Als Reaktion wird ein Signal RD, das auf dem H-Pegel während des Burstlesens ist, von dem Befehlsdecodier/Steuerabschnitt **1200** an den spaltenbezogenen Steuerabschnitt **1206** ausgegeben. Zu diesem Zeitpunkt wird jedoch das Spaltenverriegelungssignal ZCE gleich dem H-Pegel zum Unterdrücken der Aktivierung des Signals CDE, das von dem spaltenbezogenen Steuerabschnitt **1206** ausgegeben wird.

[0583] Das heißt, während das Signal RD zum Aktivieren des Signals CDE an den spaltenbezogenen Steuerabschnitt **1206** von dem Befehlsdecodier/Steuerabschnitt **1200** zu dem Zeitpunkt t3 gemäß der Ausgabe eines Befehls RD gegeben wird,

startet der spaltenbezogene Steuerabschnitt **1206** keine Aktivierungstätigkeit des Signals CDE während einer Periode, wenn das Signal ZCE in einem aktiven Zustand ist.

[0584] Das von der Spaltenverriegelungssignalerzeugerschaltung **1204** ausgegeben Signal ZCE geht auf den L-Pegel zu einem Zeitpunkt t4, wenn eine Zeitlänge Δt abläuft, nachdem das Signal SON, das von dem Lesesteuerabschnitt **1202** ausgegeben ist, zu einem Zeitpunkt t2 aktiviert ist. Als Reaktion wird das Signal CDE auf einem aktiven Pegel von dem spaltenbezogenen Steuerabschnitt **1206** ausgegeben.

[0585] Mit anderen Worten, wenn das Spaltenverriegelungssignal ZCE auf dem H-Pegel ist, startet kein spaltenbezogener Betrieb, selbst wenn ein Lesebefehl (oder ein Schreibbefehl) eingegeben wird.

[0586] **Fig.** 78 zeigt hier mit einer gestrichelten Linie eine Zeit, zu der das Spaltenfreigabesignal CDE aktiviert wird, wenn der Spaltenverriegelungssignalerzeugerabschnitt **1204** nicht vorhanden ist, mit anderen Worten, wenn keine Steuerung durch das Spaltenverriegelungssignal ZCE vorhanden ist.

[0587] Wenn das Spaltenverriegelungssignal ZCE nicht vorhanden ist, gibt es ein Risiko, daß das Auswahlsignal CSL in einen aktiven Zustand geht, bevor die Verstärkung durch den Leseverstärker **60.0** beendet ist, und das Bitleitungspaar BL und /BL und das I/O-Leitungspaar LI/O und /LI/O miteinander verbunden werden, bevor die Potentialdifferenz zwischen dem Bitleitungspaar BL und /BL ausreichend verstärkt ist, mit dem Resultat, daß Daten zerstört werden.

[0588] Im Gegensatz dazu, wenn das Spaltenverriegelungssignal ZCE vorhanden ist, wie durch die durchgezogene Linie in **Fig.** 78 gezeigt ist, geht das Spaltendecodierfreigabesignal CDE auf den H-Pegel zu einer Zeit, zu der das Signal ZCE auf den L-Pegel geht, nachdem die Verstärkung des Potentialdifferenzpegels auf dem Bitleitungspaar BL und /BL beendet ist.

[0589] Daher kann durch Vorsehen des Verriegelungssignalerzeugerabschnitts **1204** die Zeit tRCD auf tRCD = 2 × clkM gehalten werden, selbst wenn ein Taktzyklus des Taktsignals clkM etwas kürzer als 10 ns wird.

[0590] In einem Fall, in dem eine Lesetätigkeit oder eine Schreibtätigkeit in einem Burstmodus durchgeführt wird, ist es nur notwendig, daß das Spaltendecodierfreigabesignal CDE mit einer Aktivierungskante des Taktsignals clkM als Referenz für Daten aktiviert wird, die auf die ersten Daten folgen, auf denen eine Burst-Lesetätigkeit oder eine Burst-Schreibtätigkeit durchgeführt wurde.

[0591] In **Fig.** 78 ist gezeigt, daß eine Aktivierung des Signals CDE für zweite Daten als Reaktion auf die Aktivierung des Taktsignals clkM zu dem Zeitpunkt t5 durchgeführt wird.

[0592] Bei einem Betrieb eines Burstmodus gilt dieses auch für Mehrdaten, die den ersten Daten folgen, auf eine ähnliche Weise, wenn die Mehrdaten gelesen oder geschrieben werden.

[0593] Weiterhin wird zu einem Zeitpunkt t6, wenn der Vorladebefehl PRE als Reaktion auf die Aktivierung des Taktsignals clkM gegeben wird, eine Vorladetätigkeit gestartet, und als Reaktion geht das Signal RD auf L und zusätzlich wird das Spaltenverriegelungssignal ZCE auf den H-Pegel zurückgesetzt.

[0594] Es sei angemerkt, daß, während in der oben dargestellten Beschreibung ein Fall genommen ist, in dem ein Spezifikationswert der Zeit tRCD = 20 ns ist, die vorliegende Erfindung nicht auf solch einen Fall begrenzt ist. Folglich kann die vorliegende Erfindung auf ähnliche Weise angewendet werden in einem Fall, in dem der Spezifikationswert der Zeit tRCD auf einen vorgeschriebenen Wert gesetzt ist, wenn das Taktsignal clkM irgendwie kleiner als 1 dividiert durch eine ganze Zahl mal dem Spezifikationswert ist.

[0595] **Fig.** 79 ist ein Bild, das eine Konfiguration eines spaltenbezogenen Steuerabschnitts **1206** darstellt.

[0596] Bezug nehmend auf **Fig.** 79 enthält der spaltenbezogene Steuerabschnitt **1206**: eine Verzögerungsschaltung DL0, die das Signal ZCE empfängt, einen Inverter INV01, der eine Ausgabe der Verzögerungsschaltung DL0 empfängt und invertiert; eine NOR-Schaltung, die eine Ausgabe des Inverters INV01 und das Signal ZCE empfängt; einen Inverter INV02, der das Signal ZCE invertiert; eine AND-Schaltung GAD01, die eine Ausgabe des Inverters INV02 und das Taktsignal clkM empfängt; eine OR-Schaltung GOR01, die eine Ausgabe der NOR-Schaltung NR01 und eine Ausgabe der AND-Schaltung GAD01 empfängt; und eine AND-Schaltung GAD02, die eine Ausgabe mit der OR-Schaltung GOR01 und das Signal RD empfängt, zum Ausgeben des Signals CDE.

[0597] Wenn das Signal ZCE = H-Pegel ist, ist das Signal CDE inaktiv (auf dem L-Pegel), unabhängig von einem Pegel des Signals clkM. Ein Einmalpuls mit einer Pulsbreite entsprechend der Verzögerungszeit der Verzögerungsschaltung DL0 wird als das Signal CDE als Reaktion auf ein Abfallen des Signals ZCE ausgegeben. Da das Signal RD auf dem H-Pegel während einer Periode des Burstlesens ist, wie oben beschrieben wurde, wird das Taktsignal clkM als das Signal CDE ausgegeben, während das Signal ZCE auf dem L-Pegel ist.

[Zunahme der Leistung bei dem Schreiben-mit-Autovorladebetrieb]

[0598] **Fig.** 80 ist ein Zeitablaufdiagramm, das Zeiten darstellt, zu denen ein Schreiben mit Autovorladetätigkeit im Stand der Technik durchgeführt wird. In **Fig.** 80 ist ein Fall gezeigt, bei dem eine Burstlänge zum Beispiel 2 beträgt.

[0599] In **Fig.** 80 ist ein Schreiben-mit-Autovorladebetrieb mit einer Burstlänge von 2 in einem System mit solch einer relativ niedrigen Geschwindigkeit wie

ein Zyklus tCLK = 30 ns des Taktsignals clkM gezeigt.
[0600] Zu einem Zeitpunkt t0, wenn der Schreiben-mit-Autovorladebefehl Schreiben-AP eingegeben wird, geht ein Referenzsignal (ein Flag-Signal) WT, das "in Schreibbetrieb" anzeigt, in einen aktiven Zustand, in dem es während einer Periode von zwei Takten bleibt.
[0601] Dann startet ein Vorladebetrieb automatisch mit einer Aktivierungskante (eine Pulskante in dem Übergang auf den H-Pegel) des Taktsignals clkM zu einem Zeitpunkt t1, wenn eine Zeit von zwei Takten als eine Referenz abläuft.
[0602] Als Resultat wird verlangt, daß zu einer Zeit, zu der der ACT-Befehl beim nächsten Mal gegeben wird, eine Zeit zu einem Zeitpunkt t2 ist, wenn ein Takt abläuft nach einer Aktivierungszeit des Taktsignals clkM zu dem Zeitpunkt t1.
[0603] Wenn jedoch der Taktzyklus des Taktsignals clkM ausreichend groß ist, tritt ein Fall auf, in dem ein Zeitspielraum bis zu dem nächsten ACT-Befehl, der nach dem Start einer Vorladetätigkeit gegeben ist, zu einem Ausmaß größer als notwendig zunimmt.
[0604] **Fig.** 81 ist ein schematisches Blockschaltbild, das eine Konfiguration zum Steuern einer Schreibtätigkeit darstellt, die in dem DRAM-Steuerabschnitt **42b** in der Konfiguration des in **Fig.** 76 gezeigten integrierten Logik-DRAM **1000** enthalten ist und herausgezogen ist.
[0605] Bezug nehmend auf **Fig.** 81 erzeugt ein Schreibsteuerabschnitt **1300** ein Signal zum Steuern einer Schreibtätigkeit, zum Beispiel ein Signal zum Bestimmen einer Aktivierungszeit eines Schreibtreibers.
[0606] Ein Burststeuerabschnitt **1302** erzeugt ein Signal zum Steuern einer Burstschreibtätigkeit in einer Schreibtätigkeit, und ein Schreibsteuerabschnitt **1300** erzeugt ein Schreibsteuersignal gemäß der Steuerung des Burststeuerabschnitts **1302**.
[0607] Eine WT-Erzeugerschaltung **1304** zwingt ein Flagsignal WT zum Anzeigen "im Zustand der Schreibtätigkeit" in einen aktiven Zustand als Reaktion auf die Aktivierung eines Schreibsteuersignals, die durch den Schreibsteuerabschnitt **1300** durchgeführt wird. Die WT-Erzeugerschaltung **1304** erzwingt weiter, daß das Signal WT in den inaktiven Zustand (L-Pegel) als Reaktion auf ein Burstendsignal BEND geht, das die Beendigung der Burstschreibtätigkeit anzeigt, das von dem Burststeuerabschnitt **1302** ausgegeben ist.
[0608] Die zeilenbezogene Steuerschaltung **1310** gibt ein zeilenbezogenes Steuersignal, zum Beispiel das Signal SPR zum Aktivieren einer Vorladetätigkeit als Reaktion auf die Deaktivierung des Signals WT aus.
[0609] **Fig.** 82 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebs einer in **Fig.** 81 gezeigten schreibbezogenen Steuerschaltung.
[0610] Bei den in **Fig.** 82 gezeigten Tätigkeiten wird die Beschreibung eines Falls gegeben, in dem ein Zyklus tCLK des Taktsignals clkM länger als eine Zeit (tWR + tRP) ist, was die Summe eines Zeitspielraums tWR für eine Schreibtätigkeit und eines Zeitspielraums tPR zum Durchführen einer Vorladetätigkeit ist.
[0611] Zu einem Zeitpunkt t0 wird ein Schreib-AP-Befehl zum Befehlen einer Schreiben-mit-Autovorladetätigkeit als Reaktion auf eine Aktivierungskante des Taktsignals clkM gegeben.
[0612] Als Reaktion darauf wird ein Schreibsteuersignal von der Schreibsteuerschaltung **1300** ausgegeben, und das Signal WT, das von der WT-Erzeugerschaltung **1304** ausgegeben ist, geht auf den H-Pegel über. Das Burstendsignal BEND auf einem aktiven Pegel, was die Beendigung einer Bursttätigkeit in einem Fall einer Burstlänge von 2 anzeigt, wird von der Burststeuerschaltung **1302** als Reaktion auf eine Aktivierungskante des Taktsignals clkM zu einem Zeitpunkt t2 ausgegeben.
[0613] In der WT-Erzeugerschaltung **1304** wird das Signal WT in einen inaktiven Zustand (auf dem L-Pegel) zu einem Zeitpunkt t3 gezwungen, wenn eine Zeitlänge tWR nach der Aktivierung des Signals BEND abläuft.
[0614] Als Reaktion darauf gibt die zeilenbezogene Steuerschaltung **1310** ein Steuersignal SPR zum Starten einer Vorladetätigkeit aus.
[0615] Da die Vorladetätigkeit beendet ist, wenn eine Zeitlänge tRP nach einem Zeitpunkt t3 abläuft, wird es möglich, den ACT-Befehl zu einem Zeitpunkt t4 der nächsten Taktaktivierungskante folgend auf eine Aktivierungskante des Taktsignals clkM zu einem Zeitpunkt t2 auszugeben.
[0616] Folglich ist es möglich in einem Fall, in dem das Taktsignal clkM auf einer vergleichsweise niedrigen Geschwindigkeit ist, die Zahl der Takte in einer Periode zu verringern von dem Zeitpunkt, wenn der Schreiben-mit-Autovorladebefehl Schreiben-AP gegeben wird, bis zu der Zeit, an der der nächste ACT-Befehl gegeben wird.
[0617] Es sei angemerkt, daß bei der oben dargestellten Beschreibung ein Fall aufgenommen wird, der Schreiben-mit-Autoaufladetätigkeit, es ist auch möglich, die Zahl der Takte in einer Periode von der Zeit, zu der der Lesen-mit-Autovorladebefehl Lesen-AP gegeben wird, bis zu der Zeit, zu der der ACT-Befehl gegeben werden kann, zu verringern, indem eine ähnliche Konfiguration bei einer Lesen-mit-Autovorladetätigkeit vorgesehen wird.

[Konfiguration, die mit Autoauffrischtätigkeit verknüpft ist]

[0618] Zum Beispiel wird bei einer Konfiguration eines 64 Mbit SDRAM des Standes der Technik eine Autoauffrischtätigkeit durchgeführt, indem vier Bänke simultan aktiviert werden. In diesem Fall wird die Auffrischtätigkeit für jede 4096 Bit zu einer Zeit in einer Bank durchgeführt.
[0619] Wenn die Zahl von gleichzeitig aufzufrischenden Speicherzellen zunimmt, nimmt jedoch ein

Spitzenstromwert zu, der bei dem Autoauffrischen verbraucht wird. Daher fällt in einem System, bei dem es hart ist, einen Entkopplungskondensator großer Kapazität sicherzustellen, der Leistungsquellenpegel an dem Spitzenstrom stark ab, und daher tritt eine hohe Möglichkeit einer Fehlfunktion in dem DRAM auf.

[0620] **Fig.** 83 ist ein schematisches Blockschaltbild zum Beschreiben einer Konfiguration, die einen Autoauffrischbetrieb zum Verhindern einer Fehlfunktion in solch einer Auffrischtätigkeit steuert.

[0621] Eine in **Fig.** 83 gezeigte Konfiguration ist in dem DRAM-Steuerabschnitt **42b** von **Fig.** 76 enthalten. Bezug nehmend auf **Fig.** 83 gibt ein Autoauffrischzähler **1400** ein internes Adreßsignal QAD [0:10] zum Spezifizieren einer Speicherzellenzeile, an der eine Auffrischtätigkeit durchgeführt wird, und ein Signal QBA [0:1] zum Spezifizieren einer Bank, in der ein Selbstauffrischen in dem Autoauffrischbetriebsmodus durchgeführt wird, aus.

[0622] Der Autoauffrischzähler **1400** enthält: einen Auffrischadreßzähler **1410** zum Erzeugen eines Adreßsignals QAD [0:10] und einen Auffrischbankzähler 1420 zum Empfangen der höchsten Adresse QAD [10], die von dem Auffrischadreßzähler ausgegeben ist, zum Ausgeben einer Bankadresse QBA [0:1] zum Spezifizieren einer Bank, die aufgefrischt wird.

[0623] Eine Auffrischbankspezifizierungsschaltung **1430** empfängt das Signal QBA [0:1] von dem Auffrischbankzähler **1420** zum Aktivieren eines internen RAS-Signals entsprechend einer Bank, an der die Autoauffrischtätigkeit durchgeführt wird, aus den internen RAS-Signalen RAS_A, RAS_B, RAS_C und RAS_D entsprechend den entsprechenden vier Bänken gemäß der Steuerung des Modusregisters **50**.

[0624] Zum Beispiel werden gemäß der Spezifikation für das Modusregister **50** Betriebsfälle spezifiziert: ein Fall, in dem all die internen RAS-Signale entsprechend der vier Bänke simultan aktiviert sind; ein Fall, in dem zwei Bänke zu einer Zeit aktiviert sind; und ein Fall, in dem jede Bank als eine Einheit aktiviert ist.

[0625] Der Selbstauffrischzeitgeber **1440** gibt ein Triggersignal für ein internes RAS-Signal, das durch die Auffrischbankspezifizierungsschaltung **1430** ausgegeben ist.

[0626] Daher wird es durch Vorsehen des Auffrischbankzähler **1420** und der Auffrischbankspezifizierungsschaltung **1430** möglich, nicht nur einen Modus zu spezifizieren, indem Autoauffrischen simultan an den vier Bänken durchgeführt wird, sondern auch Autoauffrischen an zwei Bänken zu einer Zeit oder an jeder Bank durchzuführen.

[0627] Es sei angemerkt, daß, während es möglich ist zu steuern, welche Zahl von Bänken als eine Einheit beim Durchführen des Autoauffrischens durch Eingeben eines Befehls des Modusregistersatzes von Kombinationen von Adreßsignalen angenommen wird, wie oben beschrieben wurde, es auch möglich ist, Bänke zu spezifizieren, auf denen eine Auffrischtätigkeit simultan durchgeführt wird durch Zugreifen auf einen Adreßraum, der speziell in dem in **Fig.** 76 gezeigten integrierten Logik-DRAM zugeordnet ist.

[0628] Es sei angemerkt, daß, während in der oben gegebenen Beschreibung die Konfiguration aufgenommen ist, bei der Gebiete, die simultan aufgefrischt werden, alle von einer Bank als eine Einheit spezifiziert sind, ein Spitzenstrom ebenfalls auf ähnliche Weise unterdrückt werden kann in einem Fall, in dem zum Beispiel vier Bänke simultan aufgefrischt werden, indem die Zahl von Speicherzellen auf die Hälfte oder ein Viertel verringert wird, die pro Bank simultan aufzufrischen sind.

[0629] **Fig.** 84 ist ein Diagramm zum Beschreiben eines Effekts des Verringerns eines Spitzenstroms in dem in **Fig.** 83 beschriebenen Autoauffrischbetrieb.

[0630] Das heißt, wenn Autoauffrischen an zwei Bänken zu einer Zeit oder an jeder Bank anstelle einer Autoauffrischtätigkeit an vier Bänken zu einer Zeit durchgeführt wird, kann der Spitzenwert eines Betriebsstroms in dem DRAM-Abschnitt gemäß dieser Ordnung verringert werden.

[0631] Es sei angemerkt, daß bei der oben dargestellten Beschreibung ein Zyklus T0 des Selbstauffrischzeitgeber **1440** gemäß der Zahl von Speicherzellen geändert wird, die simultan aufgefrischt werden.

[0632] Wenn zum Beispiel ein Zyklus, in dem vier Bänke zu einer Zeit aufgefrischt werden, T0 ist, wird ein Zyklus als T0/2 in einem Fall angenommen, in dem zwei Bänke zu einer Zeit autoaufgefrischt werden. Weiterhin wird der Zyklus als T0/4 in einem Fall angenommen, in dem jede Bank getrennt autoaufgefrischt wird.

[0633] Indem das getan wird, selbst wenn die Zahl von Bänken, die simultan autoaufzufrischen sind, verringert wird, kann ein Zyklus, in dem eine Auffrischtätigkeit in einer individuellen Speicherzelle durchgeführt wird, im wesentlichen das gleiche Intervall sein.

[Konfiguration des Leistungsquellenabschneidemodus]

[0634] **Fig.** 85 ist ein schematisches Blockschaltbild, das eine Konfiguration darstellt, die mit einer Erzeugerschaltung **1100** des internen Leistungsquellenpotentials verknüpft ist, wie in **Fig.** 76 gezeigt ist.

[0635] Die Konfiguration der Erzeugerschaltung **1100** des internen Leistungsquellenpotentials, die in **Fig.** 85 gezeigt ist, kann einen Strom verringern, der in einer Ruhetätigkeit verbraucht wird, wie unten beschrieben wird.

[0636] Wenn das System in einen Leistungsquellenabschneidemodus gemäß einer Kombination von externen Steuersignalen geht, wie später beschrieben wird, wird ein von dem DRAM-Steuerabschnitt **42b** ausgegebenes Signal SCUT auf den H-Pegel gesetzt.

[0637] Bezug nehmend auf **Fig.** 85 enthält die Erzeugerschaltung **1100** des internen Leistungsquellenpotentials: eine Konstantstromquelle **1500** zum Erzeugen eines Referenzpotentials VBIASL; eine Referenzpotentialerzeugerschaltung **1510** zum Empfangen der Ausgabe VBIASL von der Konstantstromquelle **1500** zum Erzeugen eines ersten Referenzpotentials; eine Komparatorschaltung **1512**, die eine Ausgabe der Referenzpotentialerzeugerschaltung **1510** an einem Minuseingangsknoten davon empfängt; und einen P-Kanal-MOS-Transistor TP11, der zwischen einem externen Leistungsquellenpotential Ext.Vdd und einem Knoten N3 vorgesehen ist, zum Liefern eines Speicherzellenfeldleistungsquellenpotentials Vccs, dessen Gate mit einem Ausgangsknoten N1 der Komparatorschaltung **1412** verbunden ist. Der Knoten N3 ist mit einem Pluseingangsknoten der Komparatorschaltung **1512** verbunden.

[0638] Die Erzeugerschaltung **1100** des internen Referenzpotentials enthält weiter: eine Referenzpotentialerzeugerschaltung **1520** zum Empfangen einer Ausgabe der Konstantstromquelle **1500** zum Erzeugen eines zweiten Referenzpotentials; eine Komparatorschaltung **1522**, die das von der Referenzpotentialerzeugerschaltung **1520** ausgegebene zweite Referenzpotential an einem Minuseingangsknoten davon empfängt; einen P-Kanal-MOS-Transistor TP12, der zwischen dem externen Leistungsquellenpotential Ext.Vdd und einem Knoten N4 vorgesehen ist, zum Ausgeben eines Potentials Vcc, das an eine Peripherieschaltung des DRAM-Abschnitts geliefert wird, und dessen Gate mit einem Ausgangsknoten N2 der Komparatorschaltung **1422** verbunden ist; und einen N-Kanal-MOS-Transistor TN11, der zwischen dem Knoten N2 und einem Massepotential vorgesehen ist und das Signal SCUT an seinem Gate empfängt. Eine Peripherieschaltung des DRAM-Abschnitts steuert eine Auswahltätigkeit eines Speicherzellenfeldes und die oben beschriebenen Auswahltätigkeiten.

[0639] Es sei angemerkt, daß die Komparatorschaltung **1512** mit der Leistungsquelle Ext.Vdd durch einen Transistor TP21 beliefert wird, der das Signal SCUT an seinem Gate empfängt, und weiter mit dem Massepotential durch einen N-Kanal-MOS-Transistor TN21 beliefert wird, der ein Signal/SCUT an einem Gate empfängt, das ein invertiertes Signal des Signals SCUT ist.

[0640] Weiterhin ist ein P-Kanal-MOS-Transistor TP34, der das Signal /SCUT an seinem Gate empfängt, zwischen dem Knoten N1 und dem Leistungsquellenpotential Ext.Vdd vorgesehen. Wenn das Signal /SCUT auf den L-Pegel geht, geht der Transitor TP34 in einen leitenden Zustand zum Anheben eines Gatepotentials des Transistors TP11 auf "H" und zwingt den Transistor TP11 in einen abgetrennten Zustand.

[0641] Auf ähnliche Weise wird die Komparatorschaltung **1522** mit dem Leistungsquellenpotential Ext.Vdd durch einen Transistor TP21 beliefert, der das Signal/SCUT an seinem Gate empfängt, und weiter mit dem Massepotential durch einen Transistor TN22 beliefert, der ein Signal/SCUT an seinem Gate empfängt.

[0642] Weiterhin ist ein Transistor TN31, der das Signal VBIASL, das von der Konstantstromquelle **1500** ausgegeben ist, empfängt, als eine Konstantstromquelle tätig, die einen Konstantstrom an eine Schaltung **1550** liefert.

[0643] Es sei angemerkt, daß die Schaltung **1550** irgendeine Schaltung ist, solange die Schaltungen welche sind, an die ein Konstantstrom geliefert wird auf der Grundlage des Konstantpotentials VBIASL, das von der Konstantstromquelle **15** ausgegeben ist, und nebenbei nicht auf eine der Erzeugerschaltungen **1100** des internen Leistungsquellenpotentials begrenzt sind, und sie können ebenfalls in der Peripherieschaltungsanordnung in dem DRAM-Abschnitt tätig sein.

[0644] Ein N-Kanal-MOS-Transistor TN32, der das Signal SCUT an seienm Gate empfängt, ist zwischen einem Knoten N5, der mit dem Referenzpotential VBIASL beliefert wird, das von solch einer Konstantstromquelle **1500** geliefert wird, und dem Massepotential vorgesehen.

[0645] Die Erzeugerschaltung **1100** des internen Leistungsquellenpotentials enthält weiter: eine Vbb-Erzeugerschaltung **1600** zum Erzeugen eines Referenzpotentials Vbb; einen Detektor **1610** zum Überwachen eines Substratpotentials zum Steuern der Vbb-Erzeugerschaltung **1600**; einen P-Kanal-MOS-Transistor TP33, der zwischen dem Massepotential und einem Ausgangsknoten der Vbb-Erzeugerschaltung vorgesehen ist und das Massepotential an einem Gate empfängt; eine Vpp-Erzeugerschaltung **1630** zum Erzeugen eines verstärkten Potentials höher als das externe Leistungsquellenpotential Ext.Vdd, zum Beispiel eines verstärkten Potentials Vpp, das als ein Aktivierungspotential einer Wortleitung oder ähnliches benutzt wird; einen Detektor **1640** zum Überwachen eines Pegels des verstärkten Potentials Vpp zum Steuern der Vpp-Erzeugerschaltung **1630**; und einen N-Kanal-MOS-Transistor TN41, der zwischen einem Ausgangsknoten der Vpp-Erzeugerschaltung und der Leistungsquelle Ext.Vdd vorgesehen ist und das Potential Ext.Vdd an seinem Gate empfängt.

[0646] Nebenbei, die Erzeugerschaltung **1100** des internen Leistungsquellenpotentials enthält: eine Selbstplattenpotentialerzeugerschaltung **1650**, die das externe Leistungsquellenpotential Ext.Vdd und das Massepotential empfängt, um tätig zu sein, und eine Selbstplatte einer Speicherzelle MC mit einem Selbstplattenpotential beliefert (entgegengesetztes Elektrodenpotential eines Speicherzellenkondensators); und eine Erzeugerschaltung **1660** eines Bitleitungsausgleichspotentials zum Erzeugen eines Ausgleichspotentials VB1 auf dem Bitleitungspaar.

[0647] Als nächstes wird die Beschreibung eines Betriebs in einem Leistungsquellenabschneidemo-

dus gegeben.

[0648] Unter den oben beschriebenen Schaltungen, die nicht nur die Komparatorschaltungen **1512** und **1522** enthalten, sondern auch die erste und die zweite Referenzpotentialerzeugerschaltung **1510** und **1520**, wird verursacht, daß die Detektoren **1610** und **1540** und die Schaltung **1550** in einen inaktiven Zustand zusammen mit der Aktivierung des Signals SCUT gehen. Der Knoten N1 wird deaktiviert, da der Transistor TP34 in einen leitenden Zustand geht, und dadurch geht der Transistor TP11 in einen abgeschnittenen Zustand.

[0649] Zu dieser Zeit geht auch der Transistor TN32 in einen leitenden Zustand zum Führen des Knotens N5 auf das Massepotential.

[0650] Dann wird bewirkt, daß die Vbb-Erzeugerschaltung **1600**, die Vpp-Erzeugerschaltung **1630**, die Selbstplattenpotentialerzeugerschaltung **1650** und die Erzeugerschaltung **1660** des Bitleitungsausgleichspotentials in einen inaktiven Zustand zusammen mit der Aktivierung des Signals SCUT gehen.

[0651] Wenn das System den Leistungsquellenabschneidemodus betritt und bewirkt wird, daß die Schaltungen in der Erzeugerschaltung **1100** des internen Leistungsquellenpotentials einen inaktiven Zustand betreten, wie oben beschrieben wurde, nehmen die Potentialpegel der Schaltungen allmählich aufgrund des Vorhandenseins eines Übergangsleckens und anderer Gründe ab, die in dem Substrat vorhanden sind.

[0652] Ein Potentialpegel eines Ausgangsknotens der Vbb-Erzeugerschaltung **1600** nähert sich jedoch einem Pegel, der durch einen Schwellenwert Vth1 definiert ist, wobei der Schwellenwert Vth1 der des Transistors TP33 ist. Ein Potentialpegel eines Ausgangsknotens der Vpp-Erzeugerschaltung **1530** konvergiert zu einem Pegel von (Ext.Vdd – Vth2), worin Vth2 ein Schwellenwert des Transistors TN41 ist, da der Transistor TN41 vorhanden ist.

[0653] Weiterhin in Hinblick auf eine Leistungsquelle für die Peripherieschaltungsanordnung geht die Komparatorschaltung **1522** in einen inaktiven Zustand. Da jedoch der Transistor TN11 leitend wird und dadurch ein Gatepotential des P-Kanal-MOS-Transistors TP12 das Massepotential annimmt, wird das Potential Vccp das Leistungsquellenpotential Ext.Vdd.

[0654] Wenn die wie oben beschriebene Konfiguration angenommen wird, wird ein Pfad für einen Durchgangsstrom abgeschnitten, und dadurch wird der verbrauchte Strom verringert, und andererseits bleiben die Potentiale Vpp und Vccp auf einem aktiven Pegel; daher ist es möglich, aus dem Leistungsquellenabschneidemodus herauszutreten durch Eingeben eines speziellen Befehls, selbst nachdem der Leistungsquellenabschneidemodus betreten ist.

[0655] **Fig.** 86 ist ein Zeitablaufdiagramm zum Beschreiben einer Tätigkeit (Eintritt) des Eintretens in den Leistungsquellenabschneidemodus und einer Tätigkeit (Austritt) des Herauskommens aus dem Leistungsquellenabschneidemodus, wie oben beschrieben wurde.

[0656] Zu einem Zeitpunkt t0 wird eine Kombination eines Modusregistersetzbefehls MRS und eine Adresse V0 zum Eintreten in den Leistungsquellenabschneidemodus eingegeben.

[0657] Als Reaktion hierauf geht das Signal SCUT zum Steuern des Leistungsquellenabschneidens auf den aktiven Pegel (H-Pegel).

[0658] Als Reaktion auf die Aktivierung des Signals SCUT gehen die Schaltungen in der Erzeugerschaltung **1100** des internen Leistungsquellenpotentials in einen inaktiven Zustand.

[0659] Dann wird in den DRAM-Steuerabschnitt **42b** aufgenommen, daß das Signal CKE in einem inaktiven Zustand (L-Pegel) als Reaktion auf einen Anstieg des Signals clkM zu einem Zeitpunkt t1 ist. Zu dieser Zeit erzwingt der DRAM-Steuerabschnitt **42b** ein Taktpuffertrennsignal CBDA zur Benutzung bei der Deaktivierung eines Taktpuffers **44** zum Eintritt in einen aktiven Zustand (H-Pegel), wenn das Signal SCUT bereits auf dem H-Pegel ist.

[0660] Indem das getan wird, wird selbst der Taktpuffer inaktiv zum weiteren Verringern des verbrauchten Stroms.

[0661] Weiterhin werden Puffer für andere Eingangssignale in einen inaktiven Zustand gezwungen: welche der I/O-Puffer **52**, die Eingangspuffer **40** und **46** und andere Puffer sind.

[0662] Als nächstes wird eine Beschreibung einer Austrittstätigkeit aus dem Leistungsquellenabschneidemodus gegeben.

[0663] Als Reaktion darauf, daß das Signal CKE auf dem H-Pegel zu einem Zeitpunkt t2 ist, geht das Takttrennsignal CBDA auf den L-Pegel in asynchroner Weise.

[0664] Der Taktpuffer wird als Reaktion auf das Takttrennsignal CBDA aktiviert, und geht auf den L-Pegel.

[0665] Darauffolgend werden der Modusregistersetzbefehl und ein Adreßsignal, das für den Austritt aus dem Modus zugeordnet ist, an eine Aktivierungskante des Taktsignals clkM an einem Zeitpunkt t3 eingegeben, und als Reaktion auf die Eingabe wird das Signal SCUT auf den L-Pegel zurückgesetzt.

[0666] Nachdem der Austritt aus dem Leistungsquellenabschneidemodus bewirkt ist und die interne Leistungsquelle stabilisiert ist, wird die Eingabe von Befehlen, die nicht der Modusregistersetzbefehl sind, möglich.

[0667] Zum Beispiel gibt es in dem SDRAM des Standes der Technik, selbst wenn das Signal auf den L-Pegel gibt, keine Möglichkeit dafür, daß der Taktpuffer **44** deaktiviert wird, wenn nicht in dem Selbstauffrischtätigkeitsmodus.

[0668] Im Vergleich damit wird bei einem integrierten Logik-DRAM, der sich auf die vorliegende Erfindung bezieht, wenn das Signal CKE auf den L-Pegel geht, der Taktpuffer ebenfalls deaktiviert, wenn er in dem Leistungsquellenabschneidemodus ist, wodurch

ermöglicht wird, daß der verbrauchte Strom weiter verringert wird.

[0669] Es sei angemerkt, daß Adressen, die für den Eintritt in und den Austritt aus dem Stromquellenabschneidemodus zugeordnet sind, auch die gleichen zueinander sein können.

[0670] Es sei angemerkt, daß in der oben dargestellten Beschreibung ein Fall aufgenommen ist, in dem das Potential Vccp das Potential Ext.Vdd während einer Periode ist, in der das Signal SCUT auf dem H-Pegel ist. Es kann jedoch eine Konfiguration angenommen werden, bei der die Konstantstromquelle **1500**, die Referenzpotentialerzeugerschaltung **1520** und die Komparatorschaltung **1522** während einer Periode betätigt werden, in der das Signal SCUT auf dem H-Pegel ist, zum Steuern des Potentials Vccp auf einen gewünschten Pegel.

[0671] **Fig.** 87 ist ein Schaltbild, das eine Beispielskonfiguration des Taktpuffers **44** zeigt.

[0672] In einem Fall, wie er in **Fig.** 87 gezeigt ist, enthält der Taktpuffer **44** eine Komparatorschaltung **1700**, die mit einem Anschluß **16** an ihrem Pluseingangsknoten verbunden ist und ein Referenzpotential Vrefl an ihrem Minuseingangsknoten empfängt, wobei bewirkt werden kann, daß der Taktpuffer in einem inaktiven oder aktiven Zustand ist, wenn ein Pegel des Referenzpotentials Vrefl gemäß einem Pegel des Signals CBDA gesteuert wird, wenn das Taktpuffertrennsignal CBDA in einen aktiven Zutand geht. Eine Ausgabe der Komparatorschaltung **1700** wird von der Pufferschaltung **1702** als das interne Taktsignal int.CLK ausgegeben.

[0673] **Fig.** 88 ist ein Schaltbild, das eine andere Schaltungskonfiguration des Taktpuffers **44** darstellt.

[0674] Bei der in **Fig.** 88 gezeigten Konfiguration enthält der Taktpuffer **44**: eine Komparatorschaltung **1700**, die ein Signal von dem Anschluß **16** an einem Pluseingangsknoten davon und das Referenzpotential Vrefl an einem Minuseingangsknoten davon empfängt; eine NOR-Schaltung **1710**, die mit dem Anschluß **16** an einem Eingangsknoten davon verbunden ist und das Signal/EN an dem anderen Eingangsknoten davon empfängt; und eine OR-Schaltung **1726**, die Ausgaben der Komparatorschaltung **1700** und der NOR-Schaltung **1710** empfängt zum Ausgeben des internen Taktsignals int.CLK. In dem Taktpuffer **44** empfängt der Komparator **1700** das Leistungsquellenpotential durch einen P-Kanal-MOS-Transistor **1722**, der das Signal SCUT an seinem Gate empfängt, und das Massepotential durch einen N-Kanal-MOS-Transistor **1724**, der das Signal SCUT an seinem Gate empfängt. Der Taktpuffer **44** enthält weiter einen N-Kanal-MOS-Transistor **1728**, der zwischen einem Ausgangsknoten der Komparatorschaltung **1700** und dem Massepotential vorgesehen ist und das Signal SCUT an seinem Gate empfängt.

[0675] Daher ist es nur notwendig, daß während einer Periode, während der das Signal SCUT auf dem L-Pegel ist, das externe Taktsignal Ext.CLK durch die Komparatorschaltung **1700** eingegeben wird, wohingegen während einer Periode, während der das Signal auf dem H-Pegel in dem Leistungsabschneidemodus ist, das Referenzpotential Vrefl gesteuert wird zum Zwingen der Komparatorschaltung **1700** in einen inaktiven Zustand, und das Taktsignal wird durch die NOR-Schaltung **1710** eingegeben.

[0676] Solange das Signal/EN auf dem H-Pegel ist, hat das externe Taktsignal Ext.CLK keine Möglichkeit, daß es in dem Inneren durch die NOR-Schaltung **1710** aufgenommen wird.

[0677] Durch Annehmen einer Konfiguration, wie sie oben beschrieben wurde, kann der integrierte Logik-DRAM **1000** seine Leistung aufrecht erhalten und den Leistungsverbrauch verringern, selbst wenn das Taktsignal relativ langsam abläuft.

Zehntes Beispiel

[0678] Eine Konfiguration eines integrierten Logik-DRAM des zehnten Beispiels ist fundamental die gleiche wie die des in **Fig.** 76 gezeigten integrierten Logik-DRAM **1000**.

[0679] In dem integrierten Logik-DRAM **1000** des zehnten Beispiels ist jedoch eine Konfiguration angenommen, wie unten beschrieben wird, bei der ein Niveau der internen Stromquellenspannung variabel gemäß einem Zyklus eines Eingangstaktes ist.

[0680] Durch Annehmen einer solchen Konfiguration wird in einem System, das bei einer niedrigen Geschwindigkeit tätig sein kann, ein interner Spannungspegel derart geändert, daß eine Geschwindigkeit eines internen Betriebs des DRAM-Abschnitts in Übereinstimmung mit einer Systemgeschwindigkeit verringert wird, wodurch eine Verringerung in dem Leistungsverbrauch bei einer gegebenen Taktfrequenz erzielt werden kann.

[0681] Es sei angemerkt, daß die folgende Beschreibung nicht nur begrenzend auf einen Fall angewendet werden kann, indem eine Logikschaltungsanordnung und ein DRAM gemischt auf einem Chip hergestellt werden, sondern auch auf einen Fall, in dem nur der DRAM auf einen Chip integriert ist.

[0682] **Fig.** 89 ist ein konzeptuelles Blockschaltbild, das eine Konfiguration eines Systems darstellt, in dem ein integrierter Logik-DRAM **1000** verwendet ist.

[0683] In dem System sind eine Mikrosteuereinheit MCU und ein integrierter Logik-DRAM **1000** vom taktsynchronen Typ verbunden.

[0684] Ein Taktsignal CLK, ein Befehlssignal und ein Adreßsignal werden von der Mikrosteuereinheit MCU geliefert, und das Liefern/Empfangen von Daten wird zwischen der Mikrosteuereinheit MCU und dem integrierten Logik-DRAM **1000** durchgeführt.

[0685] **Fig.** 90 ist ein Diagramm, das Betriebsfrequenzen eines Speichers darstellt, die gemäß Anwendungen in dem System, wie es in **Fig.** 89 gezeigt ist, benötigt werden. Eine Bandbreite einer Datenübertragung, die für einen Speicher verlangt wird, unterscheidet sich gemäß einer Anwendung. In **Fig.** 90

ist ein Fall eines tragbaren Telefons gedacht.

[0686] Zum Beispiel in einem Fall, in dem nur Stimmdaten verarbeitet werden, wird eine CLK-Frequenz des DRAM-Abschnitts niedrig gesetzt, während in einem Fall, in dem ein Videosignal (hauptsächlich Decodieren von Videodaten) oder eine Videokonferenz (hauptsächlich Codierung von bewegenden Bildern) verarbeitet werden, wird eine Taktfrequenz des DRAM-Abschnitts hoch gesetzt, wodurch die Verarbeitungsfähigkeit vergrößert wird.

[0687] Auf solche Weise kann durch Ändern einer Betriebstaktfrequenz des DRAM-Abschnitts gemäß einer Anwendung der Leistungsverbrauch des DRAM-Abschnitts unterdrückt werden.

[0688] **Fig.** 91 ist ein konzeptuelles Blockschaltbild zum Beschreiben einer Konfiguration, die eine Betriebsgeschwindigkeit eines DRAM-Abschnitts gemäß einer in **Fig.** 90 gezeigten Taktfrequenz ändern kann.

[0689] Bezug nehmend auf **Fig.** 91 enthält die Takterzeugerschaltung **44**: eine Frequenzerfassungsschaltung **1800**, die eine Frequenz des externen Taktsignals Ext.CLK erfaßt, das an den Anschluß **16** gegeben wird, zum Ausgeben des Steuersignals Ctrl.

[0690] Die Erzeugerschaltung **1100** des internen Leistungsquellenpotentials ändert einen Pegel des Leistungsquellenpotentials, das davon ausgegeben ist, gemäß dem Steuersignal Ctrl.

[0691] In **Fig.** 91 ist eine Konfiguration gezeigt, in der die Leistungsquelle VDCp, die an den Logikabschnitt von der Leistungsquellenschaltung **1100** durch eine Leistungsquellenleitung Lvcc geliefert wird, gesteuert wird. Zum Beispiel, wenn eine Frequenz des externen Taktsignals Ext.CLK abnimmt, nimmt ein Pegel der internen Leistungsquelle ab, während, wenn die Frequenz zunimmt, der interne Leistungsquellenpegel zunimmt.

[0692] **Fig.** 92 ist ein schematisches Blockschaltbild, das eine andere Konfiguration darstellt, die ein internes Leistungsquellenpotential gemäß dem externen Taktsignal Ext.CLK steuert.

[0693] Eine Takterzeugerschaltung **44** ähnlich zu dem Fall von **Fig.** 91 enthält eine Frequenzerfassungsschaltung **1800** zum Erfassen einer Frequenz des externen Taktsignals Ext.CLK.

[0694] Eine Erzeugerschaltung **1100** des internen Leistungsquellenpotential enthält: eine interne Leistungsquellenschaltung **1810** zum Erzeugen eines Leistungsquellenpotentials VDCs für einen Speicherzellenfeldabschnitt; eine Leistungsquellenleitung Lvc2 zum Übertragen des Potentials Vdcs zu dem Speicherzellenfeldabschnitt; eine interne Leistungsquellenschaltung **1820** zum Erzeugen eines internen Leistungsquellenpotentials VDCp, das an die Logikschaltungsanordnung geliefert wird; und eine Leistungsquellenleitung Lvc1 zum Übertragen des Potentials VDCp an den Logikschaltungsanordnungsabschnitt.

[0695] Es sei angemerkt, daß es auch möglich ist, einen Pegel des internen Leistungsquellenpotentials gemäß einer Frequenz des externen Taktsignals Ext.CLK nur durch das Leistungsquellenpotential VDCs für das Speicherzellenfeld zu steuern.

[0696] Bei einer in **Fig.** 92 gezeigten Konfiguration werden ebenfalls die Pegel VDCs und VDCp der internen Leistungsquelle gemäß einer Frequenz des externen Taktsignals Ext.CLK gesetzt.

[0697] **Fig.** 93 ist ein Diagramm zum Beschreiben einer Steuertätigkeit eines internen Leistungsquellenpotentials, wie es in **Fig.** 91 oder 92 gezeigt ist, worin die Abzisse eine Taktfrequenz auf einer logarithmischen Skala darstellt, die darauf vorgesehen ist, und die Ordinate einen internen Leistungsquellenpotentialpegel auf einer linearen Skala darstellt, der darauf vorgesehen ist.

[0698] Ein interner Leistungsquellenpotentialpegel wird gemäß der Zunahme in der Taktfrequenz gesetzt. Ein interner Leistungsquellenpotentialpegel nimmt mit der Zunahme der Taktfrequenz zu, während im Gegensatz dazu ein internes Leistungsquellenpotential mit der Abnahme in der Taktfrequenz abnimmt.

[0699] Durch Annehmen einer Konfiguration, wie sie oben beschrieben wurde, ändert sich ein interner Leistungsquellenpotentialpegel gemäß einer Taktfrequenz, und dadurch können der DRAM-Abschnitt und der Logik-Abschnitt bei einer Geschwindigkeit entsprechend einer Taktfrequenz tätig sein, was eine Realisation eines niedrigen Leistungsverbrauchs ermöglicht.

[0700] **Fig.** 94 ist ein schematisches Blockschaltbild zum Beschreiben einer Konfiguration einer in **Fig.** 91 oder 92 gezeigten Frequenzerfassungsschaltung **1800**.

[0701] Bezug nehmend auf **Fig.** 94 enthält die Frequenzerfassungsschaltung **1800**: einen Taktpuffer **1801**, der das externe Taktsignal Ext.CLK von dem Anschluß **16** empfängt; Verzögerungsschaltungen **1802.1** bis **1802.4**, die eine Ausgabe des Taktpuffers empfangen und in Reihe in einer Kette geschaltet sind; einen Phasenkomparator **1804.1**, der Ausgaben des Taktpuffers **1801** und der Verzögerungsschaltung **1802.1** empfängt; einen Phasenkomparator **1804.2**, der Ausgaben des Taktpuffers **1801** und der Verzögerungsschaltung **1802.2** empfängt, einen Phasenkomparator **1804.3**, der Ausgaben des Taktpuffers **1801** und der Verzögerungsschaltung **1802.3** empfängt; und einen Phasenkomparator **1804.4**, der Ausgaben des Taktpuffers **1801** und der Verzögerungsschaltung **1802.4** empfängt.

[0702] Die Erzeugerschaltung **1100** des internen Leistungsquellenpotentials ändert ein erzeugtes Leistungsquellenpotential durch sich selbst gemäß dem Signal Ctrl, das von den Phasenkomparatoren **1804.1** bis **1804.4** ausgegeben wird.

[0703] **Fig.** 95 ist ein Zeitablaufdiagramm zum Beschreiben eines Betriebs der in **Fig.** 94 gezeigten Frequenzerfassungsschaltung.

[0704] Bezug nehmend auf **Fig.** 95 werden Signale DT1 bis DT4 auf einem aktiven Pegel von den Verzö-

gerungsschaltungen **1802.1** bis **1802.4**, wobei jedes Signal um eine vorbestimmte Zeit zwischen einem und dem nächsten verzögert wird, sequentiell als Reaktion auf die Aktivierungskante des externen Taktsignals Ext.CLK zu einem Zeitpunkt t0 ausgegeben.

[0705] Hinsichtlich der Tätigkeiten der Phasenkomparatoren **1804.1** bis **1804.4** hier gehen die Tätigkeiten auf diese Weise: Phasendifferenzen des externen Taktsignals Ext.CLK und Ausgaben der Verzögerungsschaltungen **1802.1** bis **1802.4** werden durch die Phasenkomparatoren **1804.1** bis **1804.4** verglichen.

[0706] In dem in **Fig.** 95 gezeigten Zustand ist eine Phase des externen Taktsignals Ext.CLK hinter einer Ausgabe DT3 der Verzögerungsschaltung **1802.3** verzögert, und die Phase des externen Taktsignals Ext.CLK ist gegenüber der Ausgabe DT4 der Verzögerungsschaltung **1802.4** verzögert.

[0707] Es ist nur notwendig, daß ein Ausgangspegel der Erzeugerschaltung **1100** des internen Leistungsquellenpotentials gemäß einer Phasendifferenz eingestellt wird, die auf solche weise erfaßt wird.

[0708] Wenn das externe Taktsignal Ext.CLK hinter der Aktivierung der Verzögerungsschaltung **1802.1** in der Phase hergeht, und die Aktivierung des externen Taktsignals Ext.CLK hinter der Aktivierung eines Ausgangssignals DT2 der Verzögerungsschaltung **1802.2** hergeht, ist eine Frequenz höher als in dem in **Fig.** 95 gezeigten Fall.

[0709] **Fig.** 96 ist ein schematisches Blockschaltbild zum Beschreiben einer Konfiguration einer internen Leistungsquellenschaltung in der Erzeugerschaltung **1100** des internen Leistungsquellenpotentials, die durch die Frequenzerfassungsschaltung **1800** gesteuert wird. In **Fig.** 96 werden die Potentiale VDCs und VDCp gemeinsam als Potential VDC bezeichnet.

[0710] Die in **Fig.** 96 gezeigte interne Leistungsquellenschaltung enthält: eine Referenzpotentialerzeugerschaltung **1850**, die ein durch sie selbst erzeugtes Referenzpotential gemäß dem Steuersignal Ctrl von der Frequenzerfassungsschaltung **1800** ändert; eine Komparatorschaltung **1860**, die eine Ausgabe der Referenzpotentialerzeugerschaltung **1850** an ihrem Minuseingangsknoten empfängt; einen P-Kanal-MOS-Transistor TP51, der zwischen einem Leistungsquellenausgangsknoten NC und dem externen Leistungsquellenpotential Ext.Vdd vorgesehen ist und eine Ausgabe der Komparatorschaltung **1860** an einem Gate empfängt; und einen N-Kanal-MOS-Transistor TN51, der zwischen einem Ausgangsknoten der Komparatorschaltung **1860** und dem Massepotential vorgesehen ist und zum Beispiel ein Steuersignal Ctr0, das von dem DRAM-Steuerabschnitt **42b** ausgegeben wird, an seinem Gate empfängt, worin der Knoten NC und ein Pluseingangsknoten der Komparatorschaltung **1860** miteinander verbunden sind.

[0711] In einem Betrieb der in **Fig.** 96 gezeigten internen Leistungsquellenschaltung wird ein Pegel des Knotens N10, der das interne Leistungsquellenpotential ausgibt, gemäß einem Pegel der Referenzpotentialerzeugerschaltung **1850** gesteuert, wobei das Signal Ctr10 auf dem L-Pegel ist. In einem Hochgeschwindigkeitssystem kann jedoch der Pegel des internen Leistungsquellenpotentials auf dem externen Leistungsquellenpotential fixiert werden. Das heißt, es ist nur notwendig, daß in einem Hochgeschwindigkeitssystem ein Potentialpegel des Gates des Transistors T51 von **Fig.** 96 auf 0V fixiert ist, wodurch der Transistor TN51 in einen leitenden Zustand versetzt wird, so daß ein Pegel des internen Leistungsquellenpotentials VDC das externe Leistungsquellenpotential Ext.Vdd annimmt.

[0712] Mit einer Konfiguration, wie sie oben beschrieben angenommen ist, wird es möglich, daß der interne Leistungsquellenpotentialpegel gemäß einer Taktfrequenz zum Verringern des Leistungsverbrauchs geändert wird.

[0713] **Fig.** 97 ist ein schematisches Blockschaltbild, das eine andere Konfiguration zum Steuern des internen Leistungsquellenpotentials darstellt.

[0714] In **Fig.** 91 bis 96 ist die Frequenzerfassungsschaltung **1800** vorgesehen, und ein durch die Erzeugerschaltung **1100** des internen Leistungsquellenpotentials ausgegebener Potentialpegel wird gemäß dem Erfassungsresultat des Frequenzerfassungsschaltung geändert.

[0715] Im Kontrast dazu wird in **Fig.** 97 das Steuersignal Ctrl für die interne Quellenschaltung **1100** durch Abgeben eines Befehlssignals und eines Adreßsignals an das Modusregister **50** von außen gesteuert, und gemäß dem Signal Ctrl wird ein Pegel der internen Leistungsquelle geändert. Die anderen Punkte in der Konfiguration sind die gleichen wie die entsprechenden Punkte in der Konfiguration von **Fig.** 91.

[0716] In Betriebsmodi der drei Arten einschließlich einer niedrigen Geschwindigkeit, einer mittleren Geschwindigkeit und einer hohen Geschwindigkeit, falls sie zur Verfügung stehen, werden zum Beispiel drei Arten von Modusregister den entsprechenden Modi zum Steuern der Modusregister und weiter eines Pegels des internen Leistungsquellenpegels zugeordnet.

[0717] In einem Hochgeschwindigkeitssystem ist zum Beispiel die Steuerung derart möglich, daß ein Pegel des internen Leistungsquellenpotentials hoch ist. Es sei angemerkt, daß in diesem Fall ebenfalls es möglich ist, daß in dem Hochgeschwindigkeitssystem ein Potentialpegel des Gates des Transistors TP51 von **Fig.** 96 auf 0V fixiert ist, wodurch der Transistor TN51 in einen leitenden Zustand gesetzt wird, so daß bewirkt wird, daß ein Pegel des internen Stromquellenpotentials VDC das externe Leistungsquellenpotential Ext.Vdd annimmt.

[0718] Es sei angemerkt, daß der Logikabschnitt und der DRAM-Abschnitt so aufgebaut sein können, daß sie ähnlich zu dem in **Fig.** 92 gezeigten all getrennt gesteuert werden.

[0719] Mit einer angenommenen Konfiguration, wie

sie oben beschrieben wurde, kann ein Pegel des erzeugten internen Leistungsquellenpotentials gemäß einem Steuersignal geändert werden.

[0720] **Fig.** 98 ist ein Speicherabbild zum Beschreiben eines Beispiels einer Zuordnung in einem Speicherraum, wenn eine Mehrzahl von Arten von Betriebsgeschwindigkeitsmodi vorhanden ist.

[0721] Bei dem in **Fig.** 98 gezeigten Beispiel ist ein Speicherraum in Anwendungen für eine niedrige Geschwindigkeit, eine mittlere Geschwindigkeit und eine hohe Geschwindigkeit zuvor unterteilt.

[0722] Zum Beispiel ist eine Anwendung, die einem Raum #AC0 von **Fig.** 98 zugeordnet ist, für eine niedrige Geschwindigkeitsbenutzung, und wenn ein Zugriff auf den Raum durchgeführt wird, wird ein Pegel der internen Leistungsquelle niedrig gesetzt.

[0723] Im Gegensatz dazu ist ein Speicherraum #AC3 für eine hohe Geschwindigkeitsbenutzung, und das interne Leistungsquellenpotential ist gemäß der Auswahl des Speicherraums hoch gesetzt.

[0724] Ein Speicherraum #AC2 ist für eine mittlere Geschwindigkeitsbenutzung, und das interne Leistungsquellenpotential ist auf einen mittleren Wert zwischen jenen der Speicherräume #AC0 und #AC3 gemäß der Auswahl des Speicherraums gesetzt.

[0725] Solch eine Klassifikation eines Speicherraums kann durch das Modusregister **50** gesetzt werden. Weiterhin kann solch eine Klassifikation eines Speicherraums ursprünglich in der Stufe einer Herstellung einer Vorrichtung spezifiziert werden.

[0726] Alternativ kann die Klassifikation eines Speicherraums auch für jede Bank zugeordnet werden.

[0727] In jedem der oben beschriebenen Fälle wird eine Spaltenadresse zu der unteren Adresse zugeordnet, und die minimale Einheit des Speicherraums wird als Zeilenadresse angenommen, wodurch eine Beurteilung, welche der niedrigen, mittleren und hohen Geschwindigkeiten spezifiziert sind, durch Prüfen einer Zeilenadresse möglich wird, wenn ein ACT-Befehl eingegeben wird.

[0728] **Fig.** 99 ist ein schematisches Blockschaltbild zum Beschreiben einer Konfiguration einer Treiberschaltung in einem in **Fig.** 76 gezeigten I/O-Puffer 52.

[0729] Entsprechend einem Ausgangsanschluß Dout sind eine Treiberschaltung **1900** mit einer kleinen Stromtreiberfähigkeit und eine Treiberschaltung **1920** mit einer großen Stromtreiberfähigkeit vorgesehen.

[0730] Eine Stromtreiberfähigkeit kann auf einen gewünschten Wert durch Ändern einer Transistorgröße eines Treibertransistors geändert werden.

[0731] Wenn ein Signal Sslow auf einem H-Pegel ist, dann ist die Treiberschaltung **1920** deaktiviert, und dadurch kann die gesamte Treiberfähigkeit der Treiberschaltungen verringert werden. Wenn andererseits das Signal Sslow auf dem L-Pegel ist, dann sind die beiden Treiberschaltungen **1900** und **1920** aktiviert.

[0732] Wenn daher eine Frequenz des Taktsignals clkM niedrig ist und das Signal Sslow auf dem H-Pegel ist, wird die Treiberschaltung **1920** deaktiviert.

[0733] Wenn andererseits eine Frequenz des Taktsignals clkM hoch ist und das Signal Sslow auf "L" ist, werden die beiden Treiberschaltungen **1900** und **1920** aktiviert.

[0734] Wenn daher eine Frequenz des Taktsignals clkM niedrig ist, wird Ausgabedatentreiben mit einer kleinen Treiberfähigkeit durchgeführt, wohingegen, wenn eine Frequenz des Taktsignals clkM hoch ist, Ausgangsdatentreiben mit einer hohen Treiberfähigkeit durchgeführt werden kann.

[0735] Mit einer angenommenen Konfiguration, wie sie oben beschrieben wurde, kann das Setzen des Signals Sslow durch einen Modusregistersetzbefehl für das Modusregister **50** geschaltet werden, und weiterhin kann eine andere Konfiguration angenommen. werden, bei der ein spezieller Adreßraum zur Steuerbenutzung zum Schalten des Setzens des Signals Sslow zugeordnet wird.

[0736] Eine in **Fig.** 99 gezeigten Treiberschaltung enthält: einen Inverter INV10, der ein auszugebendes Signal empfängt und invertiert; und eine Logikgatterschaltung **1930**, die ein logisches Produkt zwischen einem invertierten Signal des Signals Sslow und einem Ausgangsaktivierungssignal En ausgibt.

[0737] Die Treiberschaltung **1900** enthält: einen Inverter INV11, der das Treibersignal En empfängt; eine NOR-Schaltung NR11, die eine Ausgabe des Inverters INV10 an einem Eingangsknoten davon und eine Ausgabe des Inverters INV11 an einem anderen Eingangsknoten davon empfängt; einen Inverter INV12, der eine Ausgabe der NOR-Schaltung NR11 empfängt und invertiert; eine NAND-Schaltung ND11, die eine Ausgabe des Inverters INV10 an einem Eingangsknoten davon und das Signal En an dem anderen Eingangsknoten davon empfängt; einen Inverter INV13, der eine Ausgabe der NAND-Schaltung ND11 empfängt; und einen P-Kanal-MOS-Transistor TP101 und einen N-Kanal-MOS-Transistor TN101, die in Reihe zwischen dem Leistungsquellenpotential und dem Massepotential geschaltet sind.

[0738] Der Transistor TP101 empfängt eine Ausgabe des Inverters INV12 an seinem Gate, und der Transistor TN101 empfängt eine Ausgabe des Inverters INV13 an seinem Gate. Ein Verbindungsknoten zwischen den Transistoren TP101 und TN101 ist mit dem Anschluß Dout verbunden.

[0739] Die Treiberschaltung **1920** enthält: einen Inverter INV21, der eine Ausgabe des Logikgatters **1930** empfängt; eine NOR-Schaltung NR21, die eine Ausgabe des Inverters INV10 an einem Eingangsknoten davon und eine Ausgabe des Inverters INV11 an dem anderen Eingangsknoten davon empfängt; einen Inverter INV22, der eine Ausgabe der NOR-Schaltung NR21 empfängt und invertiert; eine NAND-Schaltung ND21, die eine Ausgabe des Inverters INV10 an einem Eingangsknoten davon und eine Ausgabe des Logikgatters **1930** an dem anderen Eingangsknoten davon empfängt; einen Inverter INV23, der eine Ausgabe der NAND-Schaltung ND21 emp-

fängt; und einen P-Kanal-MOS-Transistor TP201 und einen N-Kanal-MOS-Transistor TN201, die in Reihe zwischen dem Leistungsquellenpotential und Massepotential geschaltet sind.

[0740] Der Transistor TP201 empfängt eine Ausgabe des Inverters INV22 an seinem Gate, und der Transistor TN201 empfängt eine Ausgabe des Inverters INV23 an seinem Gate. Ein Verbindungsknoten zwischen den Transistoren TP201 und TN201 ist mit dem Anschluß Dout verbunden.

[0741] Mit einer angenommenen Konfiguration, wie sie oben beschrieben wurde, wird der Leistungsverbrauch eines Ausgangspuffers variabel gemäß einer Taktfrequenz, und eine weitere Verringerung im Leistungsverbrauch kann möglich werden.

[0742] **Fig.** 100 ist ein schematisches Blockschaltbild zum Beschreiben einer anderen Konfiguration einer Treiberschaltung in einem in **Fig.** 76 gezeigten I/O-Puffer **52**.

[0743] Die in **Fig.** 100 gezeigte Treiberschaltung enthält einen Inverter INV30, der ein auszugebendes Signal D0 empfängt und invertiert, einen Inverter INV31, der das Treibersignal En empfängt; eine NOR-Schaltung NR31, die eine Ausgabe des In-verters INV30 an einem Eingangsknoten davon und eine Ausgabe des Inverters INV31 an dem anderen Eingangsknoten davon empfängt; einen Inverter INV32, der eine Ausgabe der NOR-Schaltung NR31 empfängt und invertiert; eine Verzögerungsschaltung DL301, die eine Ausgabe der NOR-Schaltung NR31 empfängt, zum Verzögern der Ausgabe um eine vorbestimmte Zeit; und ein zusammengesetztes Logikgatter CG301, das eine Ausgabe der Verzögerungsschaltung DL301 und das Signal Sslow und eine Ausgabe der NOR-Schaltung NR31 und ein Signal/Sslow empfängt. Das zusammengesetzte Logikgatter CG301 gibt ein Resultat einer NOT-OR-Operation zwischen einem ersten logischen Produkt zwischen einer Ausgabe der Verzögerungsschaltung DL301 und dem Signal Sslow und einem zweiten logischen Produkt zwischen einer Ausgabe der NOR-Schaltung NR31 und dem Signal/Sslow aus.

[0744] Die in **Fig.** 100 gezeigte Treiberschaltung enthält weiter: eine NAND-Schaltung ND41, die eine Ausgabe des Inverters INV30 an einem Eingangsknoten und das Signal En an dem anderen Eingangsknoten davon empfängt; einen Inverter INV42, der eine Ausgabe der NAND-Schaltung ND41 empfängt und invertiert, eine Verzögerungsschaltung DL401, die eine Ausgabe der NAND-Schaltung ND41 empfängt, zum Verzögern der Ausgabe um eine vorbestimmte Zeit; und ein zusammengesetztes Logikgatter CG401, das eine Ausgabe der Verzögerungsschaltung DL401. und das Signal/Sslow und eine Ausgabe des Inverters INV42 und das Signal Sslow empfängt. Das zusammengesetzte Logikgatter CG401 gibt ein Resultat einer NOT-AND-Operation zwischen einer ersten logischen Summe zwischen einer Ausgabe der Verzögerungsschaltung DL401 und dem Signal/Sslow und einer zweiten logischen Summe zwischen einer Ausgabe des Inverters INV42 und dem Signal Sslow aus.

[0745] Die in **Fig.** 100 gezeigte Treiberschaltung enthält weiter: einen P-Kanal-MOS-Transistor TP301 und einen N-Kanal-MOS-Transistor TN401, die in Reihe zwischen dem Leistungsquellenpotential und dem Massepotential geschaltet sind; und einen P-Kanal-MOS-Transistor TP302 und einen N-Kanal-MOS-Transistor TN402, die in Reihe zwischen dem Leistungsquellenpotential und dem Massepotential geschaltet sind. Ein Verbindungsknoten zwischen dem Transistor TP301 und dem Transistor TN401 ist mit dem Ausgangsanschluß Dout verbunden, und ein Verbindungsknoten zwischen dem Transistor TP302 und dem Transistor TN402 ist ebenfalls mit dem Ausgangsanschluß Dout verbunden.

[0746] Der Transistor TP301 empfängt eine Ausgabe des Inverters INV32 an seinem Gate, und der Transistor TN402 empfängt eine Ausgabe des Inverters INV41 an seinem Gate. Andererseits empfängt der Transistor TP302 eine Ausgabe des zusammengesetzten Logikgatters CG301 an seinem Gate, und der Transistor TP402 empfängt eine Ausgabe des zusammengesetzten Logikgatters CG401 an seinem Gate.

[0747] Eine Größe des Transistors TP302 ist größer als eine Größe des Transistors TP301, und eine Größe des Transistors TN402 ist größer als eine Größe des Transistors TN401.

[0748] Bei der in **Fig.** 100 gezeigten Treiberschaltung sind, wenn eine Betriebsfrequenz niedrig ist und das Signal Sslow auf dem H-Pegel ist und das Signal/Sslow auf dem L-Pegel ist, die Transistoren TP302 und 402 später in der Treiberzeit als die entsprechenden Transistoren TP301 und TN401 durch Verzögerungszeiten durch die Verzögerungsschaltungen DL301 und DL401.

[0749] Im Gegensatz dazu, wenn eine Betriebsfrequenz hoch ist und das Signal/Sslow auf "H" ist, sind die Transistoren TP302 und 402 praktisch die gleichen wie die entsprechenden Transistoren TP301 und TN401 in der Treiberzeit unabhängig von Verzögerungstätigkeiten der Verzögerungsschaltungen DL301 und DL401.

[0750] Wenn eine Konfiguration, wie oben beschrieben wurde, angenommen wird, wird auch der Leistungsverbrauch eines Ausgangspuffers variabel gemäß einer Taktfrequenz, und eine weitere Verringerung des Leistungsverbrauchs kann möglich sein.

[0751] Obwohl die vorliegende Erfindung im einzelnen beschrieben und dargestellt worden ist, ist klar zu verstehen, daß dieses nur zur Illustration und als Beispiel dient und nicht als Begrenzung genommen werden kann, der Umfang der vorliegenden Erfindung ist nur durch den Inhalt der beigefügten Ansprüche begrenzt.

Patentansprüche

1. Integrierte Halbleiterschaltungsvorrichtung

mit:
einer Anschlussgruppe (**10**, **12**, **14**, **16**), die ein extern geliefertes Steuersignal, Adresssignal und Eingangsdaten empfängt;
einem Speicherzellenfeld (**4**; **54**), das gemäß dem Steuersignal Speicherdaten in einem Gebiet speichert, das durch das Adresssignal spezifiziert ist; und
einer Logikschaltungsanordnung (**8**; **74**), die eine Logikoperation auf den Eingangsdaten und/oder Speicherdaten gemäß mindestens einem von dem Steuersignal, dem Adresssignal und den Eingangsdaten ausführt;
gekennzeichnet durch:
einen Schnittstellenabschnitt (**2**; **36**), der das Steuersignal, das Adresssignal und die Daten von der Anschlussgruppe (**12**, **14**, **16**, **18**) empfängt und das Steuersignal, das Adresssignal und die Eingangsdaten an das Speicherzellenfeld (**4**; **57**) und die Logikschaltungsanordnung (**8**; **74**) überträgt;
einen Datenhalteabschnitt (**6**; **78**, **80**, **82**, **84**, **86**), der mindestens eines von dem Steuersignal, dem Adresssignal und den Eingangsdaten zu der Logikschaltungsanordnung (**8**; **74**) liefert; und
einen internen Schnittstellenabschnitt (**9**; **76**), der mindestens eines von dem Steuersignal, dem Adresssignal und den Eingangsdaten an den Datenhalteabschnitt (**6**; **78**, **80**, **82**, **84**, **86**) liefert, wenn das von dem Schnittstellenabschnitt (**2**; **36**) gelieferte Adresssignal eine vorbestimmte Adresse spezifiziert.

2. Integrierte Halbleiterschaltungsvorrichtung nach Anspruch 1, bei der die interne Schnittstelle (**9**) Daten, die bei der Logikoperation gemäß dem Steuersignal zu bearbeiten sind, zwischen den in dem Speicherzellenfeld (**4**) gespeicherten Speicherdaten und den von der Anschlussgruppe (**10**, **12**, **14**, **16**) gelieferten Eingangsdaten schaltet.

3. Integrierte Halbleiterschaltungsvorrichtung nach Anspruch 2, bei der das Speicherzellenfeld (**4**; **54**) ein Resultat der Logikoperation auf den von der Anschlussgruppe (**10**, **12**, **14**, **16**) gelieferten Eingangsarten in einem Gebiet des Speicherzellenfeldes (**4**; **54**) speichert, das durch das Adresssignal spezifiziert ist.

4. Integrierte Halbleiterschaltungsvorrichtung nach Anspruch 2, bei der die Logikschaltungsanordnung (**8**; **74**) die in dem ersten Gebiet des Speicherzellenfeldes (**4**; **54**), das durch das Adresssignal spezifiziert ist, gespeicherte Speicherdaten ausliest und ein Resultat der Logikoperation auf den Speicherdaten in das erste Gebiet überträgt.

5. Integrierte Halbleiterschaltungsvorrichtung nach Anspruch 2, bei der die Logikschaltungen (**8**; **74**) eine spezifizierte Datenlänge von Speicherdaten sequentiell aus einem Gebiet ausliest, bei dem die spezifizierte Datenlänge an einer Startadresse startet, die von dem Adresssignal spezifiziert ist und das Speicherzellenfeld (**4**; **54**) ein Resultat der Logikoperation auf der spezifizierten Datenlänge von Speicherdaten in das Gebiet mit: der spezifizierten Datenlänge speichert.

6. Integrierte Halbleiterschaltungsvorrichtung nach Anspruch 2, bei der die Logikschaltungsanordnung (**8**; **74**) eine spezifizierte Datenlänge von Speicherdaten sequentiell aus einem ersten Gebiet des Speicherzellenfeldes (**4**; **54**) ausliest mit einer spezifizierten Datenlänge, die an einer ersten Adresse startet, die durch das Adresssignal spezifiziert ist, und das Speicherzellenfeld (**4**; **54**) ein Resultat der Logikoperation auf der spezifizierten Datenlänge von Speicherdaten in ein zweites Gebiet mit der spezifizierten Datenlänge speichert, das an einer zweiten Adresse startet, die von dem Adresssignal spezifiziert ist.

7. Integrierte Halbleiterschaltungsvorrichtung nach Anspruch 1, bei der der Datenhalteabschnitt (**6**; **78**, **80**, **82**, **84**, **86**) ein Register zum Halten von Instruktionen von dem internen Schnittstellen abschnitt (**9**; **76**) ist zum Durchführen der Logikoperation gemäß dem Steuersignal, dem Adresssignal und den Eingangsdaten.

8. Integrierte Halbleiterschaltungsvorrichtung nach Anspruch 7, bei der der Schnittstellenabschnitt (**2**; **36**) eine Erzeugungsschaltung für einen internen Takt (**7**; **44**) aufweist, die ein externes Taktsignal zum Erzeugen eines ersten internen Taktsignales, das als eine Referenz für einen Betrieb des Speicherzellenfeldes (**4**; **54**) dient, und ein zweites internes Taktsignal, das als eine Referenz für einen Betrieb der Logikschaltungsanordnung (**8**; **74**) dient, empfängt.

9. Integrierte Halbleiterschaltungsvorrichtung nach Anspruch 7, bei der die Instruktionen enthalten:
einen Befehl, der einen Betrieb der Logikschaltungsanordnung (**34**) spezifiziert und
Eingangsdaten, die von der Logikschaltungsanordnung (**34**) zu bearbeiten sind, und worin der Datenhalteabschnitt aufweist:
eine erste Halteschaltung (**78**, **80**, **82**), die den Befehl hält;
eine zweite Halteschaltung (**84**), die die Eingangsdaten hält; und
eine dritte Halteschaltung (**86**), die ein Bearbeitungsresultat hält, das durch eine Tätigkeit auf den Eingangsdaten erhalten ist, die durch die Logikschaltungsanordnung (**34**) durchgeführt ist.

10. Integrierte Halbleiterschaltungsvorrichtung nach Anspruch 7, bei der die Instruktionen enthalten:
einen Befehl, der eine Tätigkeit der Logikschaltungsanordnung (**34**) spezifiziert, und
Eingangsdaten, die von der Logikschaltungsanordnung (**34**) zu bearbeiten sind, und worin der Datenhalteabschnitt aufweist:

eine erste Halteschaltung (**78**, **80**, **82**), die den Befehl hält; und
eine zweite Halteschaltung (**84**), die die Eingangsdaten hält und
ein Bearbeitungsresultat einer Tätigkeit auf den Eingangsdaten hält, die von der Logikschaltungsanordnung (**34**) ausgeführt ist.

Es folgen 93 Blatt Zeichnungen

Anhängende Zeichnungen

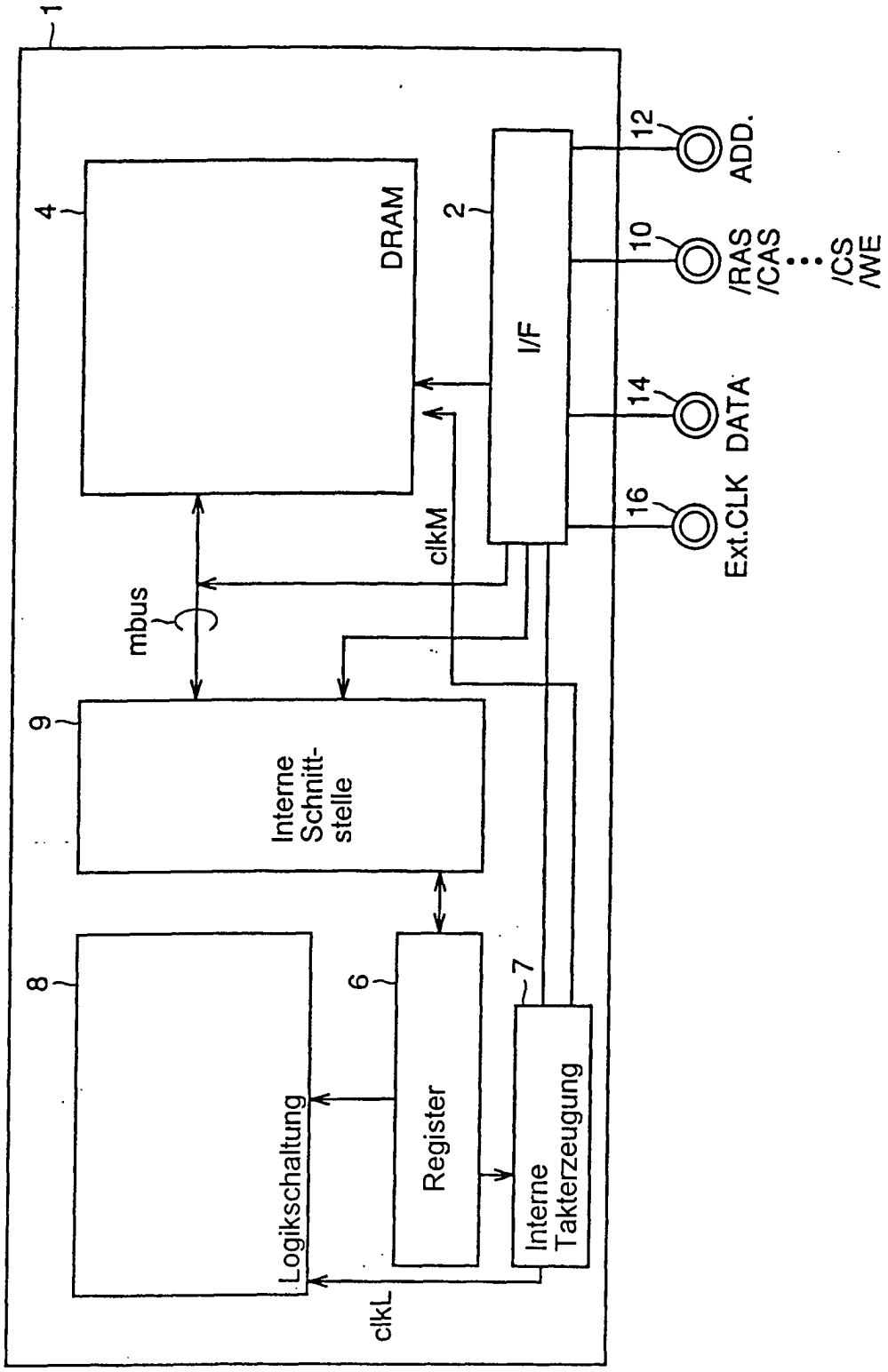


FIG.1

FIG.2

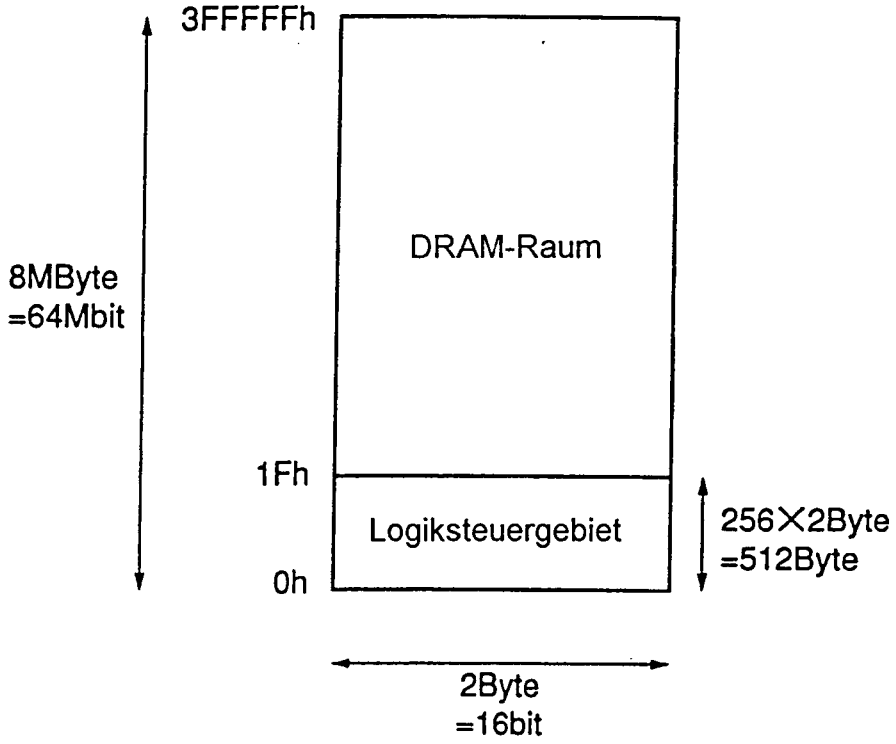

3FFFFFh
8MByte
=64Mbit
DRAM-Raum
1Fh
Logiksteuergebiet
0h
256X2Byte
=512Byte
2Byte
=16bit

FIG.3

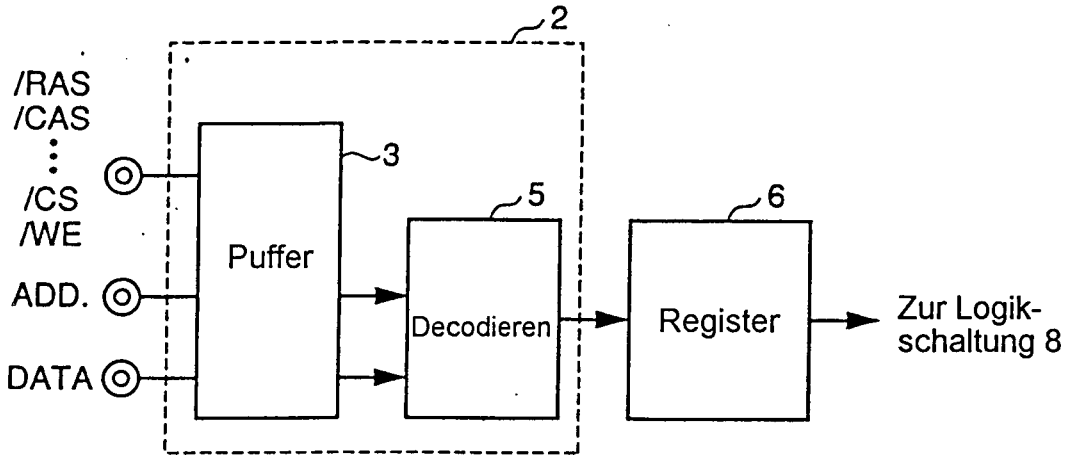

2
/RAS
/CAS
/CS
/WE
ADD.
DATA
Puffer
3
5
Decodieren
6
Register
Zur Logik-
schaltung 8

FIG.4

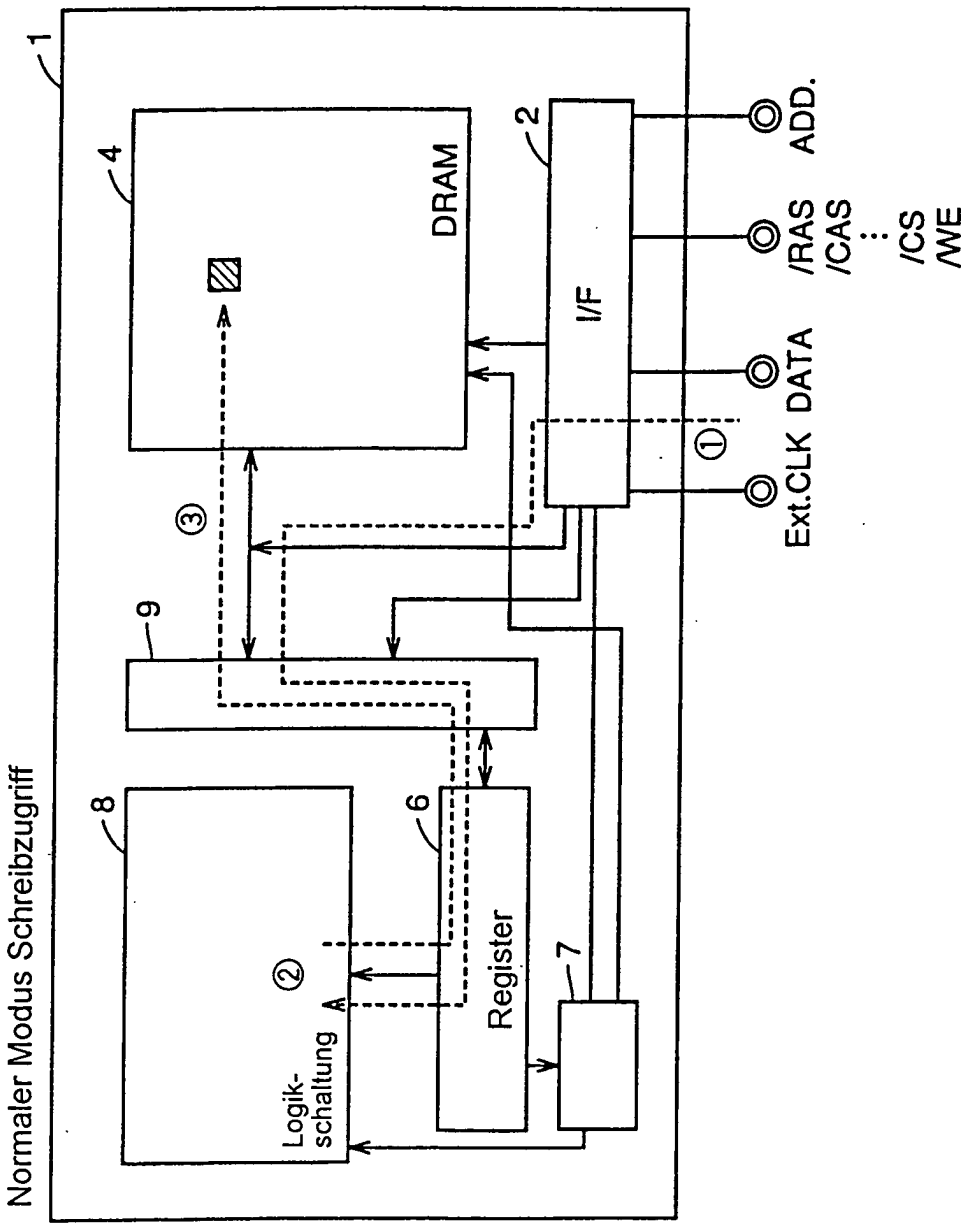
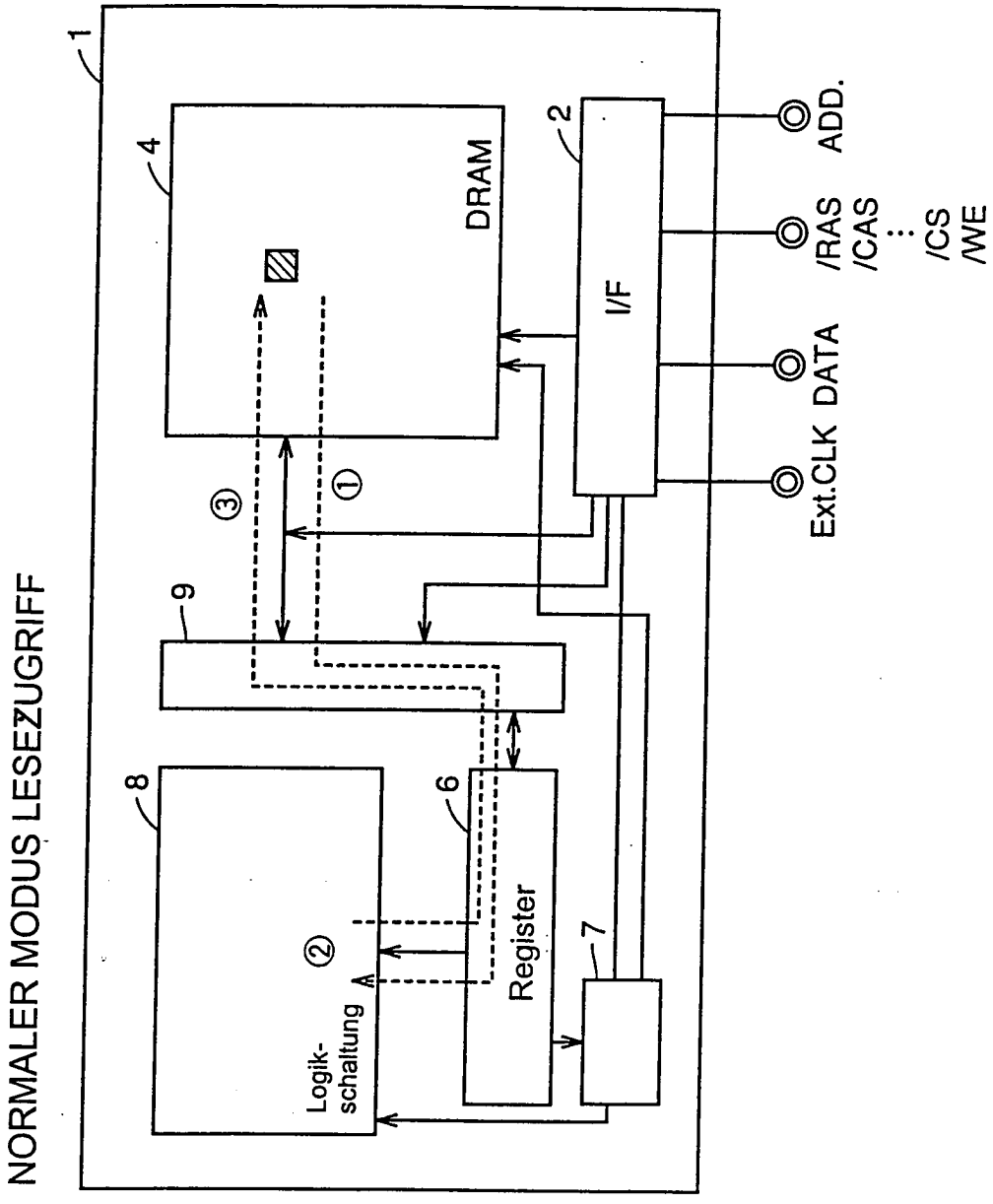

Normaler Modus Schreibzugriff
1
4
DRAM
2
I/F
ADD.
/RAS
/CAS
/CS
/WE
DATA
Ext.CLK
①
②
③
9
8
Logik-
schaltung
6
Register
7

FIG.5


NORMALER MODUS LESEZUGRIFF
1
2
4
DRAM
I/F
ADD.
/RAS
/CAS
/CS
/WE
DATA
Ext.CLK
9
8
6
7
Register
Logik-
schaltung
①
②
③

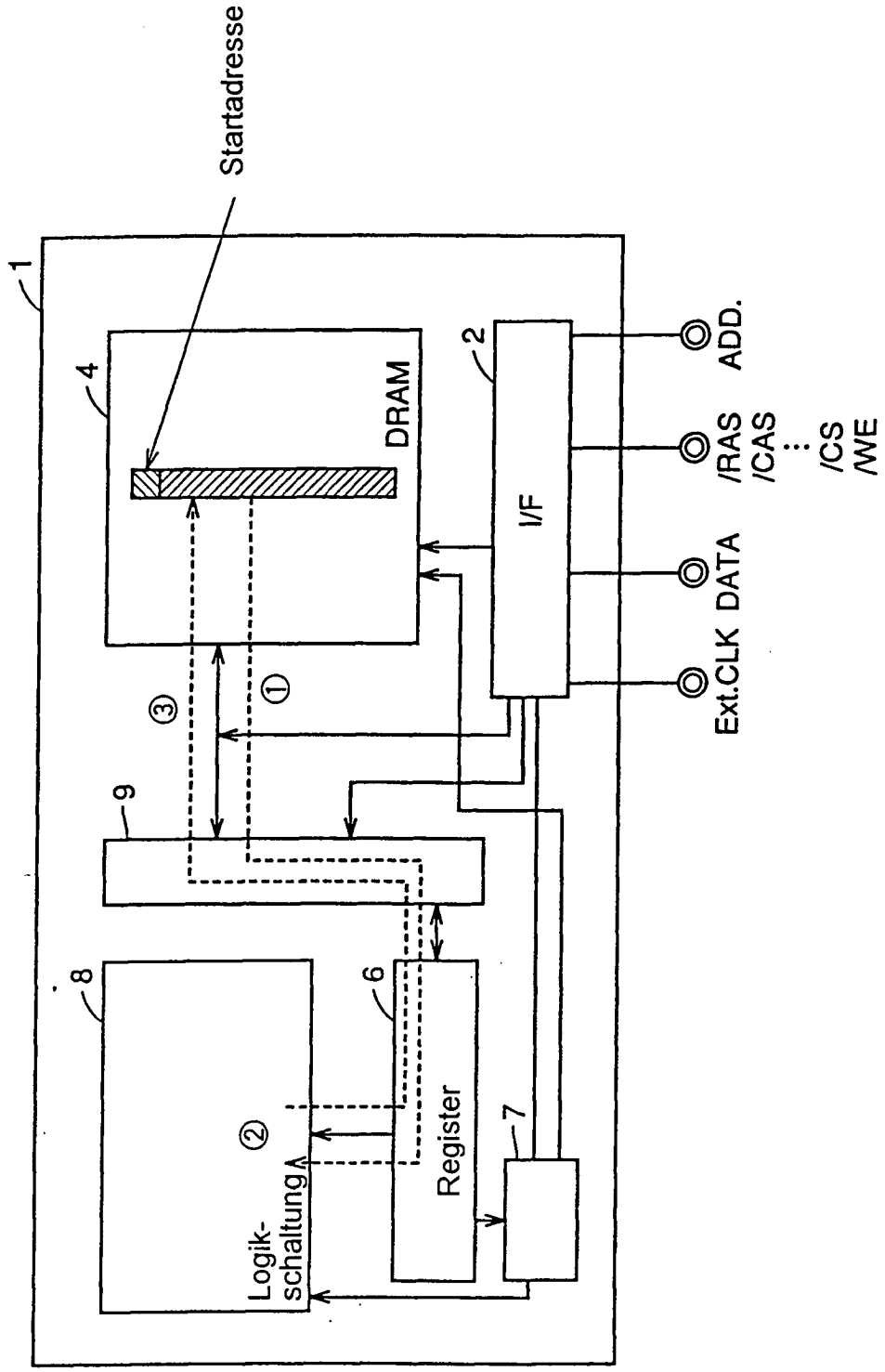

Startadresse
1
2
4
DRAM
I/F
ADD.
/RAS
/CAS
/CS
/WE
DATA
Ext.CLK
①
②
③
6
7
8
9
Register
Logik-
schaltung

FIG.6

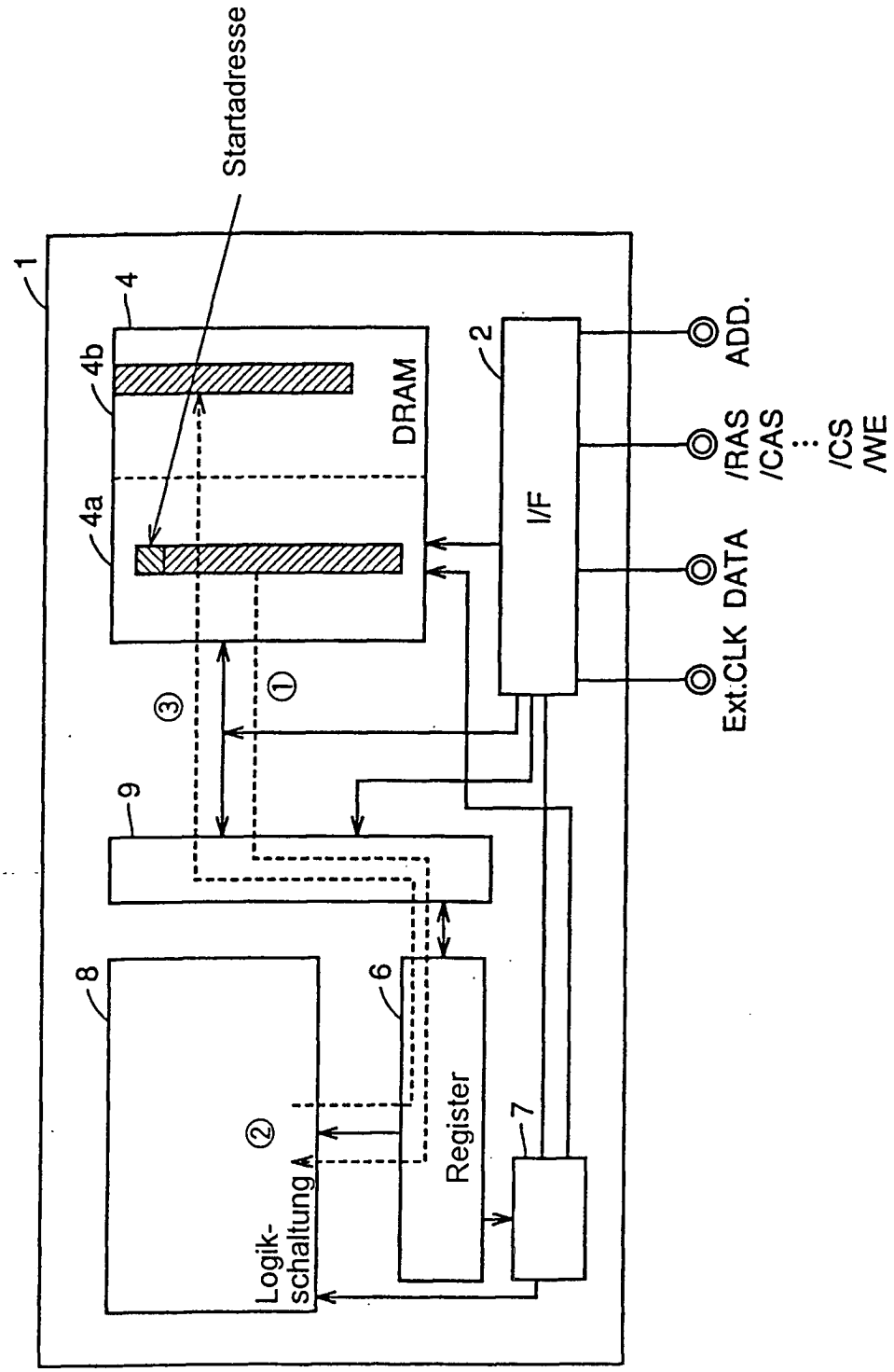

Startadresse
1
4
4b
DRAM
4a
2
I/F
ADD.
/RAS
/CAS
/CS
/WE
Ext.CLK
DATA
①
②
③
9
8
6
7
Logik-
schaltung
Register

FIG.7

FIG.8

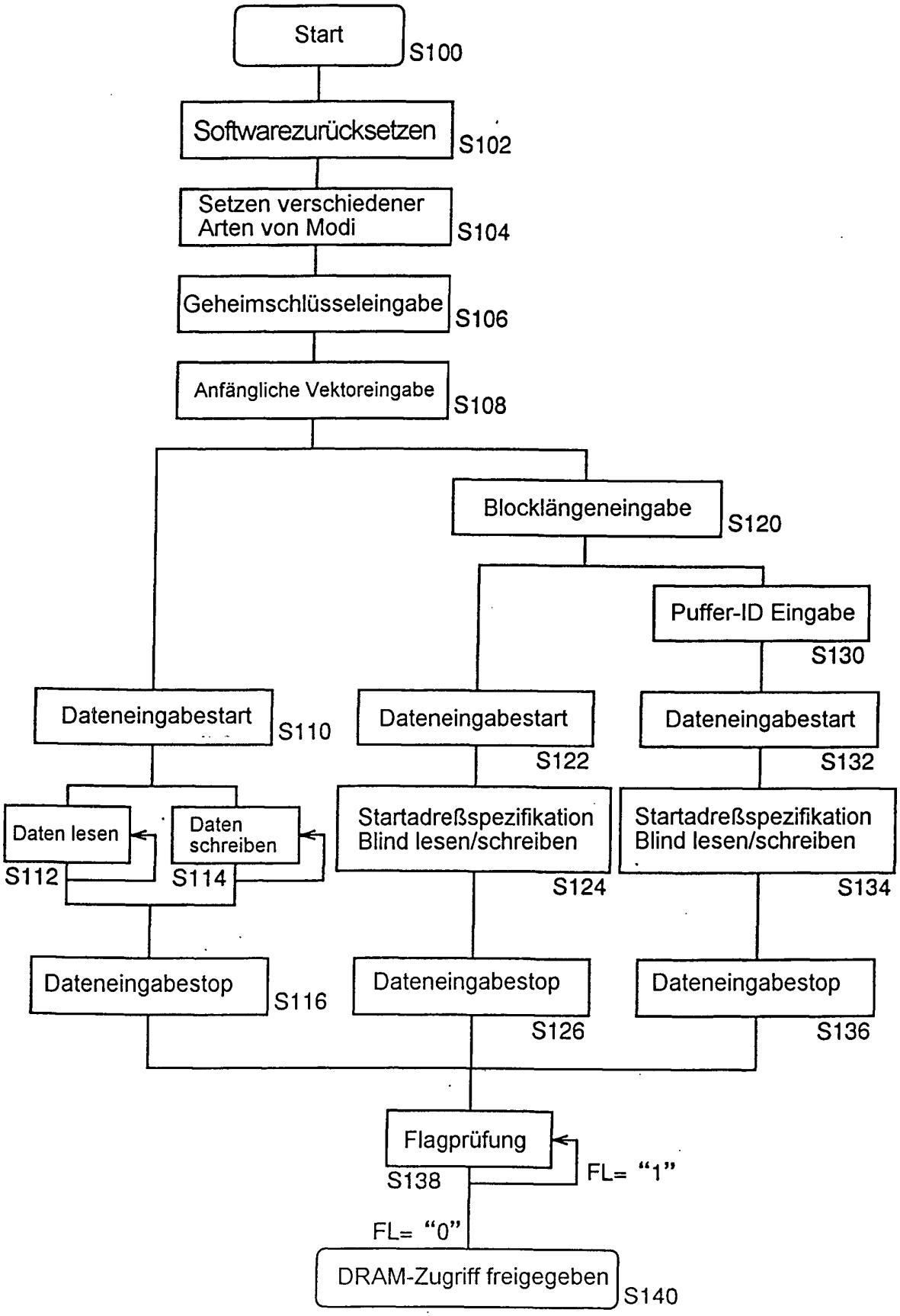
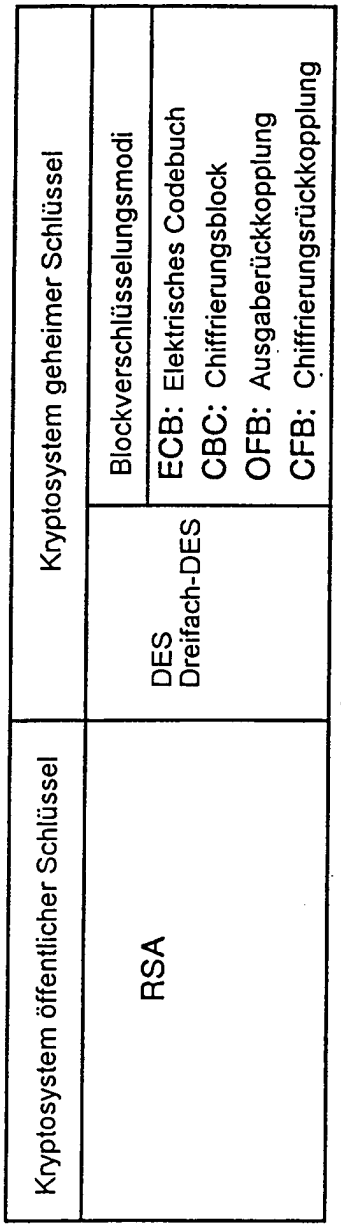
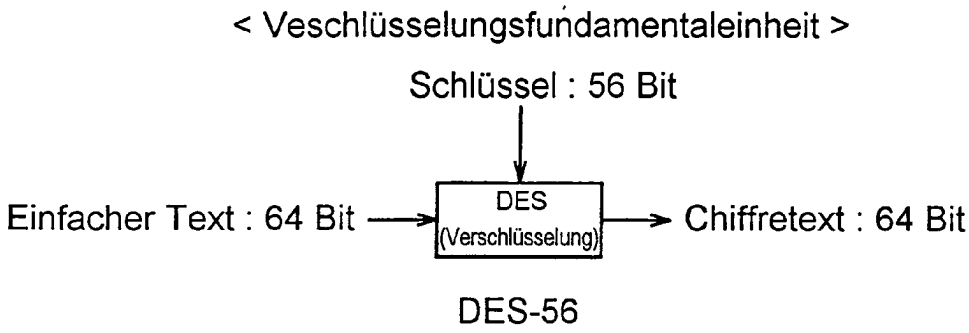

Start
S100
Softwarezurücksetzen
S102
Setzen verschiedener Arten von Modi
S104
Geheimschlüsseleingabe
S106
Anfängliche Vektoreingabe
S108
Blocklängeneingabe
S120
Puffer-ID Eingabe
S130
Dateneingabestart
S110
Dateneingabestart
S122
Dateneingabestart
S132
Daten lesen
S112
Daten schreiben
S114
Startadreßspezifikation Blind lesen/schreiben
S124
Startadreßspezifikation Blind lesen/schreiben
S134
Dateneingabestop
S116
Dateneingabestop
S126
Dateneingabestop
S136
Flagprüfung
S138
FL= "1"
FL= "0"
DRAM-Zugriff freigegeben
S140

FIG.9



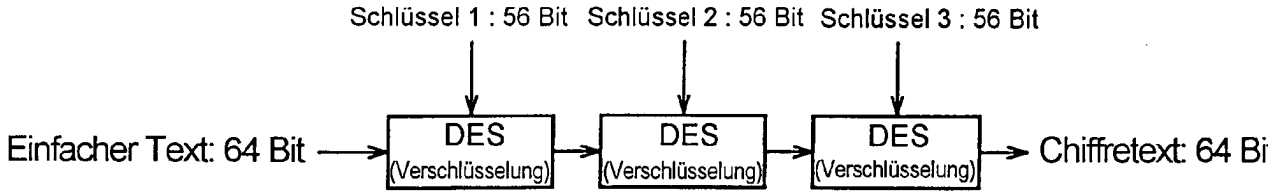
Unterstützte Kryptosysteme

FIG.10



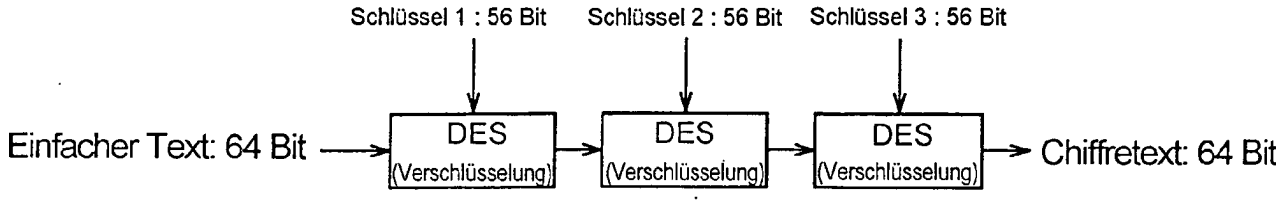
DES-56

FIG.11



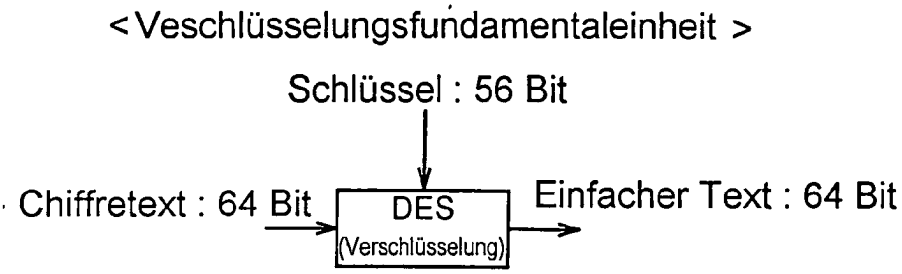
Dreifach-DES-112

FIG.12



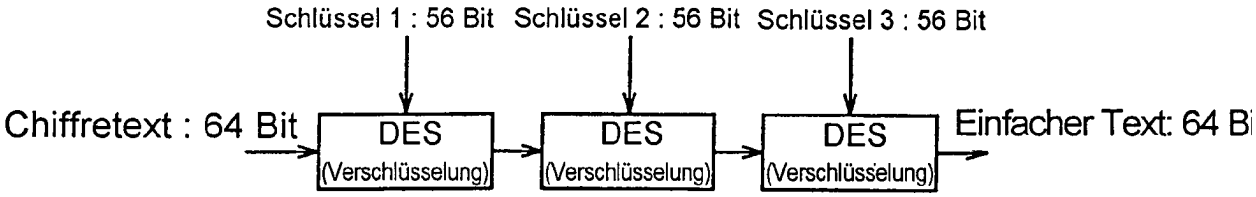
Dreifach-DES-168

FIG.13



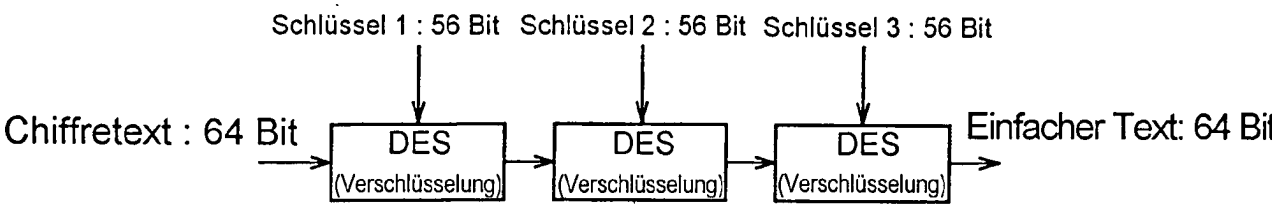
DES-56

FIG.14



Dreifach-DES-112

FIG.15



Dreifach-DES-168

FIG.16

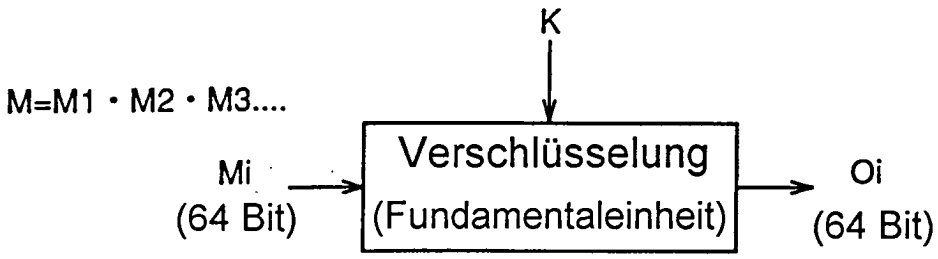

K
M=M1 • M2 • M3....
Mi
(64 Bit)
Verschlüsselung
(Fundamentaleinheit)
Oi
(64 Bit)

FIG.17

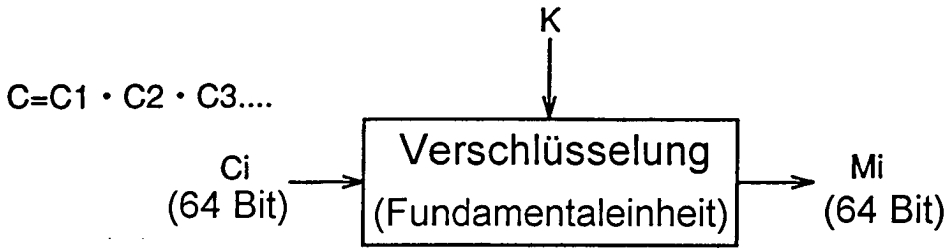
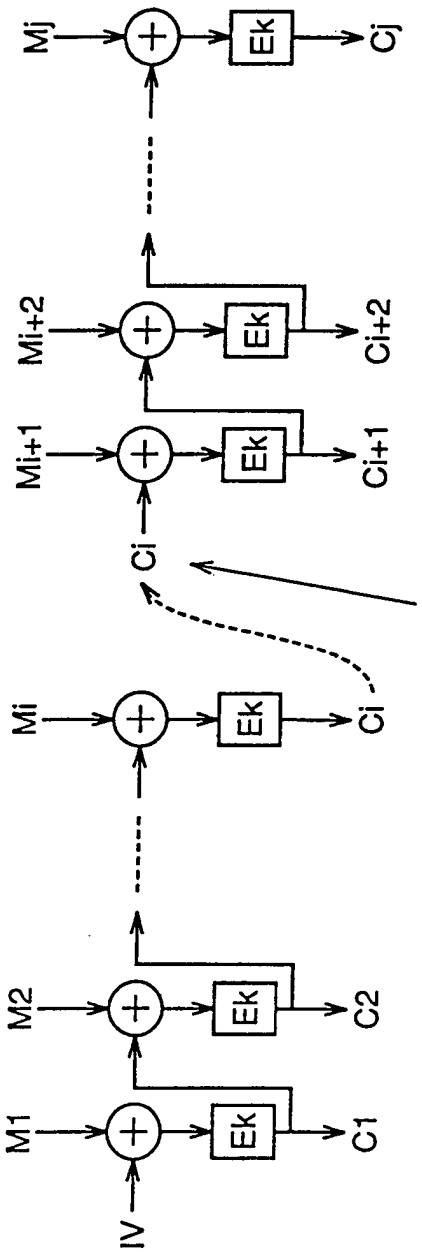

K
C=C1 • C2 • C3....
Ci
(64 Bit)
Verschlüsselung
(Fundamentaleinheit)
Mi
(64 Bit)

FIG.18



Unmittelbar vorangehender Chiffretext Ci wird als Anfangswert benutzt, wenn einfacher Text M länger als die Größe des Registers 6 ist

< Umrisse von Verschlüsselung im CBC-Modus >

FIG.19

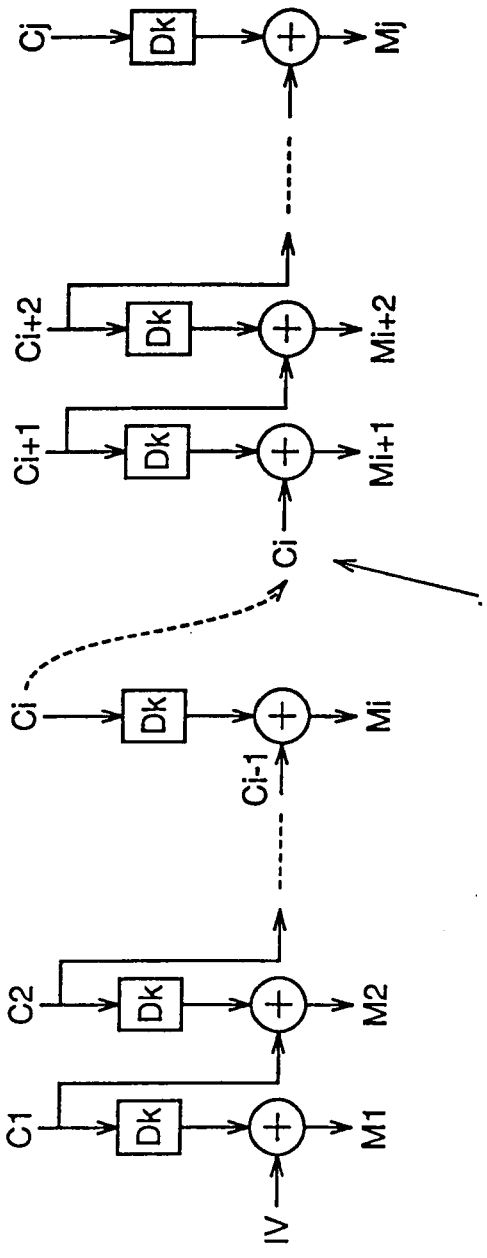

IV
C1
C2
Ci-1
Ci
Ci+1
Ci+2
Cj
Dk
M1
M2
Mi
Mi+1
Mi+2
Mj
Unmittelbar vorangehender Chiffretext Ci wird als Anfangswert benutzt, wenn einfacher Text M länger als die Größe des Registers 6 ist
< Umrisse von Verschlüsselung im CBC-Modus >

FIG.20

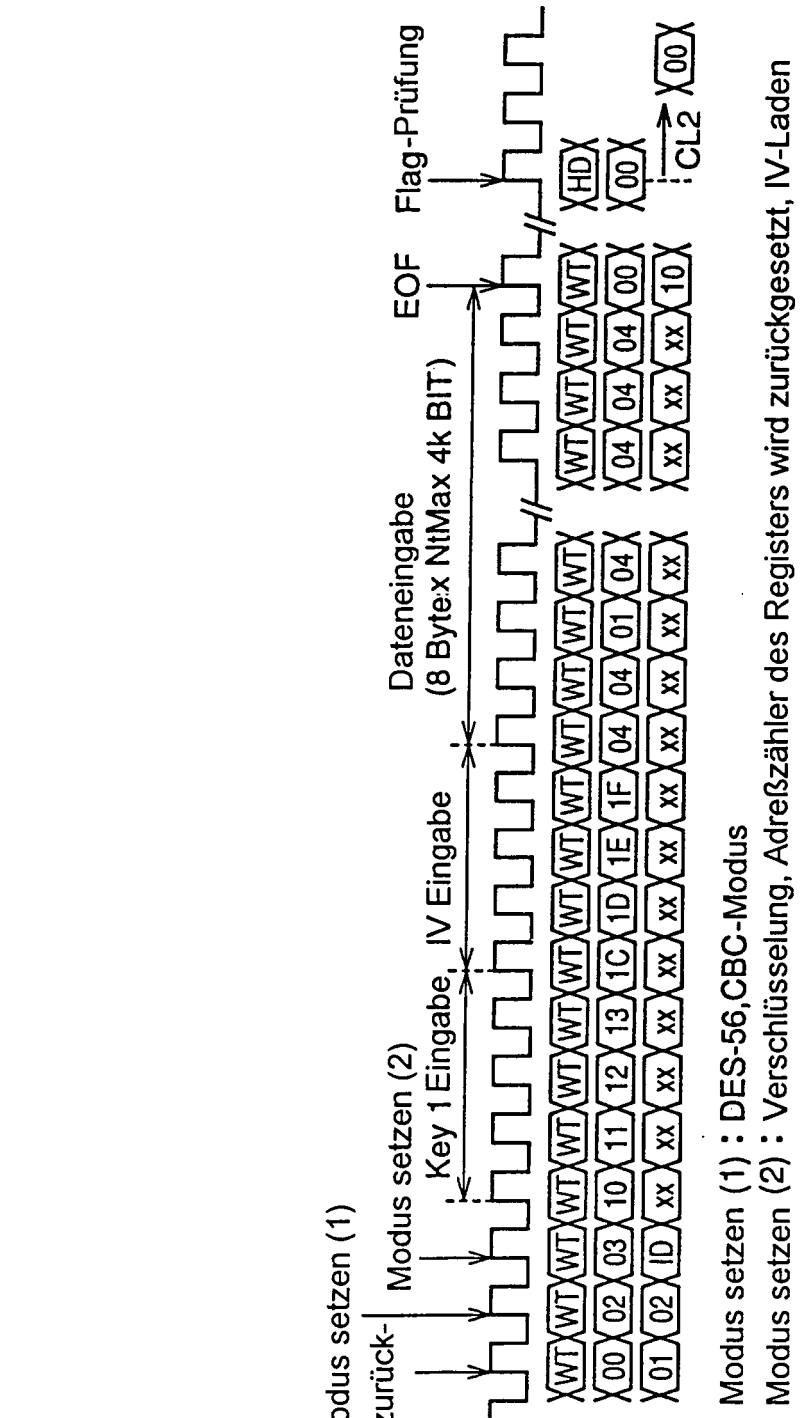

Modus setzen (1)
Software zurück-setzen
Modus setzen (2)
Key 1 Eingabe, IV Eingabe
Dateneingabe
(8 Byte x NtMax 4k BIT)
EOF
Flag-Prüfung
CLK
CMD
ADD
DQ[15:0]
ACT
3FFF
WT
00
02
03
10
11
12
13
1C
1D
1E
1F
01
04
ID
xx
HD
CL2
Modus setzen (1) : DES-56,CBC-Modus
Modus setzen (2) : Verschlüsselung, Adreßzähler des Registers wird zurückgesetzt, IV-Laden

FIG.21

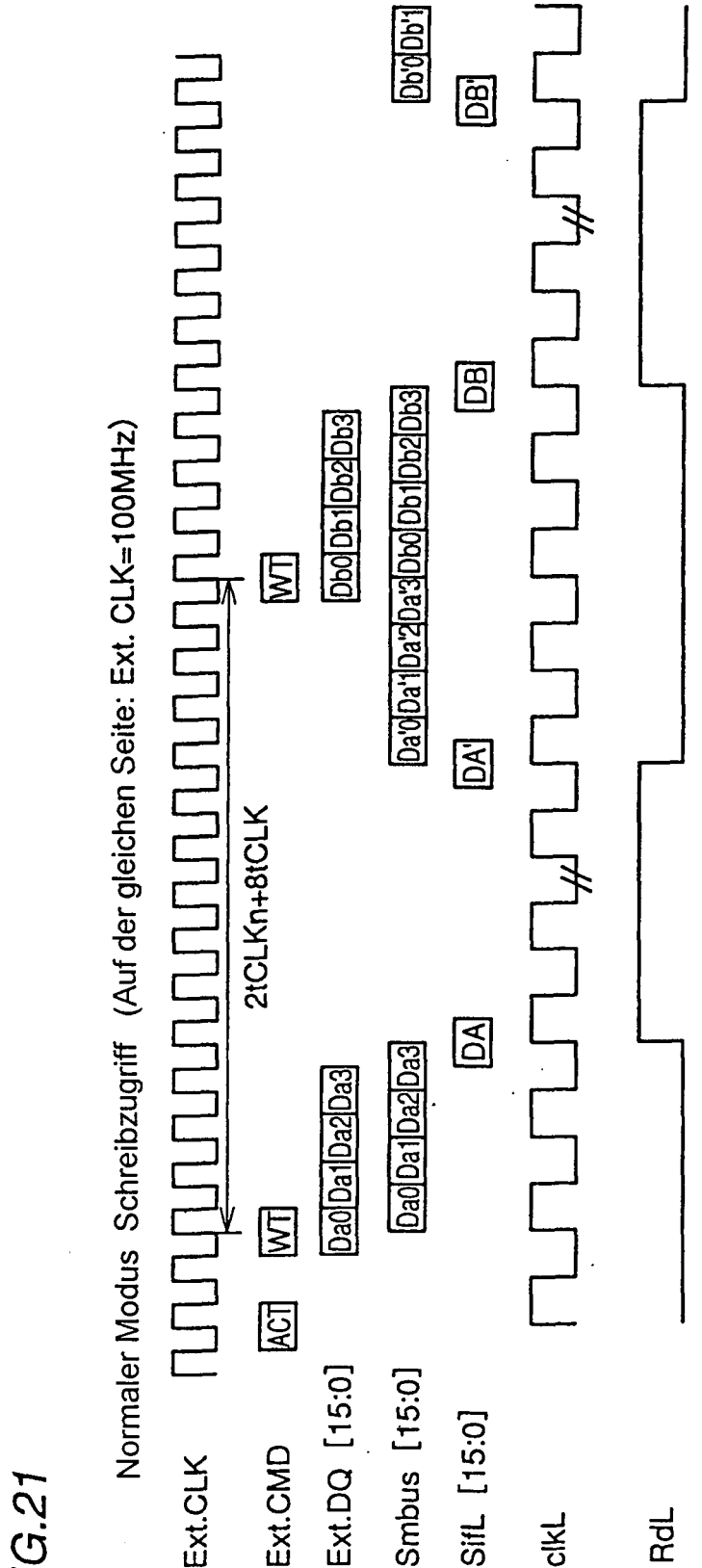

Normaler Modus Schreibzugriff (Auf der gleichen Seite: Ext. CLK=100MHz)
Ext.CLK
Ext.CMD
Ext.DQ [15:0]
Smbus [15:0]
SifL [15:0]
clkL
RdL
ACT
WT
2tCLKn+8tCLK
Da0 Da1 Da2 Da3
DA
Db0 Db1 Db2 Db3
DB
Da'0 Da'1 Da'2 Da'3 Db0 Db1 Db2 Db3
DA'
Db'0 Db'1
DB'

FIG.22

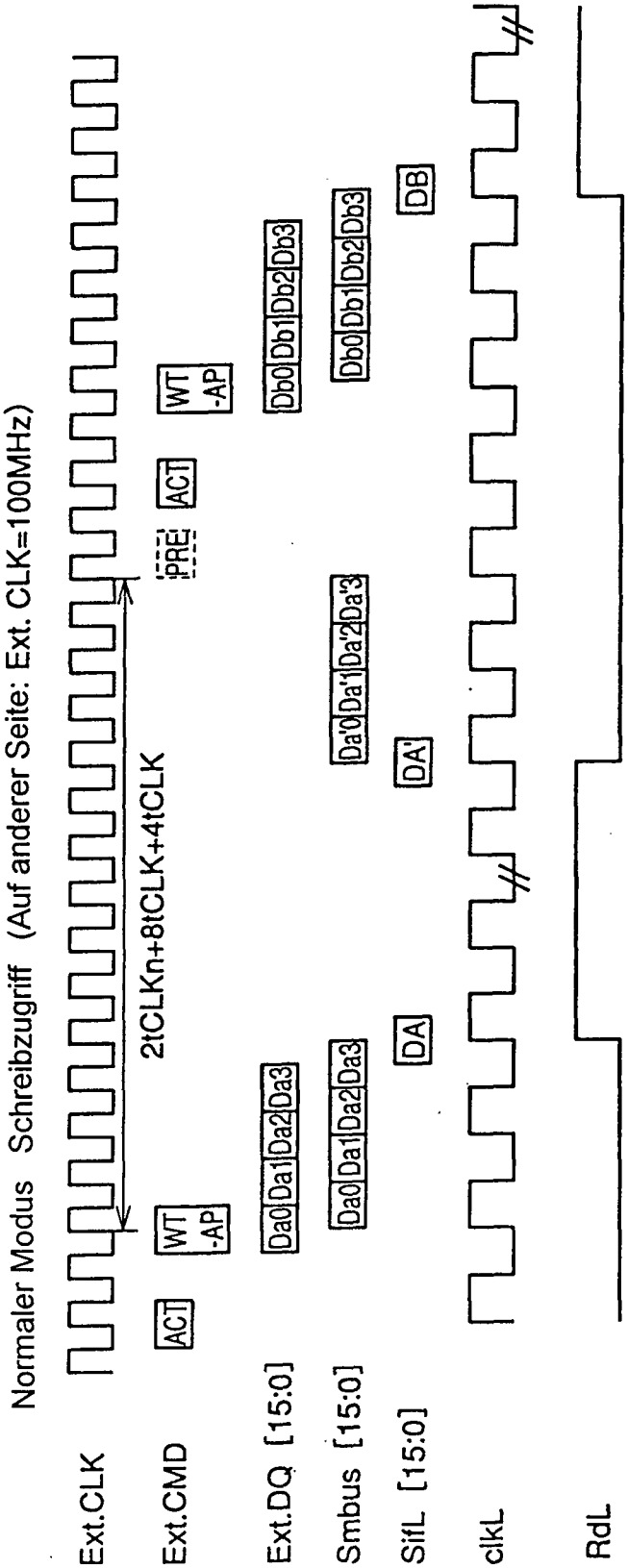

Normaler Modus Schreibzugriff (Auf anderer Seite: Ext. CLK=100MHz)
Ext.CLK
Ext.CMD
Ext.DQ [15:0]
Smbus [15:0]
SifL [15:0]
clkL
RdL
2tCLKn+8tCLK+4tCLK
ACT
WT -AP
PRE
ACT
WT -AP
Da0 Da1 Da2 Da3
Da0 Da1 Da2 Da3
DA
Da'0 Da'1 Da'2 Da'3
DA'
Db0 Db1 Db2 Db3
Db0 Db1 Db2 Db3
DB

FIG.23

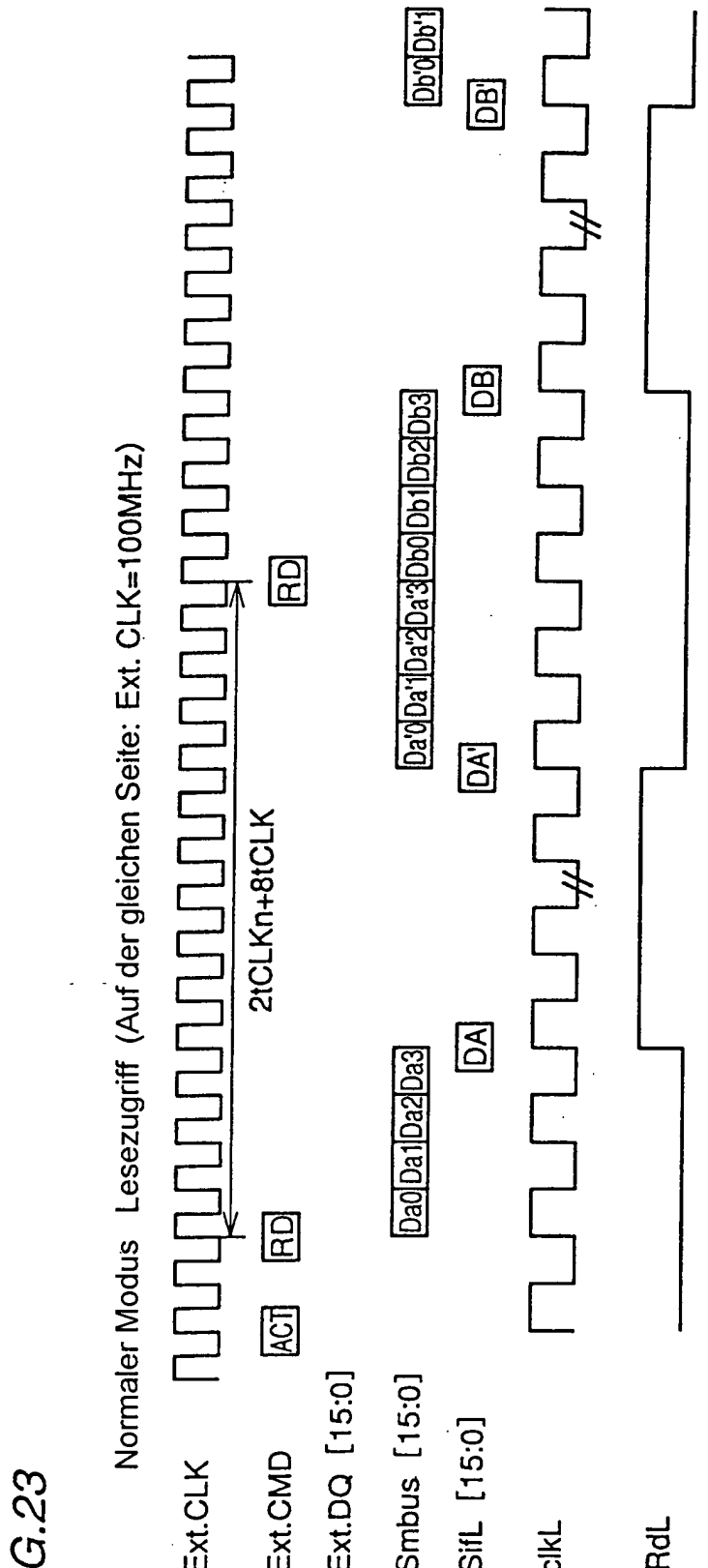

Normaler Modus Lesezugriff (Auf der gleichen Seite: Ext. CLK=100MHz)
Ext.CLK
Ext.CMD
ACT
RD
RD
2tCLKn+8tCLK
Ext.DQ [15:0]
Smbus [15:0]
Da0 Da1 Da2 Da3
Da'0 Da'1 Da'2 Da'3 Db0 Db1 Db2 Db3
Db'0 Db'1
SifL [15:0]
DA
DA'
DB
DB'
clkL
RdL

FIG.24

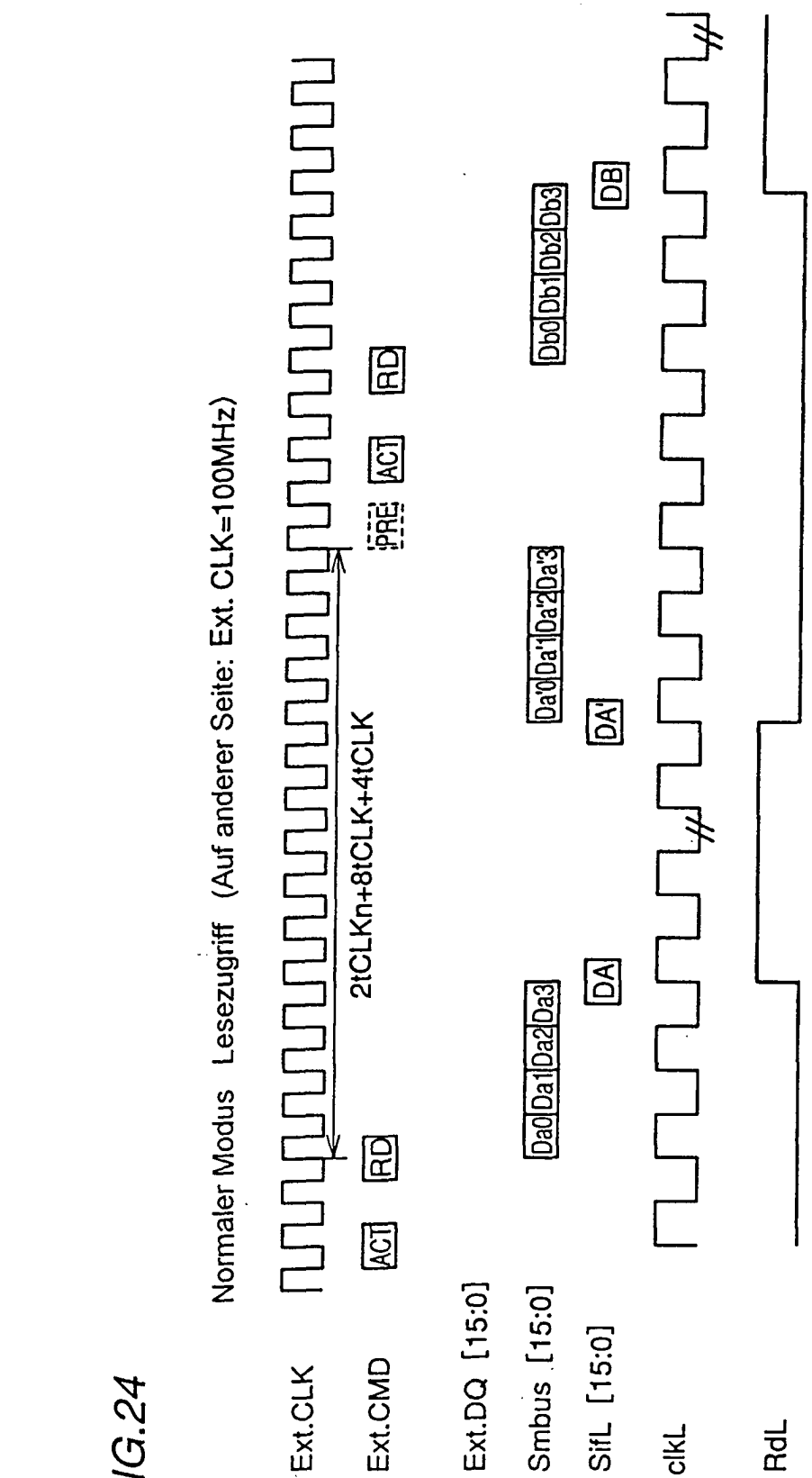

Normaler Modus Lesezugriff (Auf anderer Seite: Ext. CLK=100MHz)
Ext.CLK
Ext.CMD
Ext.DQ [15:0]
Smbus [15:0]
SifL [15:0]
clkL
RdL
ACT
RD
PRE
ACT
RD
2tCLKn+8tCLK+4tCLK
Da0 Da1 Da2 Da3
DA
Da'0 Da'1 Da'2 Da'3
DA'
Db0 Db1 Db2 Db3
DB

FIG.25

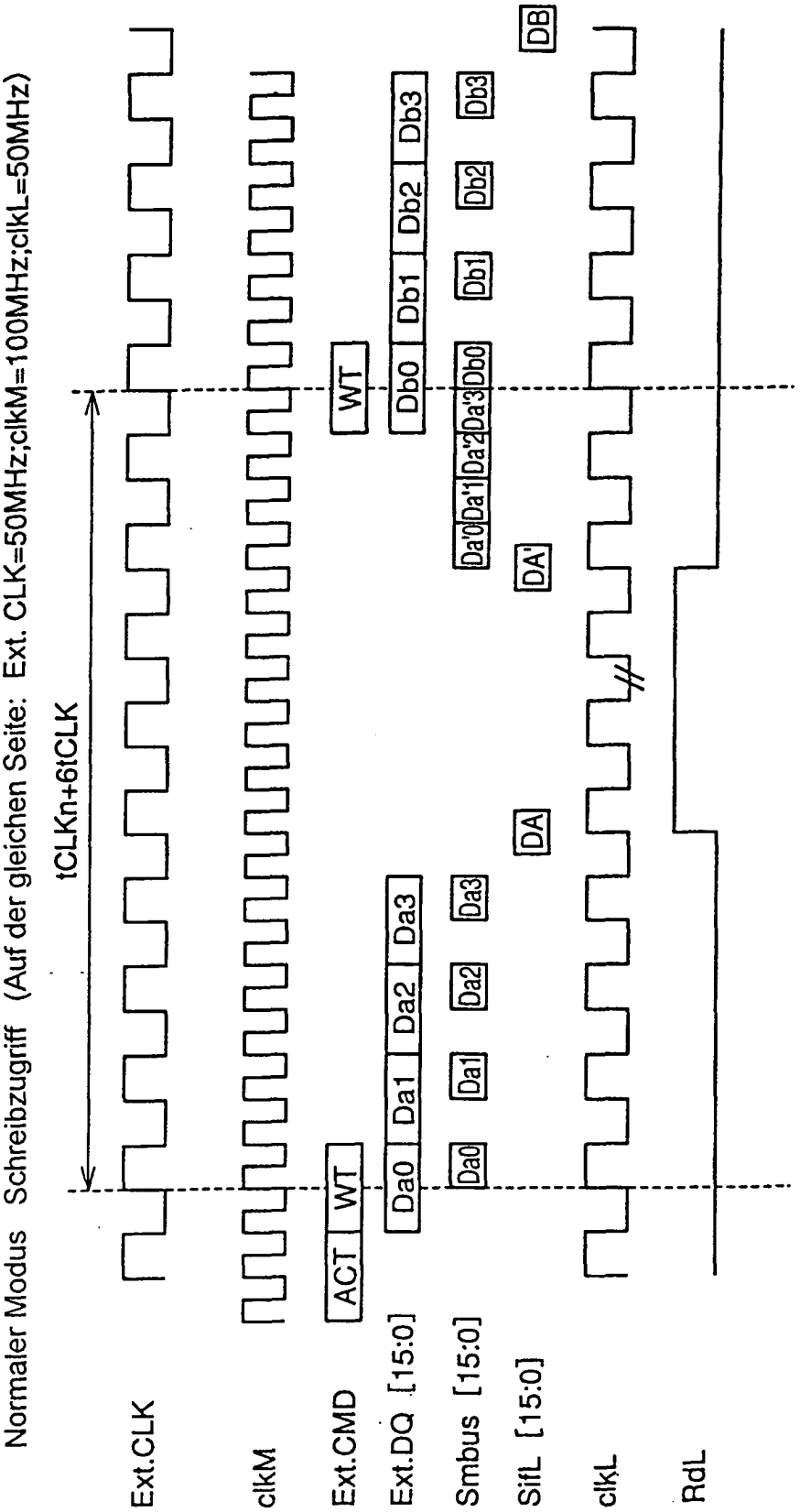

Normaler Modus Schreibzugriff (Auf der gleichen Seite: Ext. CLK=50MHz;clkM=100MHz;clkL=50MHz)
tCLKn+6tCLK
Ext.CLK
clkM
Ext.CMD
Ext.DQ [15:0]
Smbus [15:0]
SifL [15:0]
clkL
RdL
ACT
WT
Da0
Da1
Da2
Da3
DA
WT
Db0
Db1
Db2
Db3
Da'0
Da'1
Da'2
Da'3
DA'
DB

FIG.26

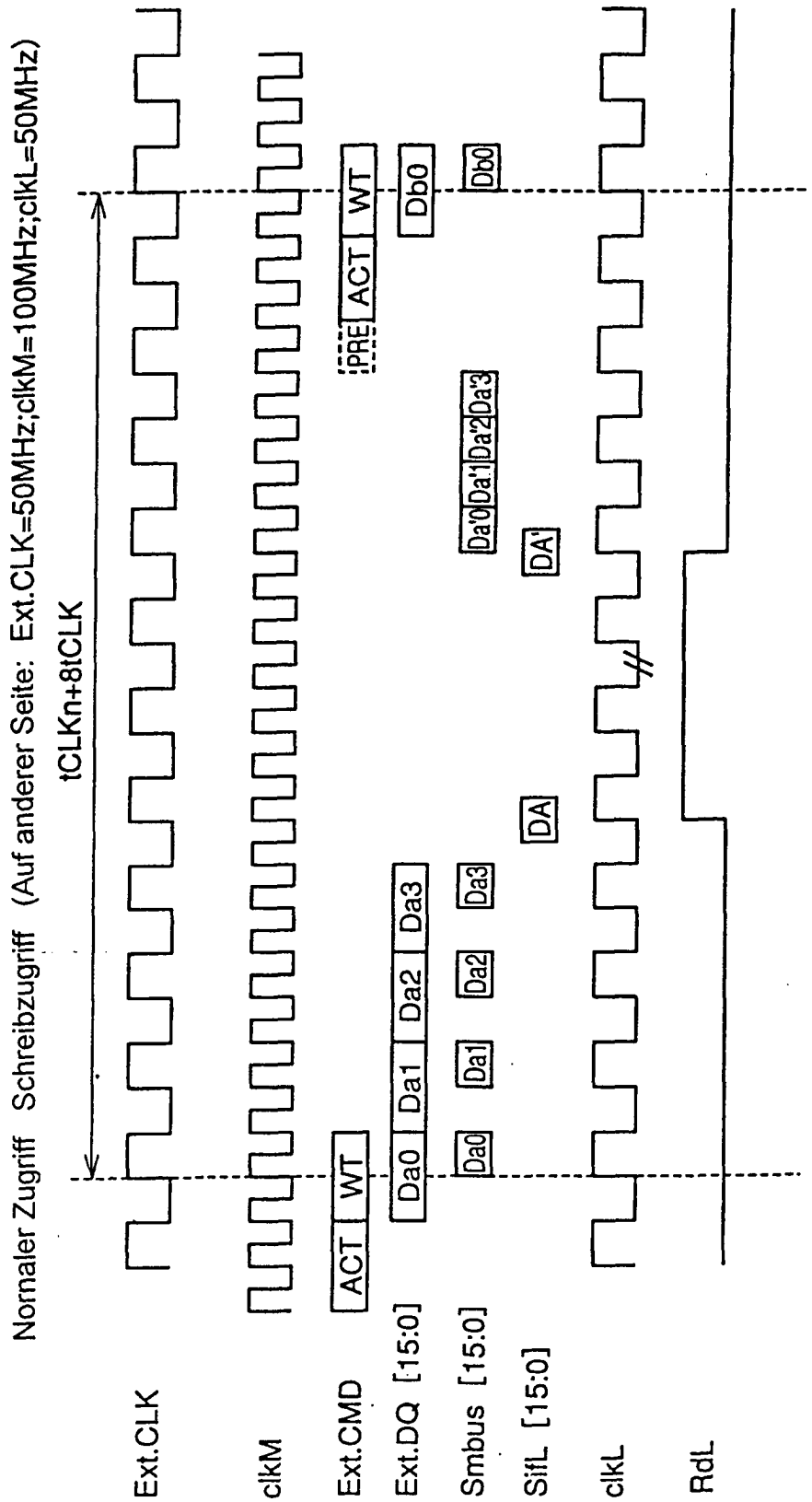

Nornaler Zugriff Schreibzugriff (Auf anderer Seite: Ext.CLK=50MHz;clkM=100MHz;clkL=50MHz)
tCLKn+8tCLK
Ext.CLK
clkM
Ext.CMD
Ext.DQ [15:0]
Smbus [15:0]
SifL [15:0]
clkL
RdL
PRE
ACT
WT
Da0
Da1
Da2
Da3
Db0
Da'0
Da'1
Da'2
Da'3
DA
DA'

FIG.27

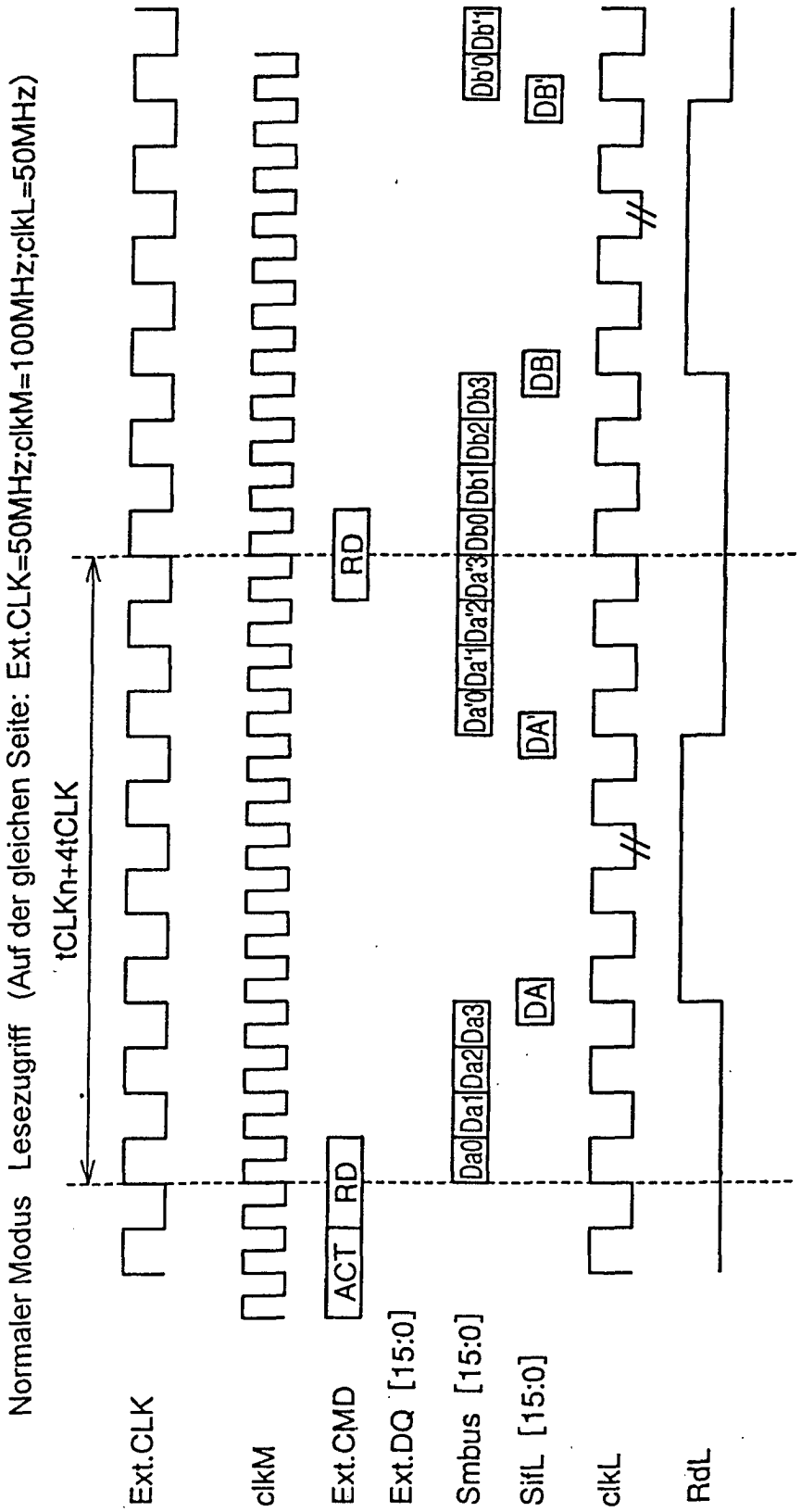

Normaler Modus Lesezugriff (Auf der gleichen Seite: Ext.CLK=50MHz;clkM=100MHz;clkL=50MHz)
tCLKn+4tCLK
Ext.CLK
clkM
Ext.CMD
ACT
RD
RD
Ext.DQ [15:0]
Smbus [15:0]
Da0 Da1 Da2 Da3
Da'0 Da'1 Da'2 Da'3 Db0 Db1 Db2 Db3
Db'0 Db'1
SifL [15:0]
DA
DA'
DB
DB'
clkL
RdL

FIG.28

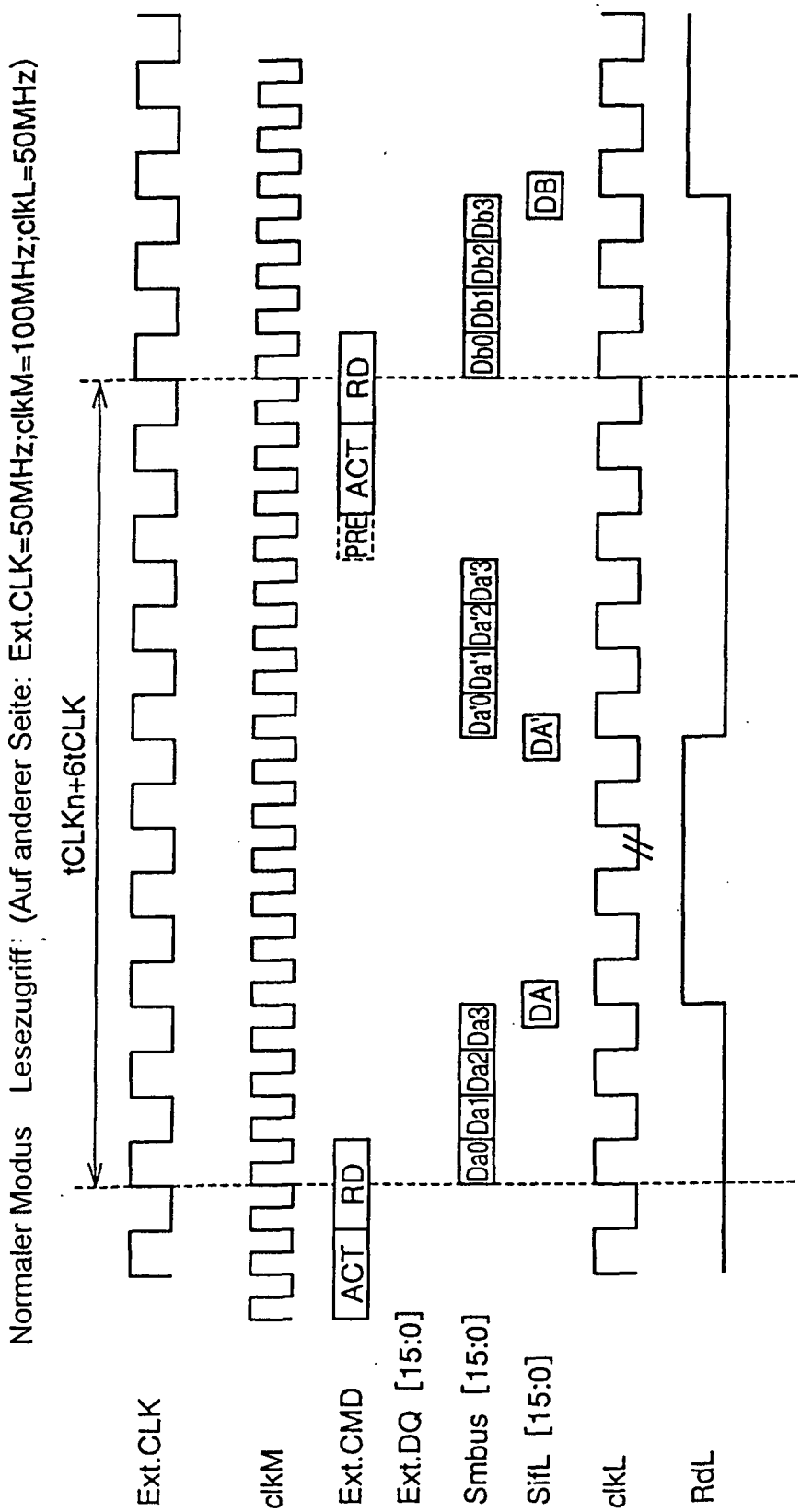

Normaler Modus Lesezugriff (Auf anderer Seite: Ext.CLK=50MHz;clkM=100MHz;clkL=50MHz)
tCLKn+6tCLK
Ext.CLK
clkM
Ext.CMD
Ext.DQ [15:0]
Smbus [15:0]
SifL [15:0]
clkL
RdL
ACT
RD
PRE
ACT
RD
Da0 Da1 Da2 Da3
DA
Da'0 Da'1 Da'2 Da'3
DA'
Db0 Db1 Db2 Db3
DB

FIG.29

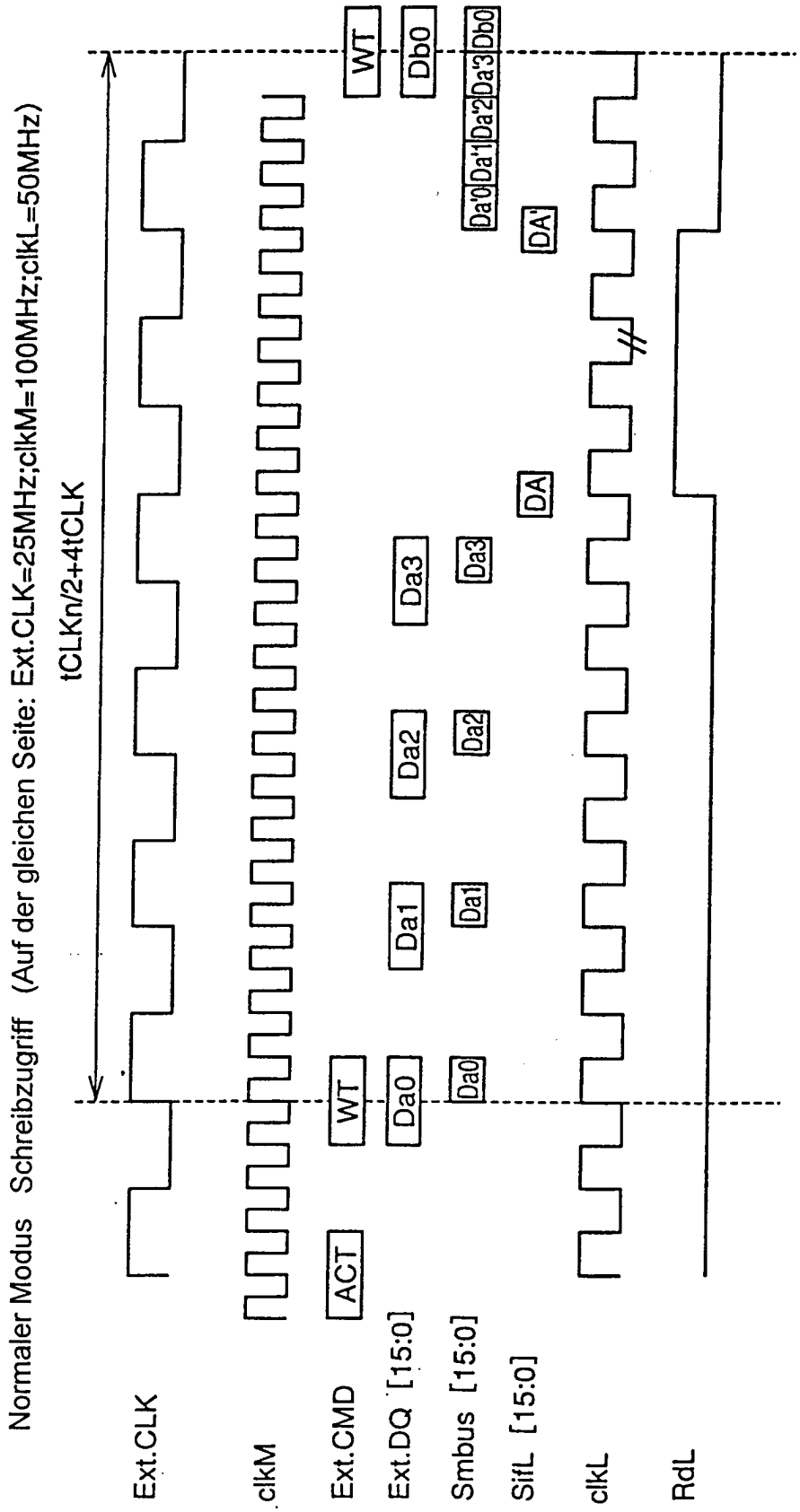

Normaler Modus Schreibzugriff (Auf der gleichen Seite: Ext.CLK=25MHz;clkM=100MHz;clkL=50MHz)
tCLKn/2+4tCLK
Ext.CLK
clkM
Ext.CMD
Ext.DQ [15:0]
Smbus [15:0]
SifL [15:0]
clkL
RdL
ACT
WT
Da0
Da1
Da2
Da3
Db0
DA
DA'
Da'0
Da'1
Da'2
Da'3

FIG.30

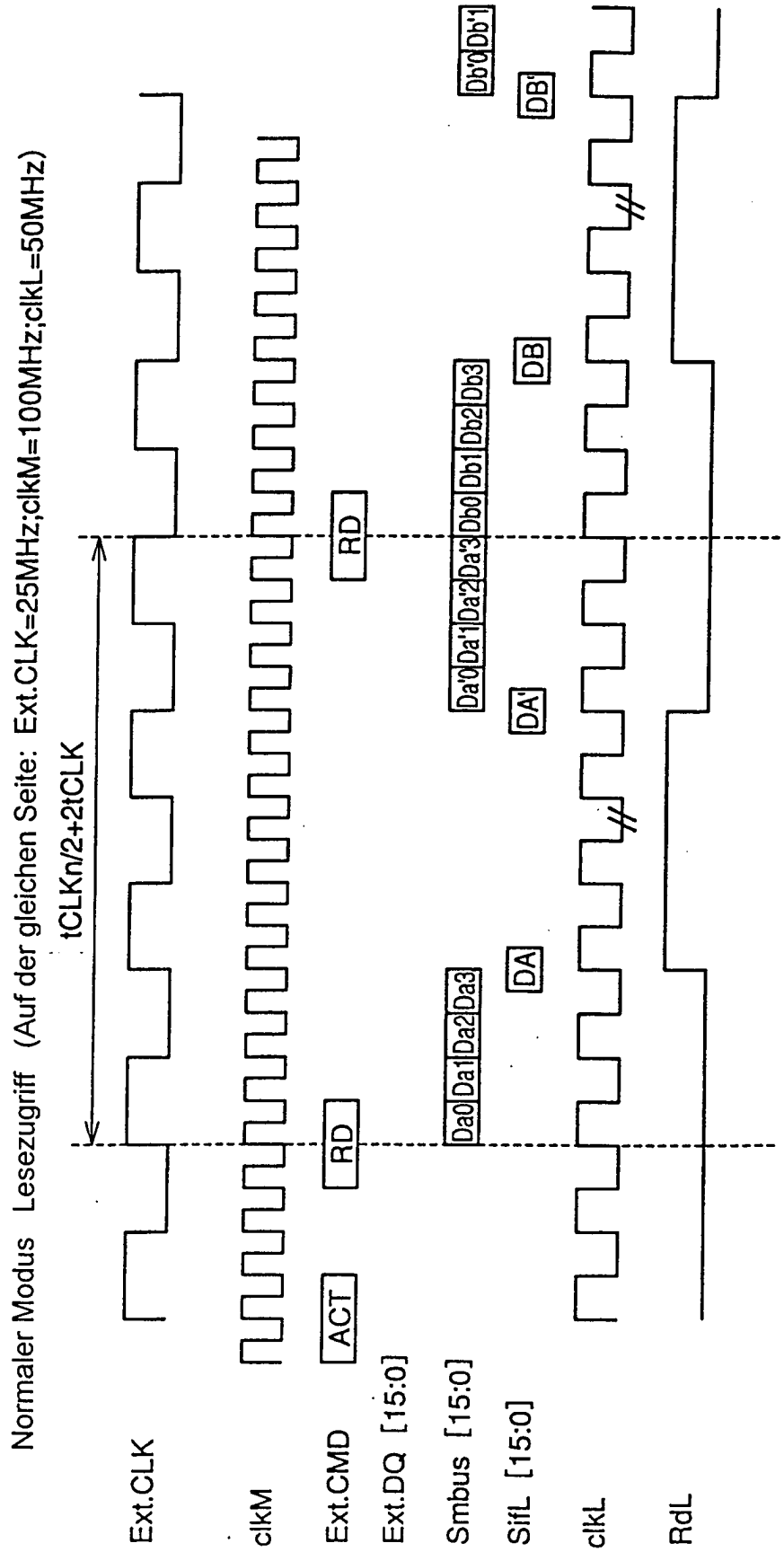

Normaler Modus Lesezugriff (Auf der gleichen Seite: Ext.CLK=25MHz;clkM=100MHz;clkL=50MHz)
tCLKn/2+2tCLK
Ext.CLK
clkM
Ext.CMD
ACT
RD
RD
Ext.DQ [15:0]
Da0 Da1 Da2 Da3
DA
Smbus [15:0]
Da'0 Da'1 Da'2 Da'3 Db0 Db1 Db2 Db3
DA'
DB
Db'0 Db'1
DB'
SifL [15:0]
clkL
RdL

FIG.31

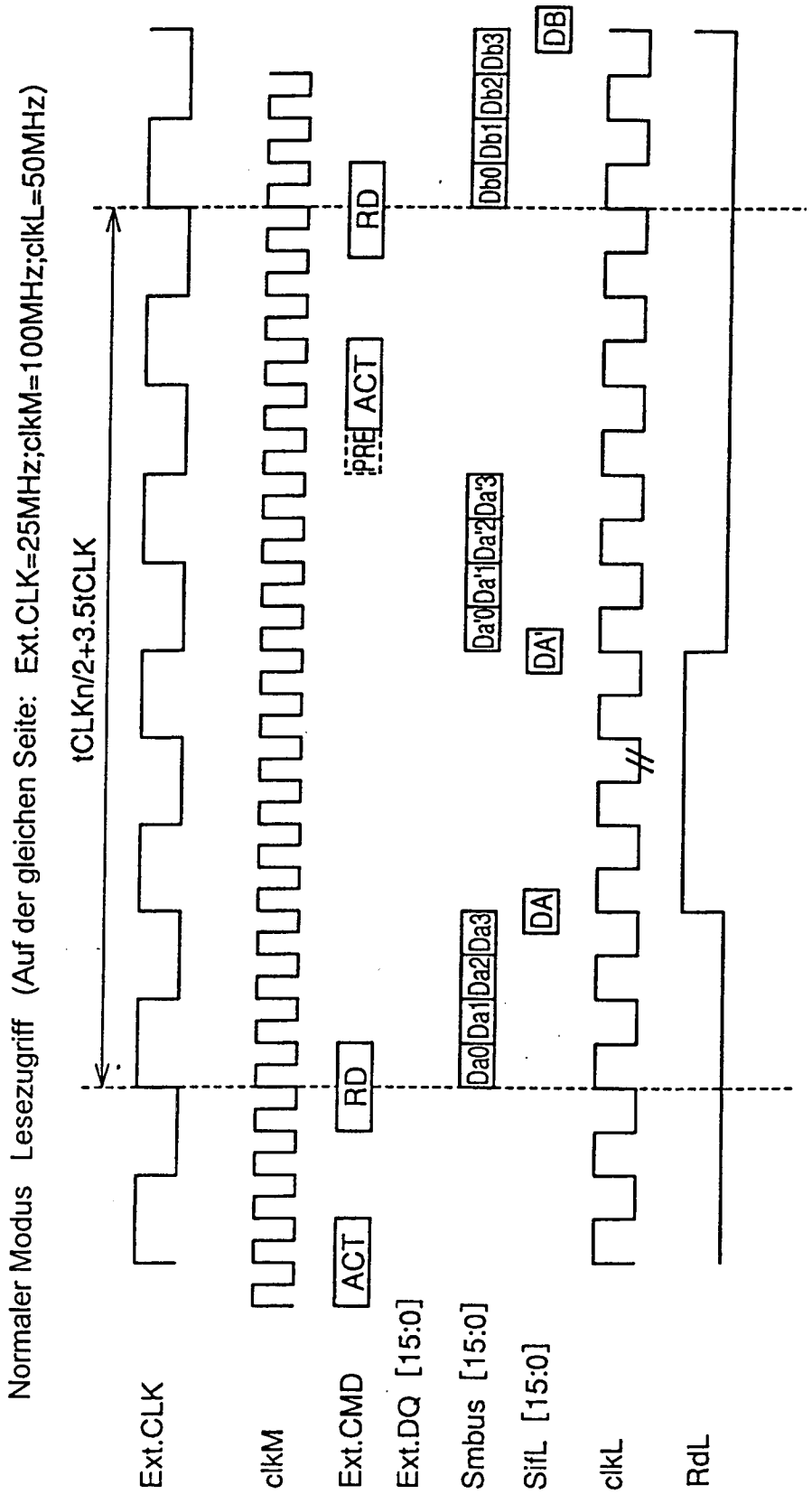

Normaler Modus Lesezugriff (Auf der gleichen Seite: Ext.CLK=25MHz;clkM=100MHz;clkL=50MHz)
tCLKn/2+3.5tCLK
Ext.CLK
clkM
Ext.CMD
Ext.DQ [15:0]
Smbus [15:0]
SifL [15:0]
clkL
RdL
ACT
RD
PRE
ACT
RD
Da0 Da1 Da2 Da3
DA
Da'0 Da'1 Da'2 Da'3
DA'
Db0 Db1 Db2 Db3
DB

FIG.32

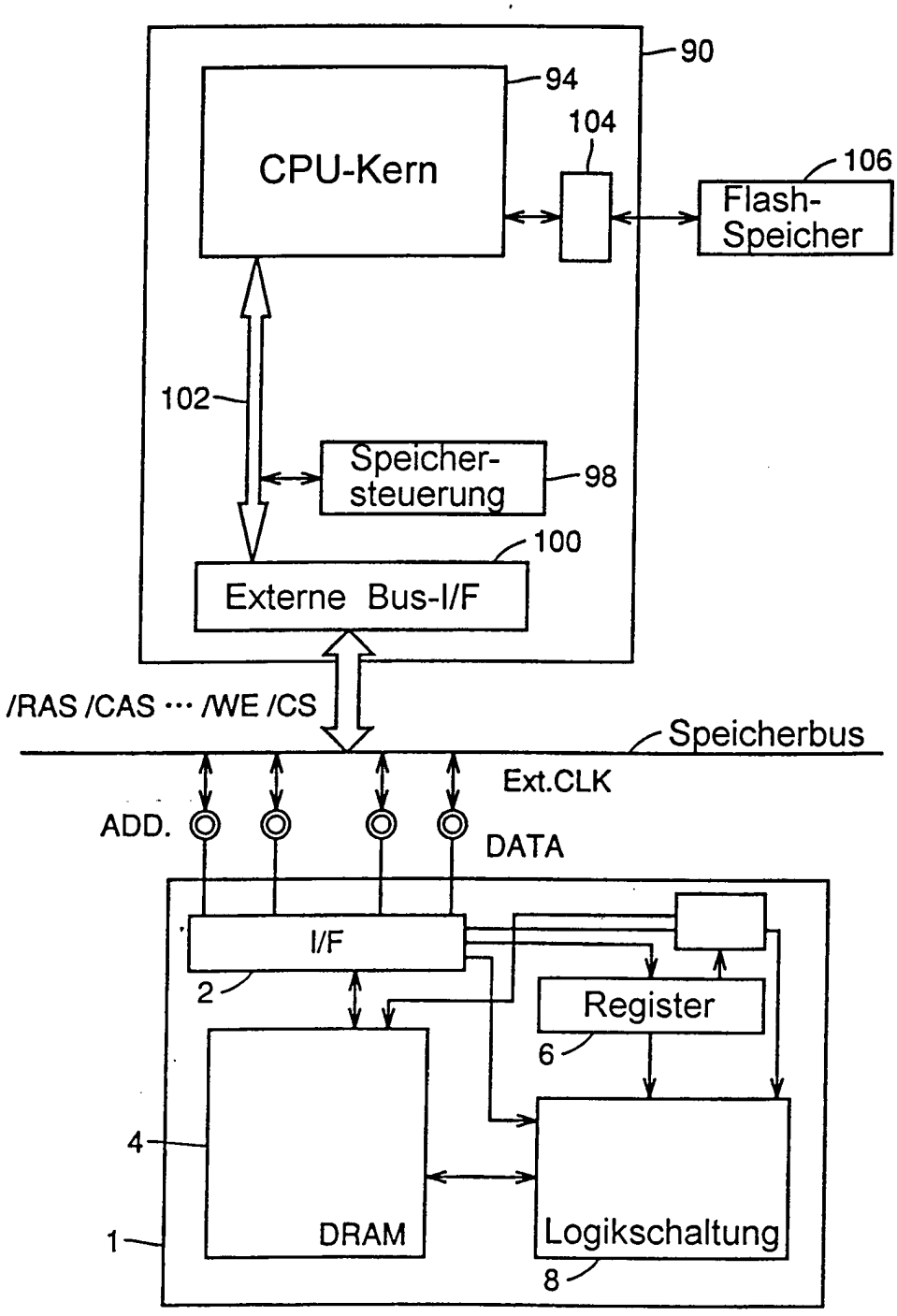

90
94
CPU-Kern
104
106
Flash-
Speicher
102
Speicher-
steuerung
98
100
Externe Bus-I/F
/RAS /CAS ··· /WE /CS
Speicherbus
Ext.CLK
ADD.
DATA
I/F
2
Register
6
4
DRAM
Logikschaltung
1
8

FIG.33

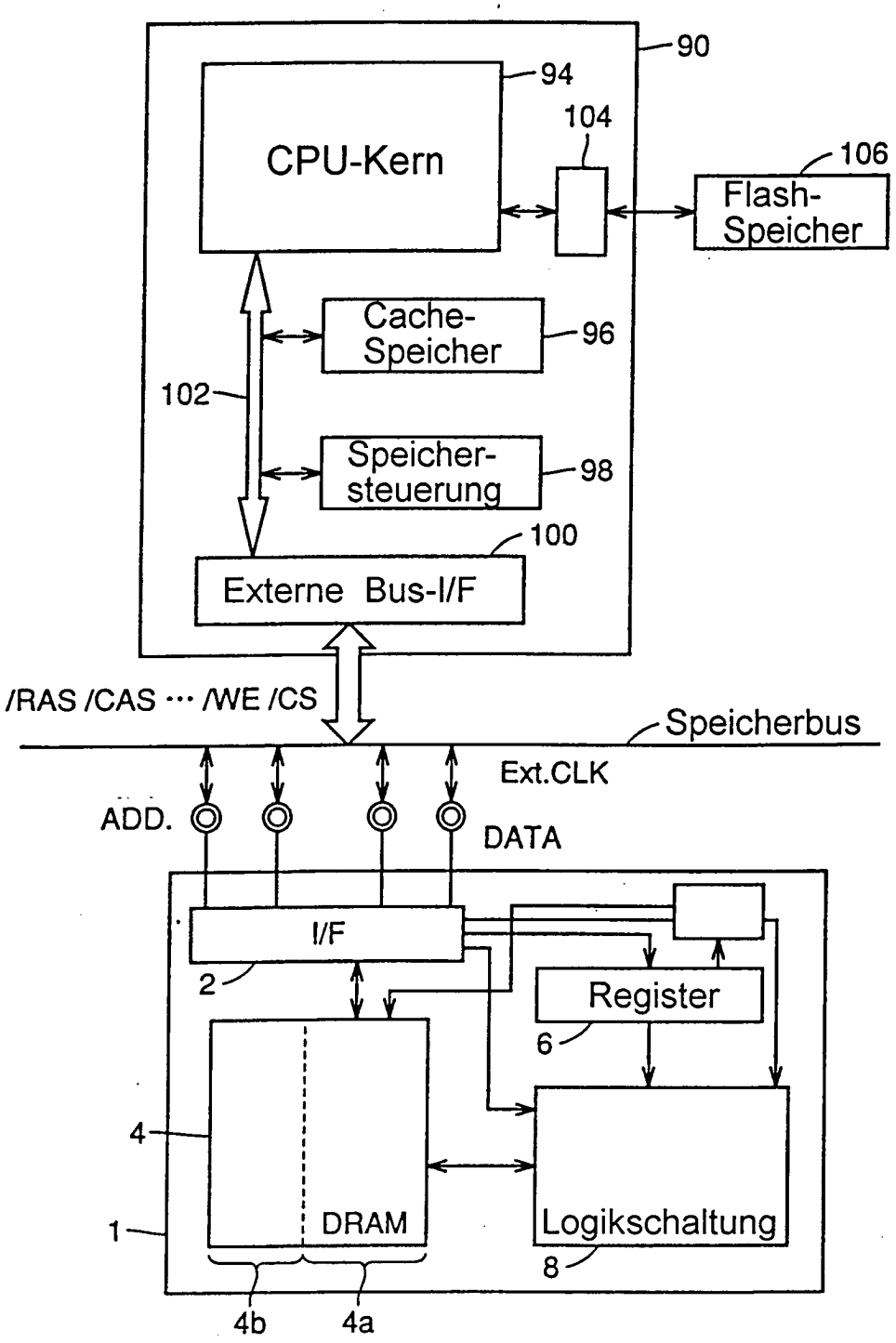

90
94
CPU-Kern
104
106
Flash-
Speicher
Cache-
Speicher
96
102
Speicher-
steuerung
98
100
Externe Bus-I/F
/RAS /CAS ··· /WE /CS
Speicherbus
Ext.CLK
ADD.
DATA
I/F
2
Register
6
4
DRAM
Logikschaltung
1
8
4b
4a

FIG.34

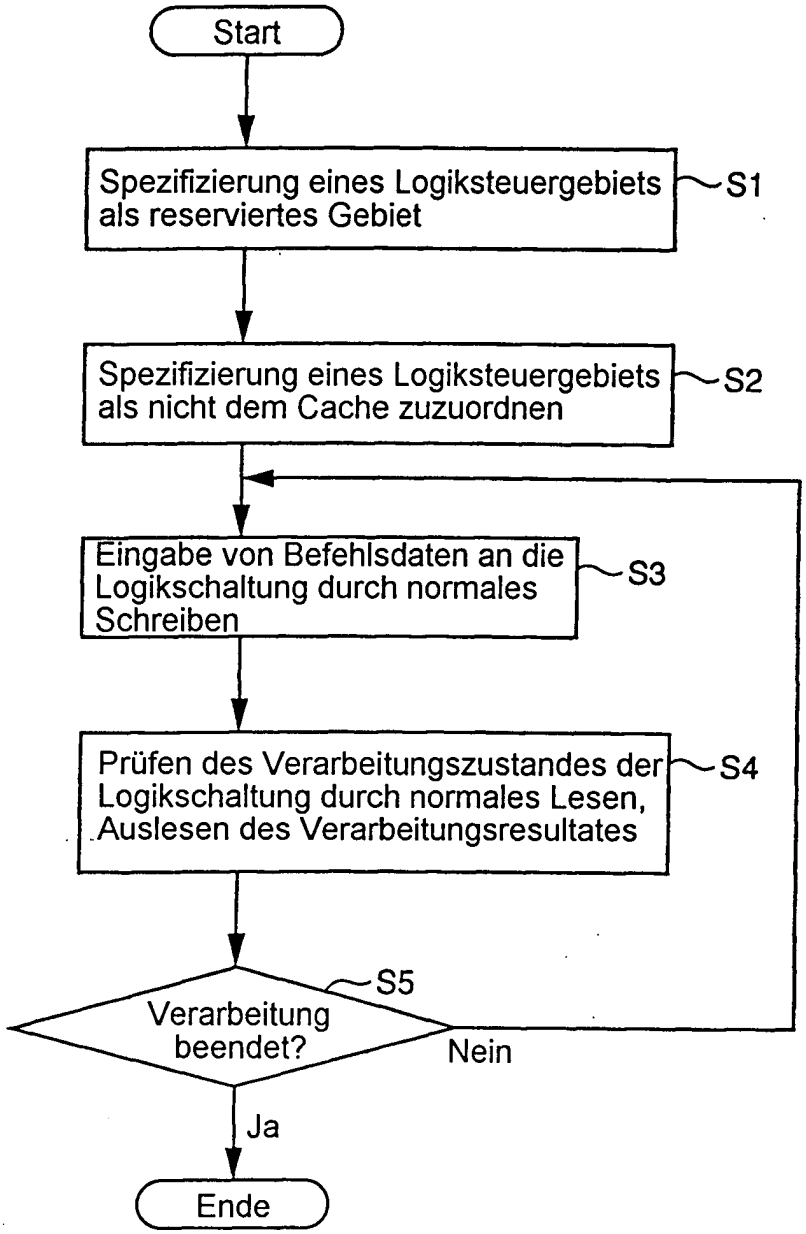

Start
Spezifizierung eines Logiksteuergebiets als reserviertes Gebiet
S1
Spezifizierung eines Logiksteuergebiets als nicht dem Cache zuzuordnen
S2
Eingabe von Befehlsdaten an die Logikschaltung durch normales Schreiben
S3
Prüfen des Verarbeitungszustandes der Logikschaltung durch normales Lesen, Auslesen des Verarbeitungsresultates
S4
S5
Verarbeitung beendet?
Nein
Ja
Ende

FIG.35

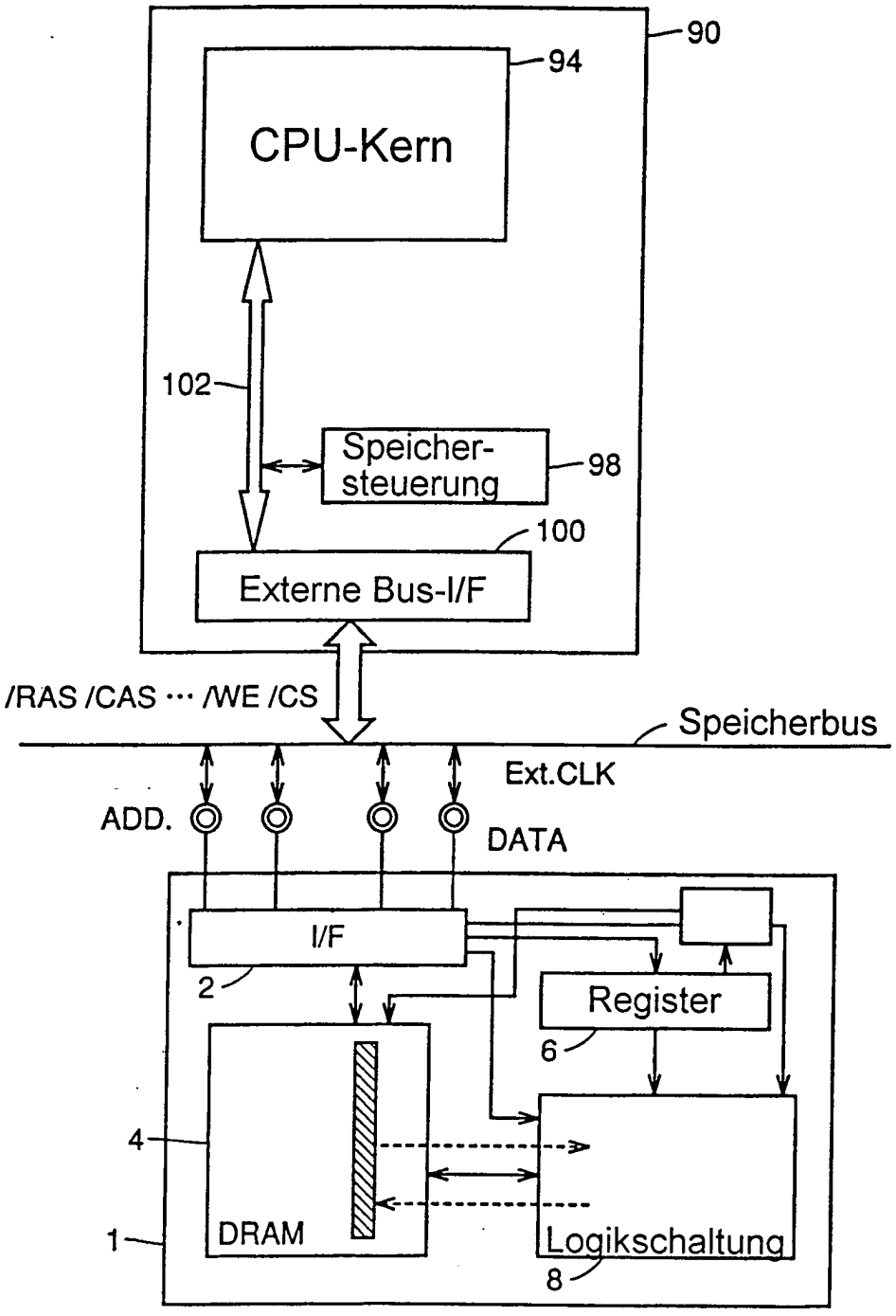

90
94
CPU-Kern
102
Speicher-
steuerung
98
100
Externe Bus-I/F
/RAS /CAS ··· /WE /CS
Speicherbus
Ext.CLK
ADD.
DATA
I/F
2
Register
6
4
DRAM
1
Logikschaltung
8

FIG.36

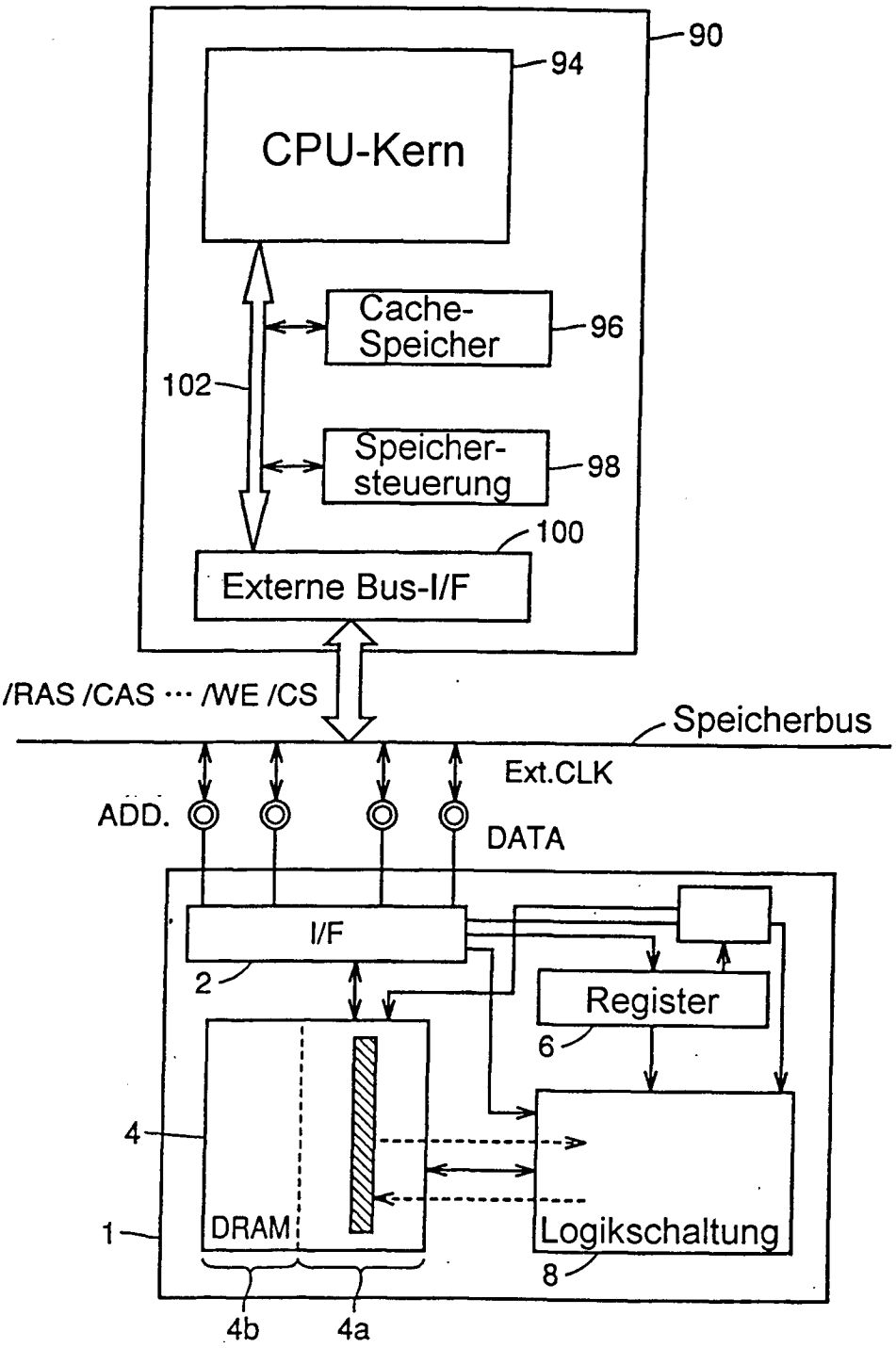

90
94
CPU-Kern
Cache-
Speicher
96
102
Speicher-
steuerung
98
100
Externe Bus-I/F
/RAS /CAS ··· /WE /CS
Speicherbus
Ext.CLK
ADD.
DATA
I/F
2
Register
6
4
DRAM
1
Logikschaltung
8
4b
4a

FIG.37

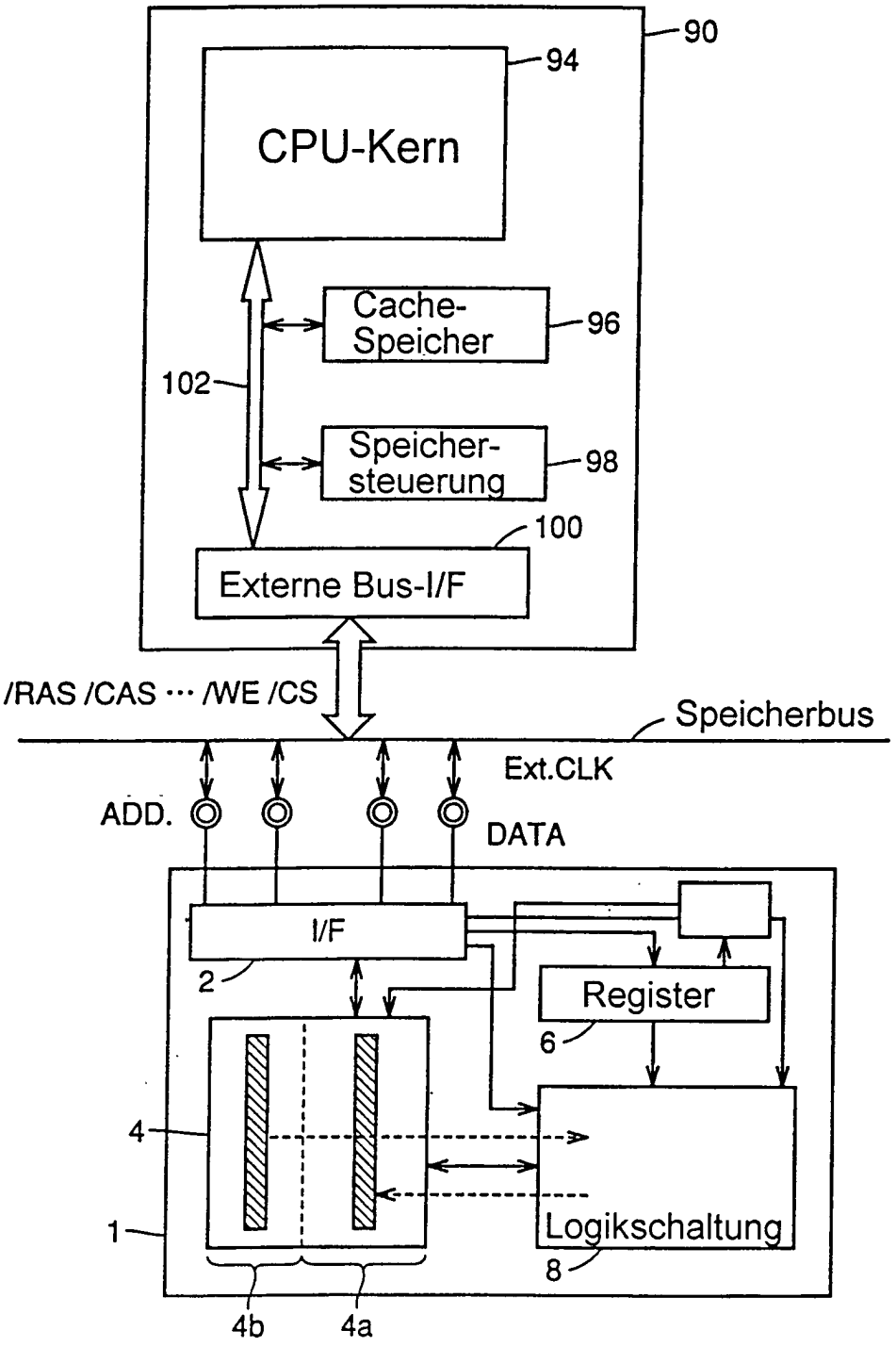

90
94
CPU-Kern
Cache-
Speicher
96
102
Speicher-
steuerung
98
100
Externe Bus-I/F
/RAS /CAS ··· /WE /CS
Speicherbus
Ext.CLK
ADD.
DATA
I/F
2
Register
6
4
1
Logikschaltung
8
4b
4a

FIG.38

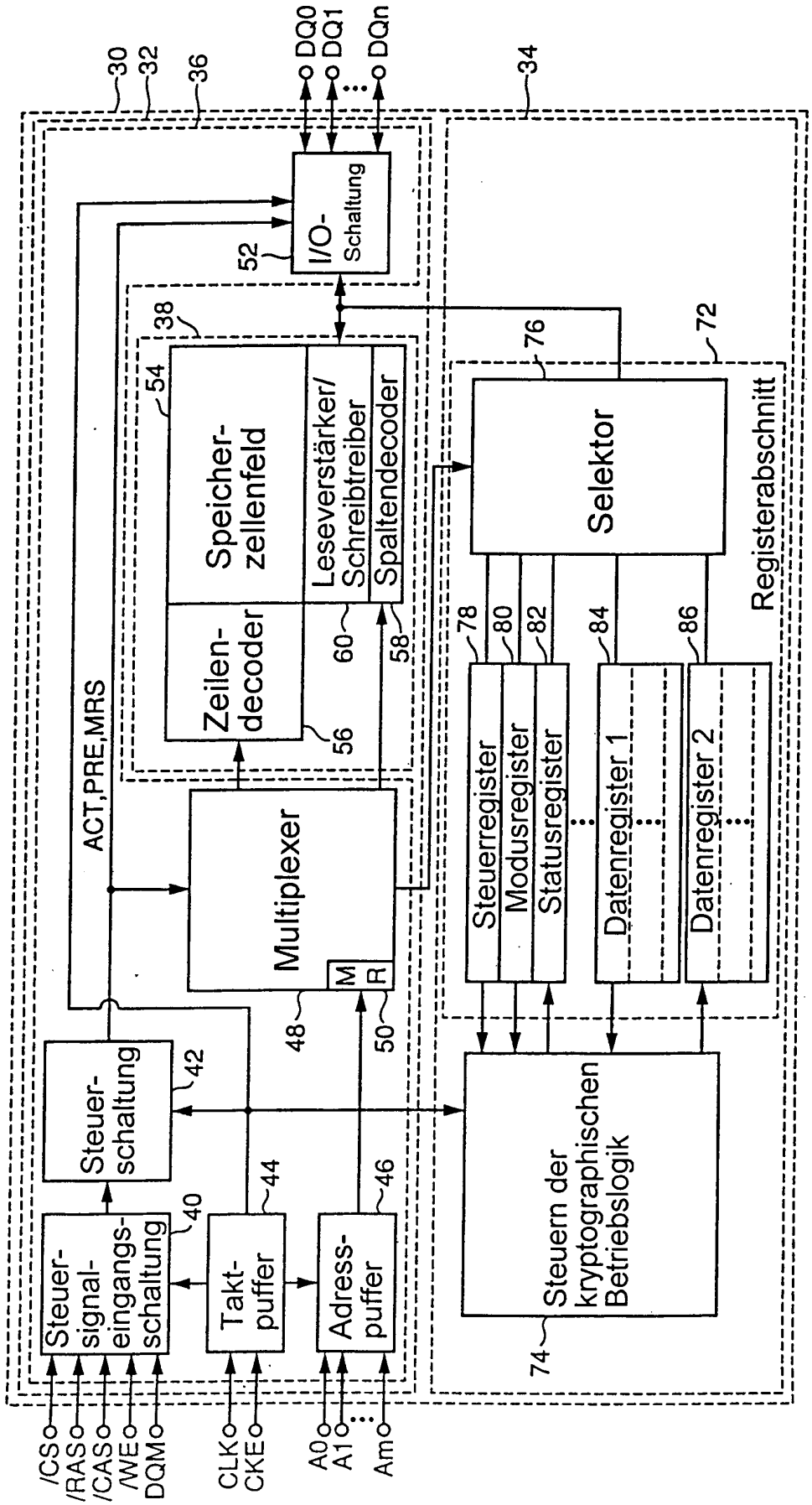

30
32
36
34
DQ0
DQ1
DQn
I/O-Schaltung
52
38
54
Speicher-zellenfeld
Leseverstärker/ Schreibtreiber
Spaltendecoder
Zeilen-decoder
56
60
58
ACT,PRE,MRS
Multiplexer
M
R
48
50
Steuer-schaltung
42
Steuer-signal-eingangs-schaltung
40
Takt-puffer
44
Adress-puffer
46
/CS
/RAS
/CAS
/WE
DQM
CLK
CKE
A0
A1
Am
72
76
Selektor
Registerabschnitt
78
80
82
84
86
Steuerregister
Modusregister
Statusregister
Datenregister 1
Datenregister 2
Steuern der kryptographischen Betriebslogik
74

FIG.39

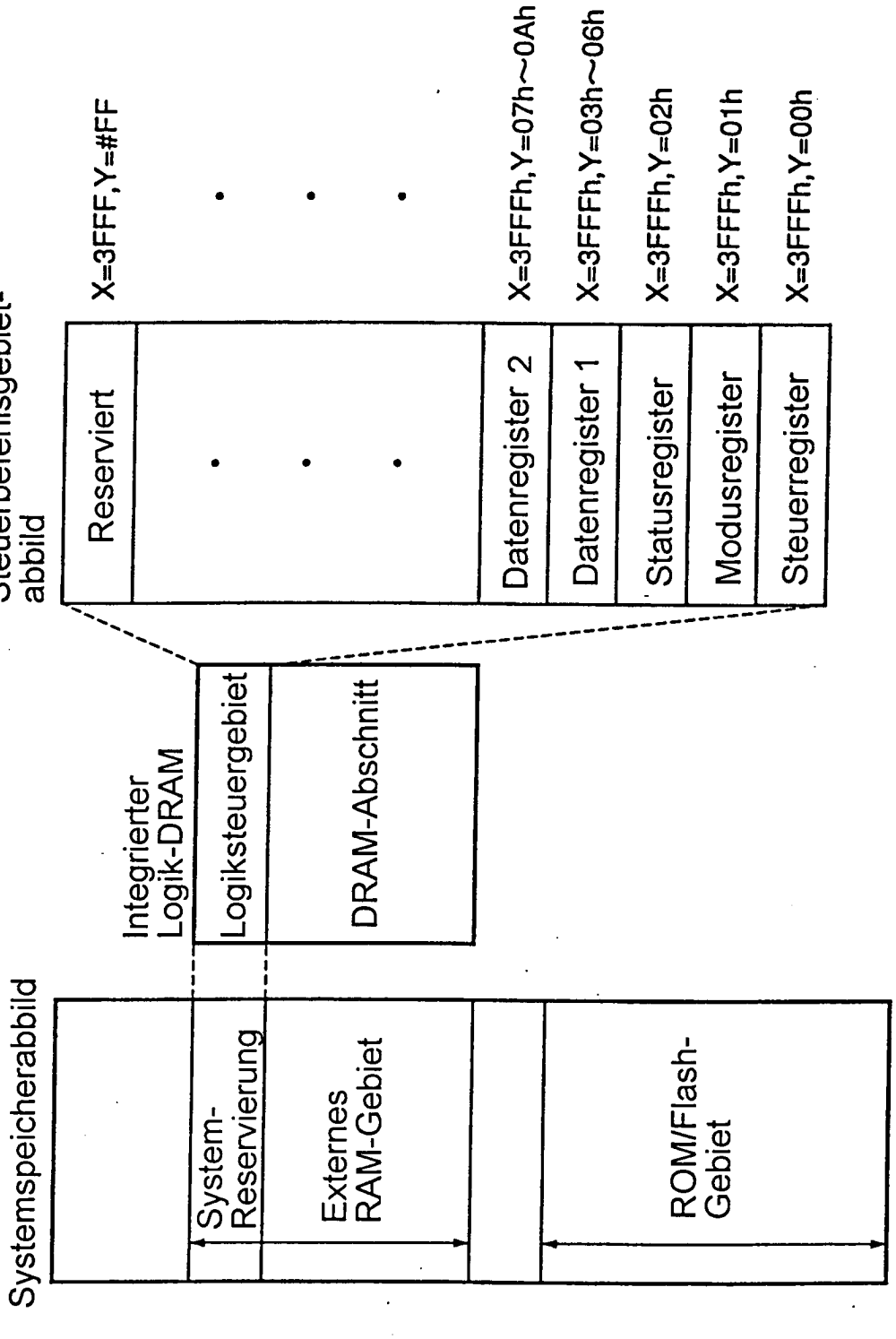

Systemspeicherabbild
System-Reservierung
Externes RAM-Gebiet
ROM/Flash-Gebiet
Integrierter Logik-DRAM
Logiksteuergebiet
DRAM-Abschnitt
Steuerbefehlsgebiet-abbild
Reserviert
Datenregister 2
Datenregister 1
Statusregister
Modusregister
Steuerregister
X=3FFF,Y=#FF
X=3FFFh,Y=07h~0Ah
X=3FFFh,Y=03h~06h
X=3FFFh,Y=02h
X=3FFFh,Y=01h
X=3FFFh,Y=00h

FIG.40

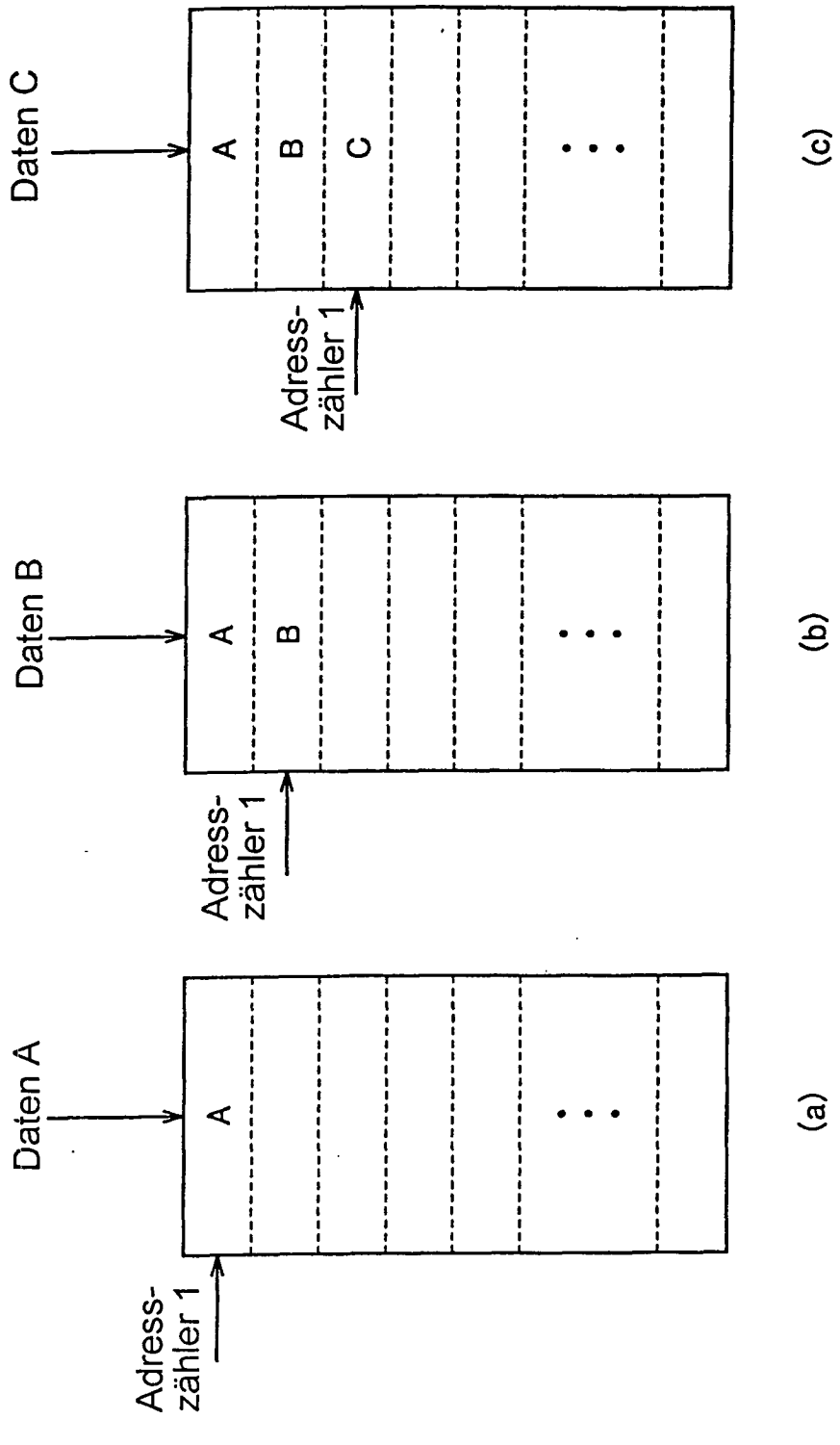

Daten A
A
Adress-
zähler 1
(a)
Daten B
A
B
Adress-
zähler 1
(b)
Daten C
A
B
C
Adress-
zähler 1
(c)

FIG.41

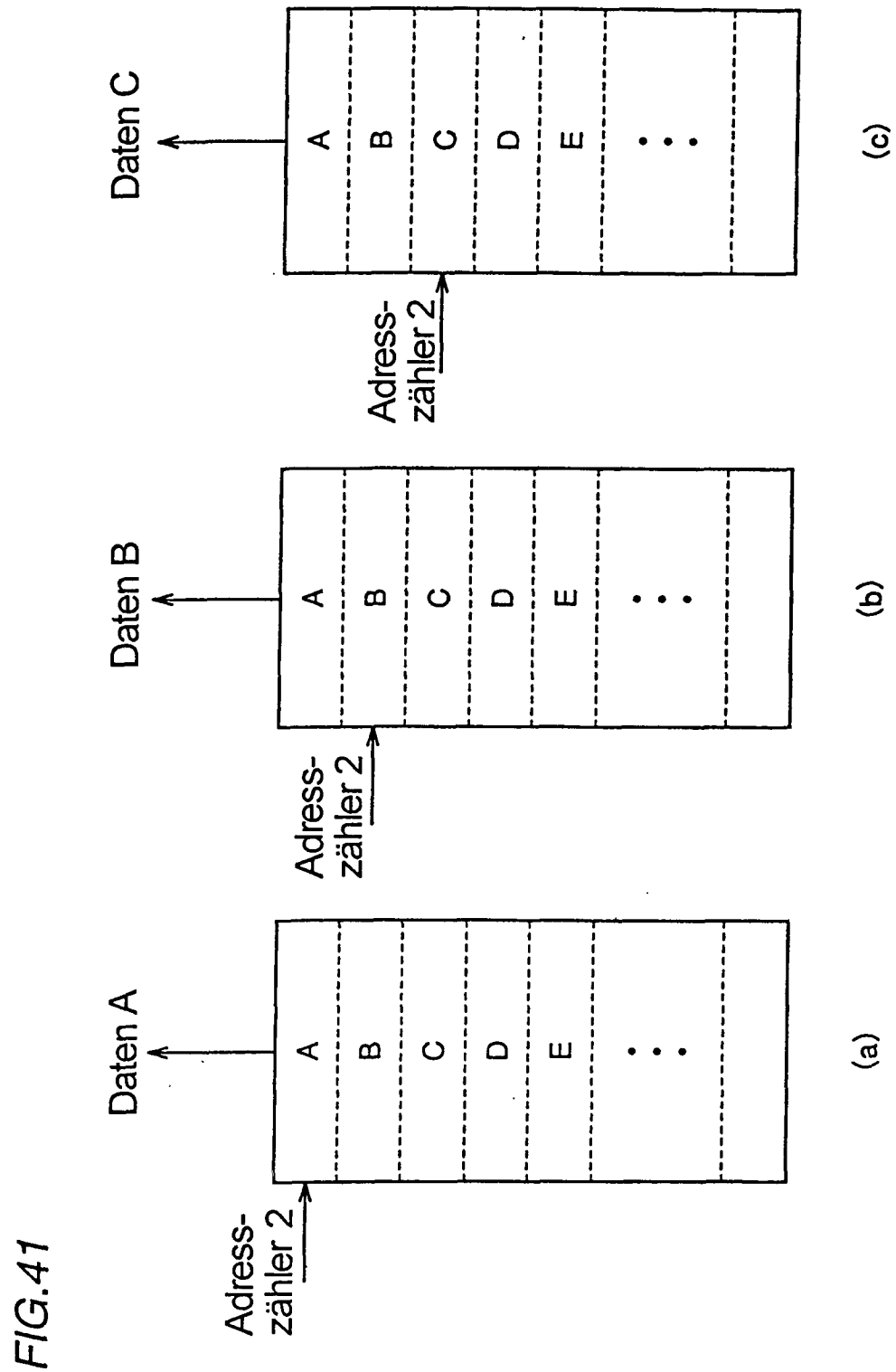

Daten A
Adress-
zähler 2
A
B
C
D
E
(a)
Daten B
Adress-
zähler 2
(b)
Daten C
Adress-
zähler 2
(c)

FIG.42

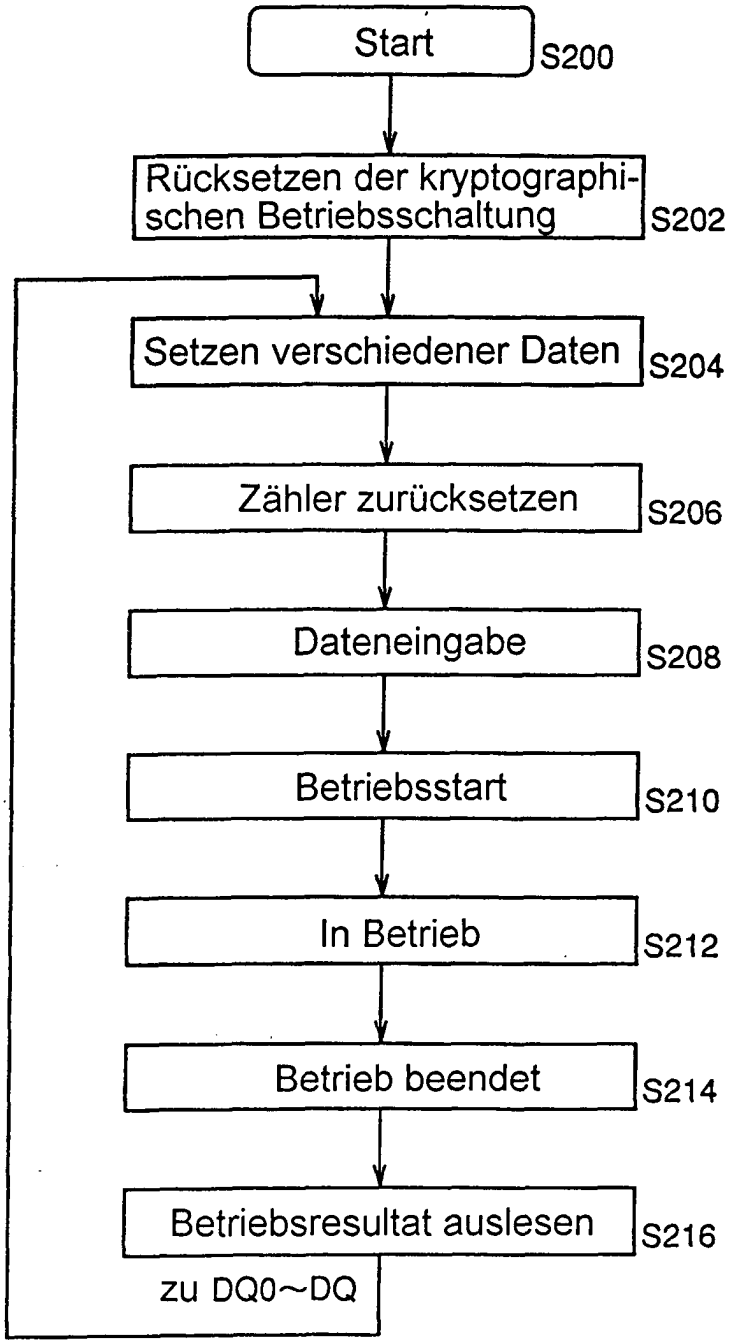

Start
S200
Rücksetzen der kryptographischen Betriebsschaltung
S202
Setzen verschiedener Daten
S204
Zähler zurücksetzen
S206
Dateneingabe
S208
Betriebsstart
S210
In Betrieb
S212
Betrieb beendet
S214
Betriebsresultat auslesen
S216
zu DQ0~DQ

FIG.43

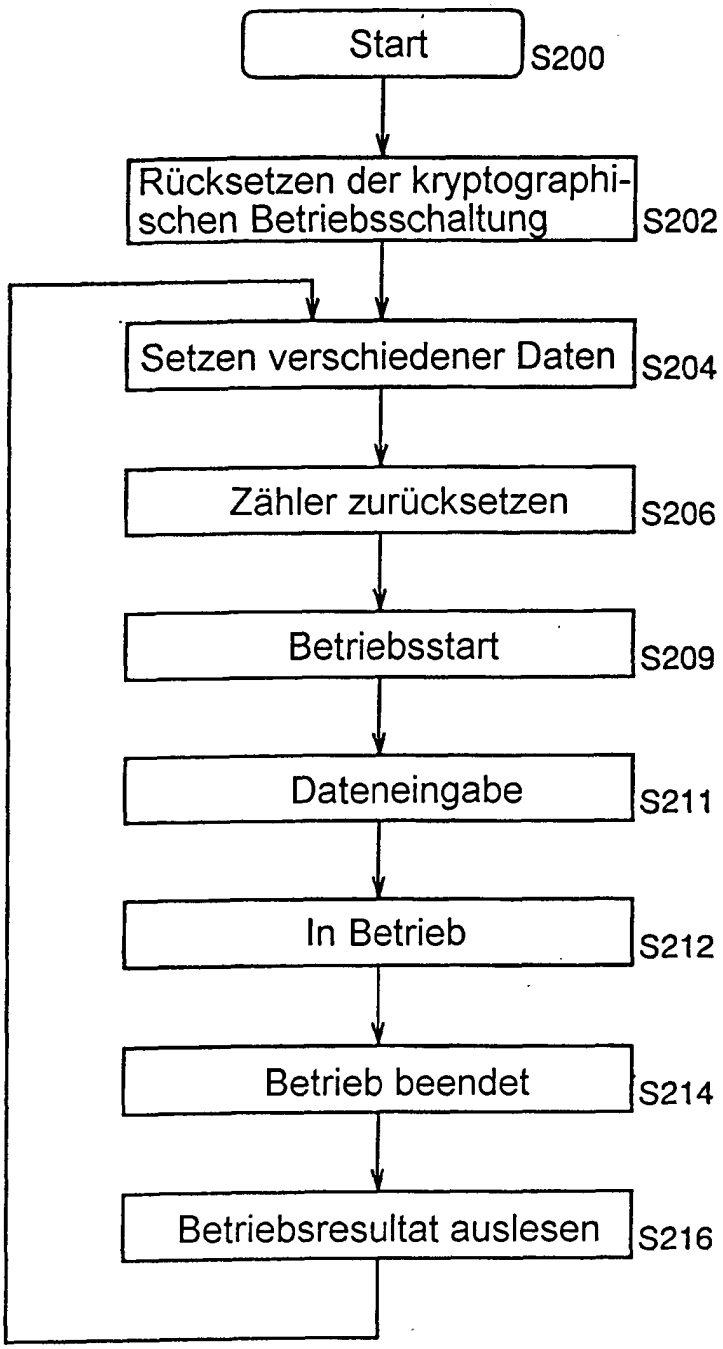

Start
S200
Rücksetzen der kryptographischen Betriebsschaltung
S202
Setzen verschiedener Daten
S204
Zähler zurücksetzen
S206
Betriebsstart
S209
Dateneingabe
S211
In Betrieb
S212
Betrieb beendet
S214
Betriebsresultat auslesen
S216

FIG.44

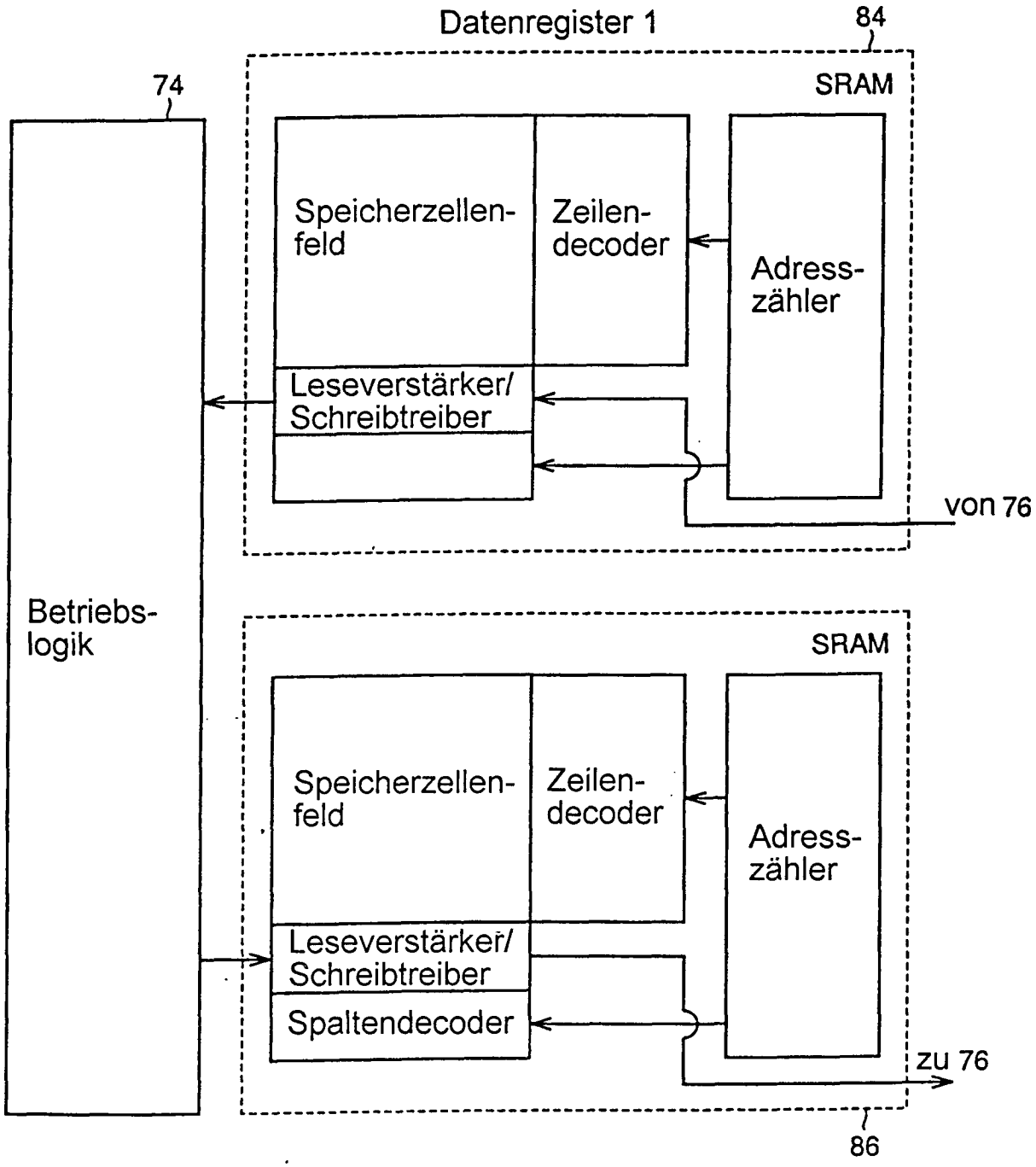

Datenregister 1
84
74
SRAM
Speicherzellen-
feld
Zeilen-
decoder
Adress-
zähler
Leseverstärker/
Schreibtreiber
von 76
Betriebs-
logik
SRAM
Speicherzellen-
feld
Zeilen-
decoder
Adress-
zähler
Leseverstärker/
Schreibtreiber
Spaltendecoder
zu 76
86

FIG.45

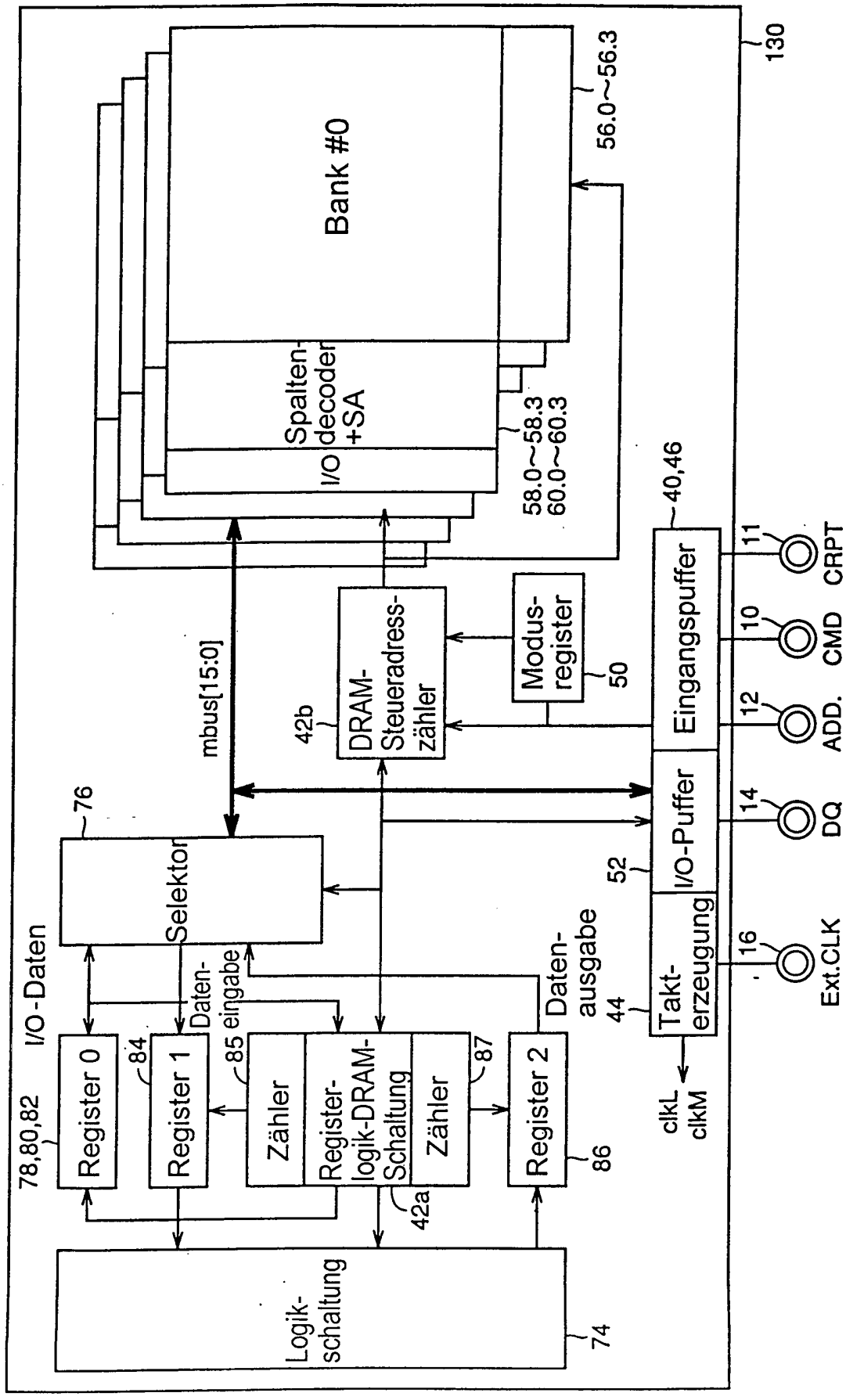

130
Bank #0
56.0~56.3
Spalten-decoder +SA
I/O
58.0~58.3
60.0~60.3
40,46
Eingangspuffer
11
CRPT
10
CMD
12
ADD.
14
DQ
16
Ext.CLK
mbus[15:0]
42b
DRAM-Steueradress-zähler
Modus-register
50
76
Selektor
I/O-Daten
52
I/O-Puffer
Takt-erzeugung
44
clkL
clkM
Daten-eingabe
Daten-ausgabe
78,80,82
Register 0
84
Register 1
85
Zähler
Register-logik-DRAM-Schaltung
Zähler
87
Register 2
86
42a
Logik-schaltung
74

FIG.46

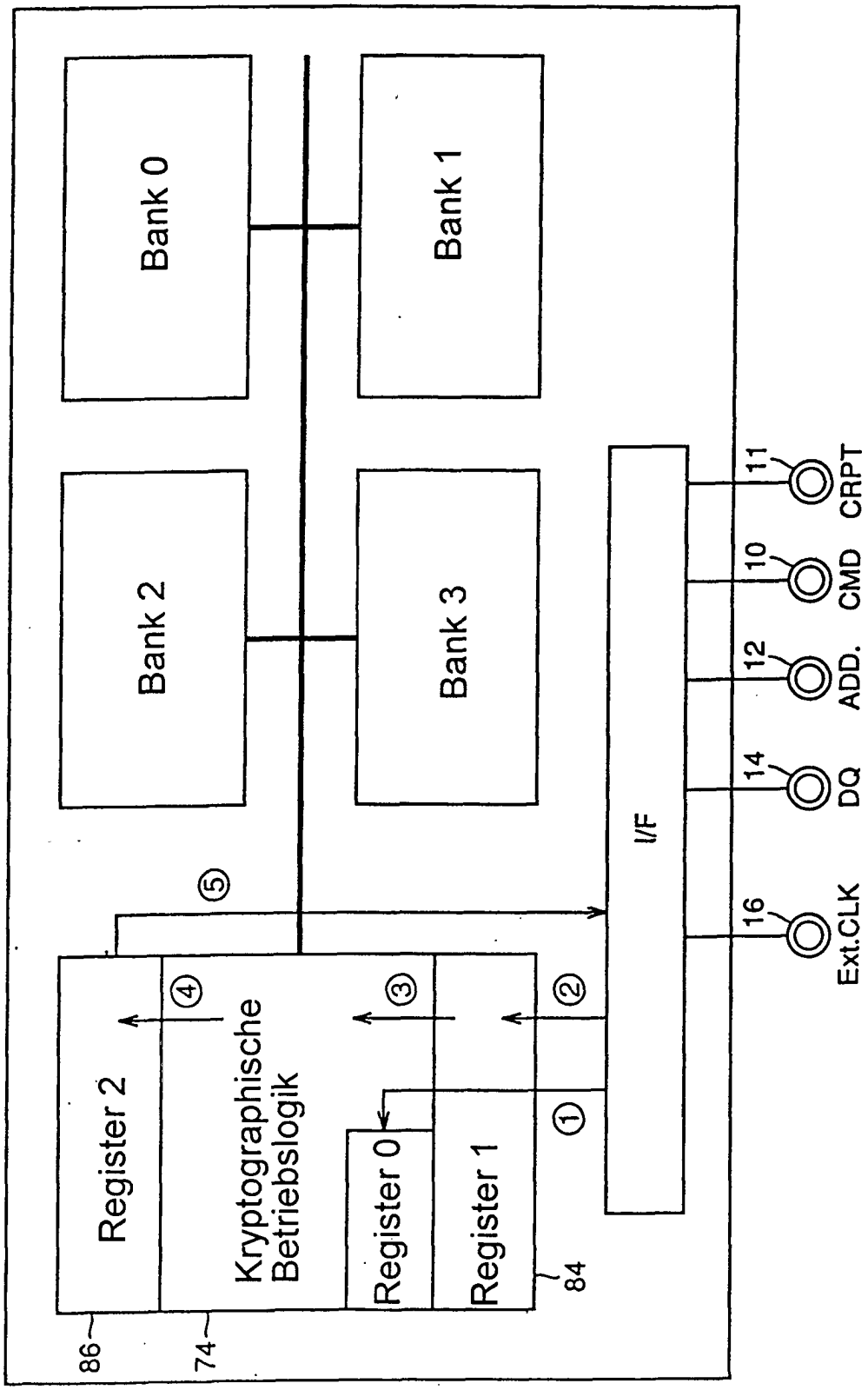

Bank 0
Bank 1
Bank 2
Bank 3
I/F
11
CRPT
10
CMD
12
ADD.
14
DQ
16
Ext.CLK
⑤
④
③
②
①
Register 2
Kryptographische Betriebslogik
Register 0
Register 1
84
86
74

FIG.47

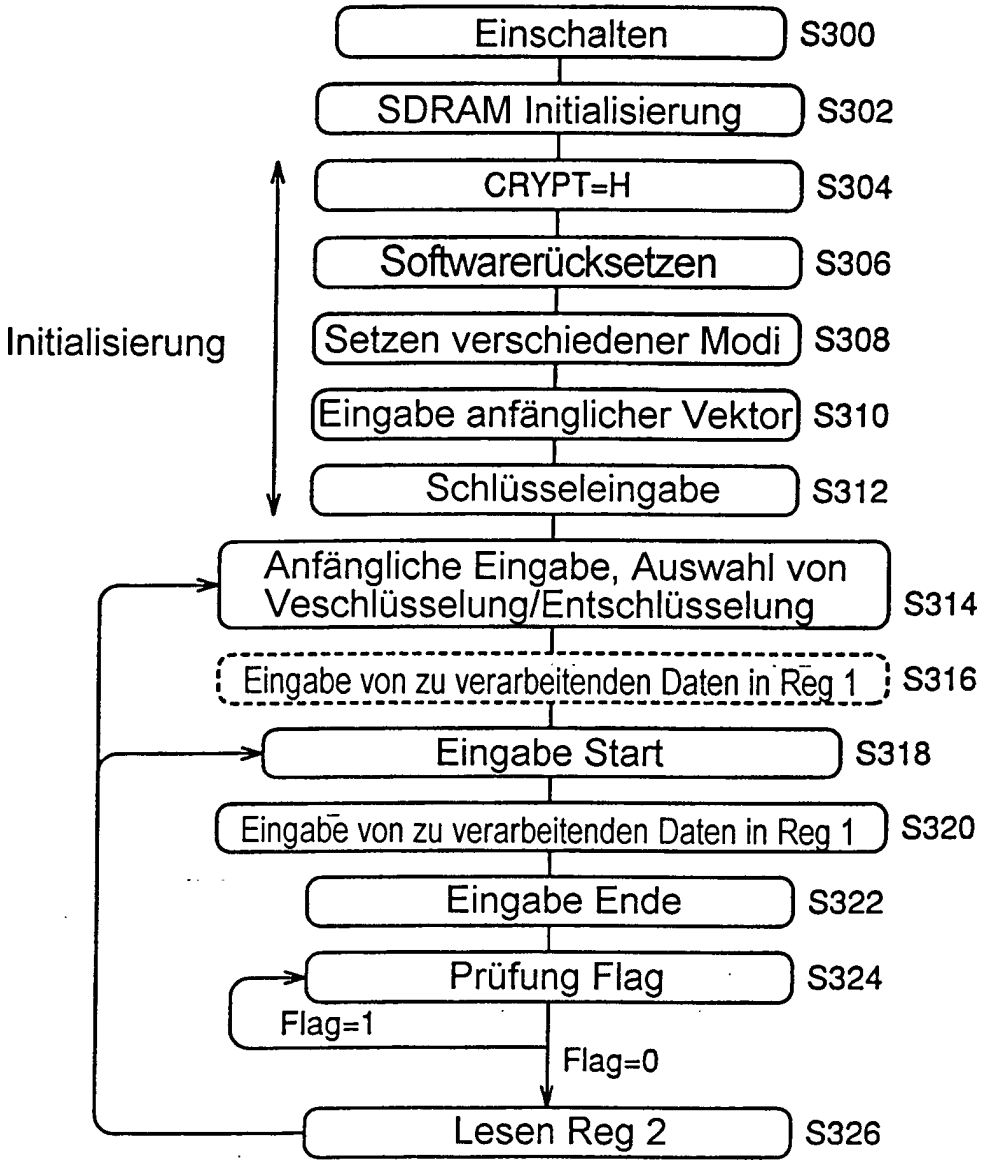

Einschalten
S300
SDRAM Initialisierung
S302
CRYPT=H
S304
Softwarerücksetzen
S306
Initialisierung
Setzen verschiedener Modi
S308
Eingabe anfänglicher Vektor
S310
Schlüsseleingabe
S312
Anfängliche Eingabe, Auswahl von Veschlüsselung/Entschlüsselung
S314
Eingabe von zu verarbeitenden Daten in Reg 1
S316
Eingabe Start
S318
Eingabe von zu verarbeitenden Daten in Reg 1
S320
Eingabe Ende
S322
Prüfung Flag
S324
Flag=1
Flag=0
Lesen Reg 2
S326

FIG.48

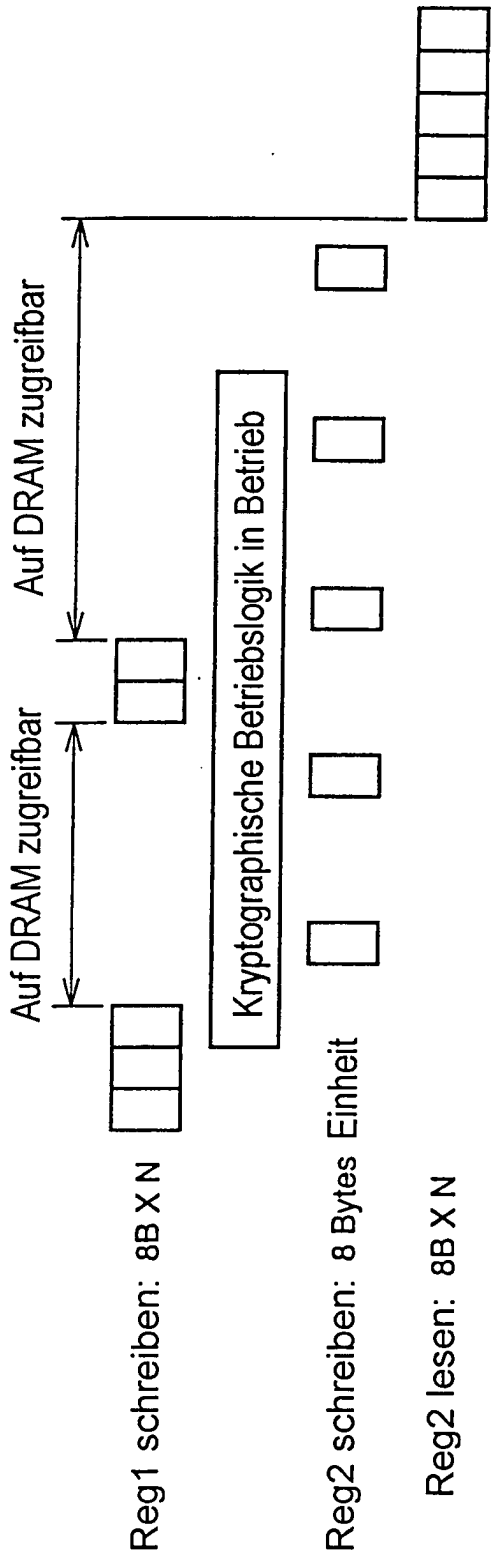
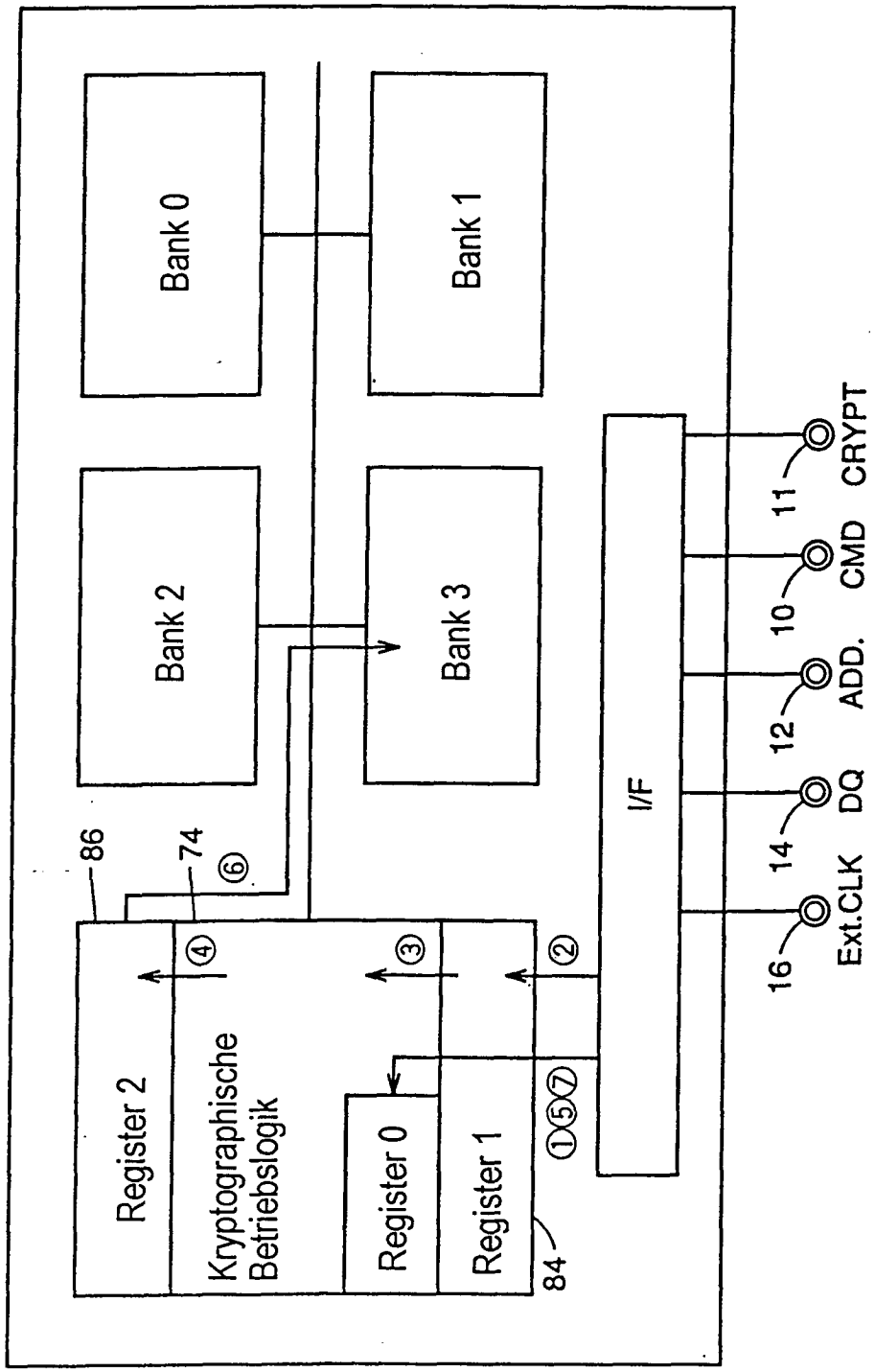

Auf DRAM zugreifbar
Auf DRAM zugreifbar
Kryptographische Betriebslogik in Betrieb
Reg1 schreiben: 8B X N
Reg2 schreiben: 8 Bytes Einheit
Reg2 lesen: 8B X N

FIG.49


Bank 0
Bank 1
Bank 2
Bank 3
I/F
Register 2
Kryptographische Betriebslogik
Register 0
Register 1
84
86
74
⑥
④
③
②
①⑤⑦
11
CRYPT
10
CMD
12
ADD.
14
DQ
16
Ext.CLK

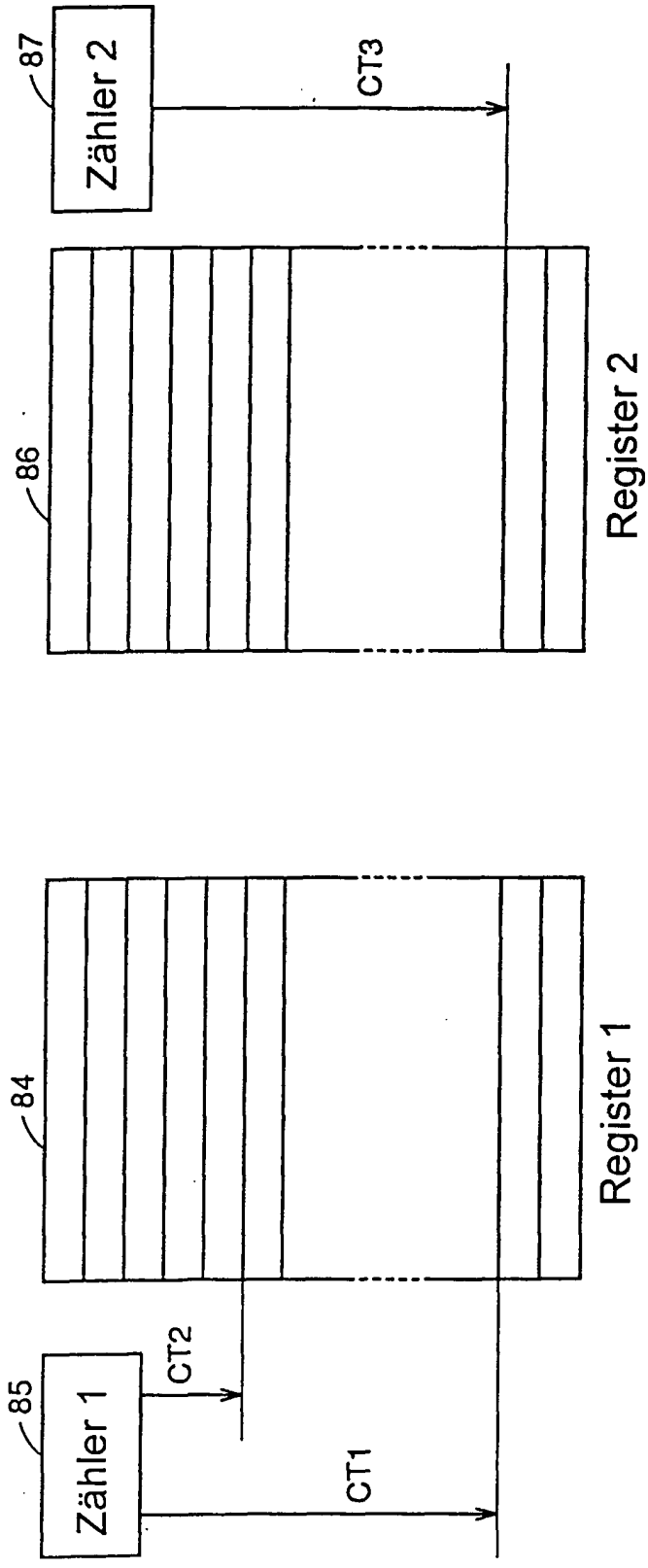


FIG.50

FIG.51

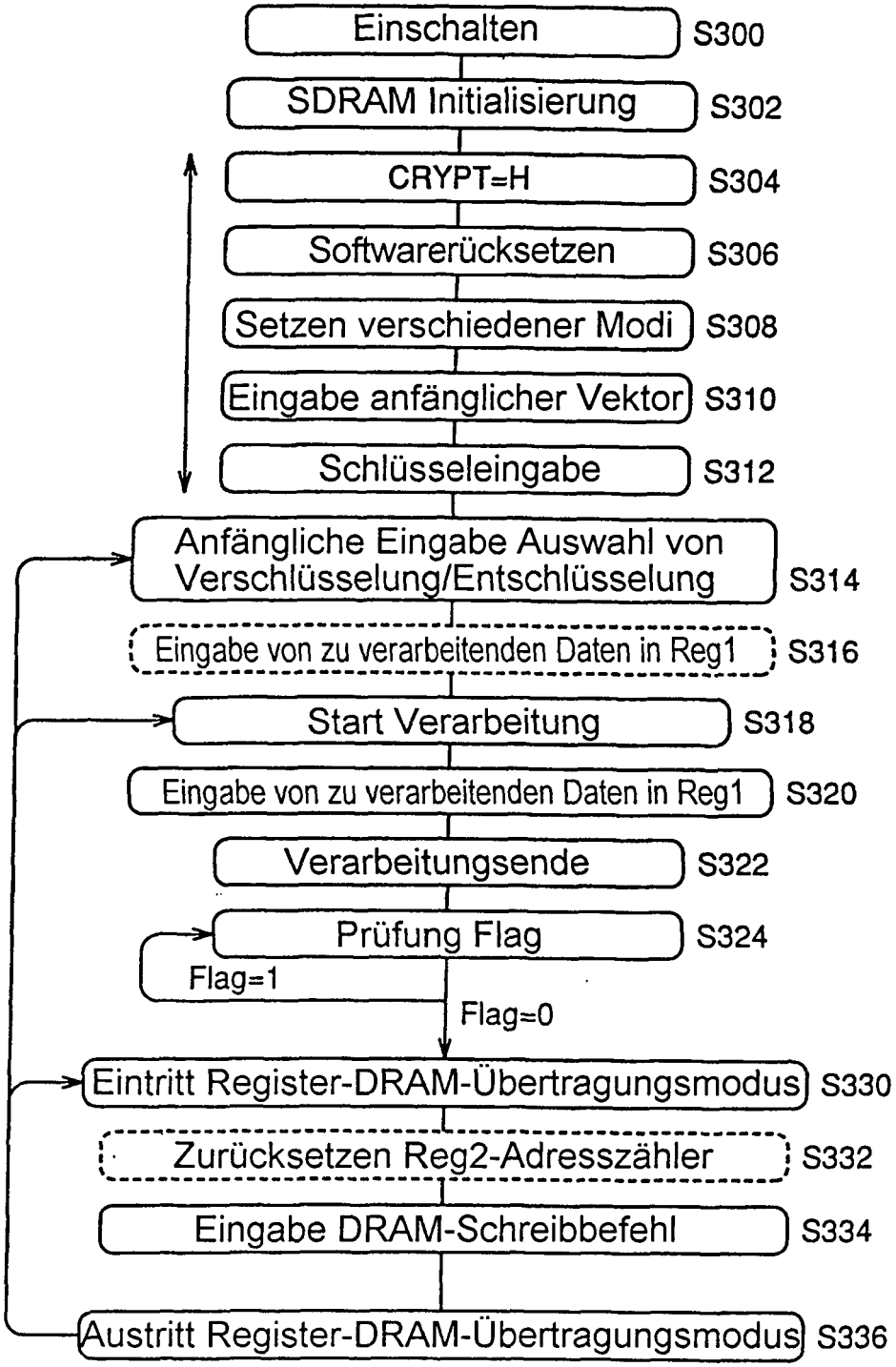

Einschalten
S300
SDRAM Initialisierung
S302
CRYPT=H
S304
Softwarerücksetzen
S306
Setzen verschiedener Modi
S308
Eingabe anfänglicher Vektor
S310
Schlüsseleingabe
S312
Anfängliche Eingabe Auswahl von Verschlüsselung/Entschlüsselung
S314
Eingabe von zu verarbeitenden Daten in Reg1
S316
Start Verarbeitung
S318
Eingabe von zu verarbeitenden Daten in Reg1
S320
Verarbeitungsende
S322
Prüfung Flag
S324
Flag=1
Flag=0
Eintritt Register-DRAM-Übertragungsmodus
S330
Zurücksetzen Reg2-Adresszähler
S332
Eingabe DRAM-Schreibbefehl
S334
Austritt Register-DRAM-Übertragungsmodus
S336

FIG.52

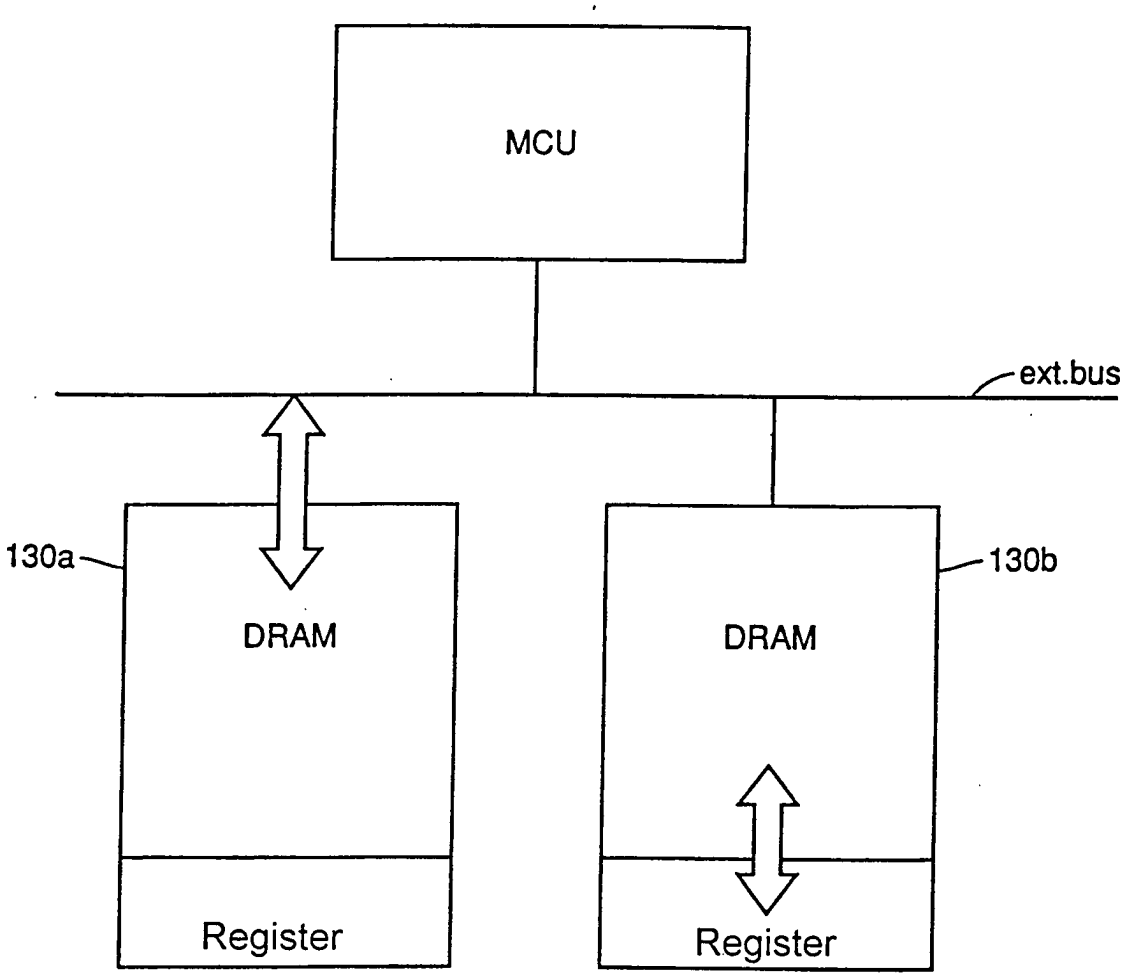


FIG.53

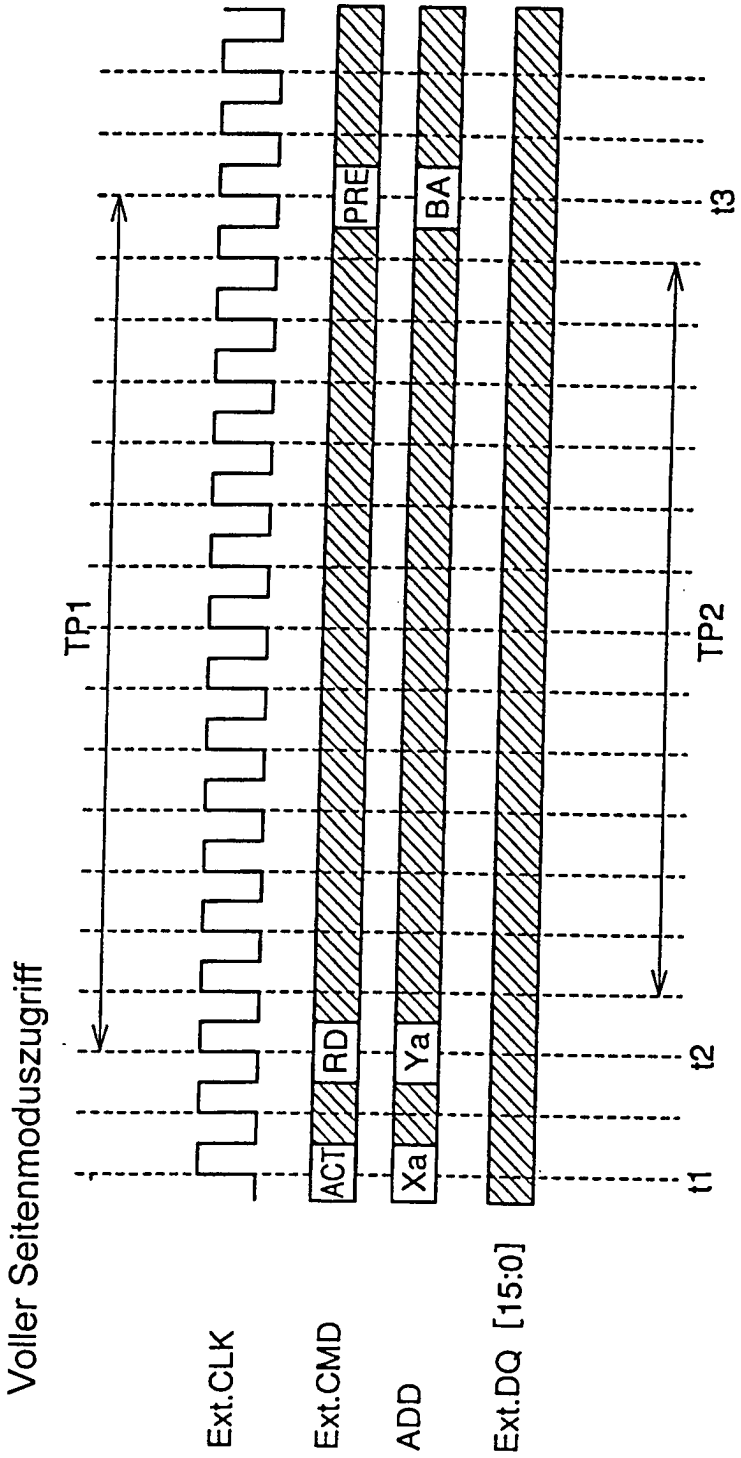

Voller Seitenmoduszugriff
Ext.CLK
Ext.CMD
ADD
Ext.DQ [15:0]
ACT
RD
PRE
Xa
Ya
BA
TP1
TP2
t1
t2
t3

FIG.54

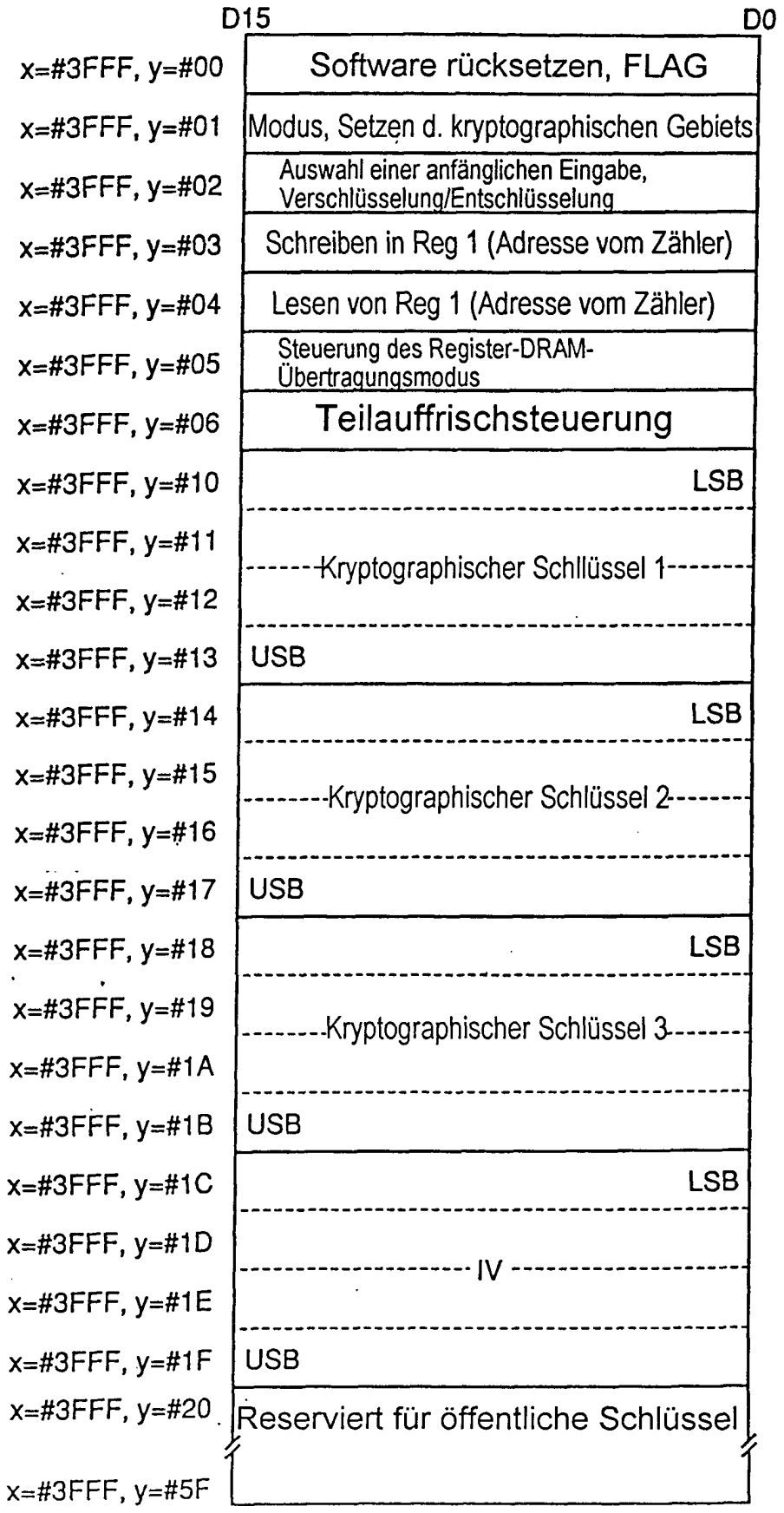

D15
D0
x=#3FFF, y=#00
Software rücksetzen, FLAG
x=#3FFF, y=#01
Modus, Setzen d. kryptographischen Gebiets
x=#3FFF, y=#02
Auswahl einer anfänglichen Eingabe, Verschlüsselung/Entschlüsselung
x=#3FFF, y=#03
Schreiben in Reg 1 (Adresse vom Zähler)
x=#3FFF, y=#04
Lesen von Reg 1 (Adresse vom Zähler)
x=#3FFF, y=#05
Steuerung des Register-DRAM-Übertragungsmodus
x=#3FFF, y=#06
Teilauffrischsteuerung
x=#3FFF, y=#10
LSB
x=#3FFF, y=#11
Kryptographischer Schllüssel 1
x=#3FFF, y=#12
x=#3FFF, y=#13
USB
x=#3FFF, y=#14
LSB
x=#3FFF, y=#15
Kryptographischer Schlüssel 2
x=#3FFF, y=#16
x=#3FFF, y=#17
USB
x=#3FFF, y=#18
LSB
x=#3FFF, y=#19
Kryptographischer Schlüssel 3
x=#3FFF, y=#1A
x=#3FFF, y=#1B
USB
x=#3FFF, y=#1C
LSB
x=#3FFF, y=#1D
IV
x=#3FFF, y=#1E
x=#3FFF, y=#1F
USB
x=#3FFF, y=#20
Reserviert für öffentliche Schlüssel
x=#3FFF, y=#5F

FIG.55

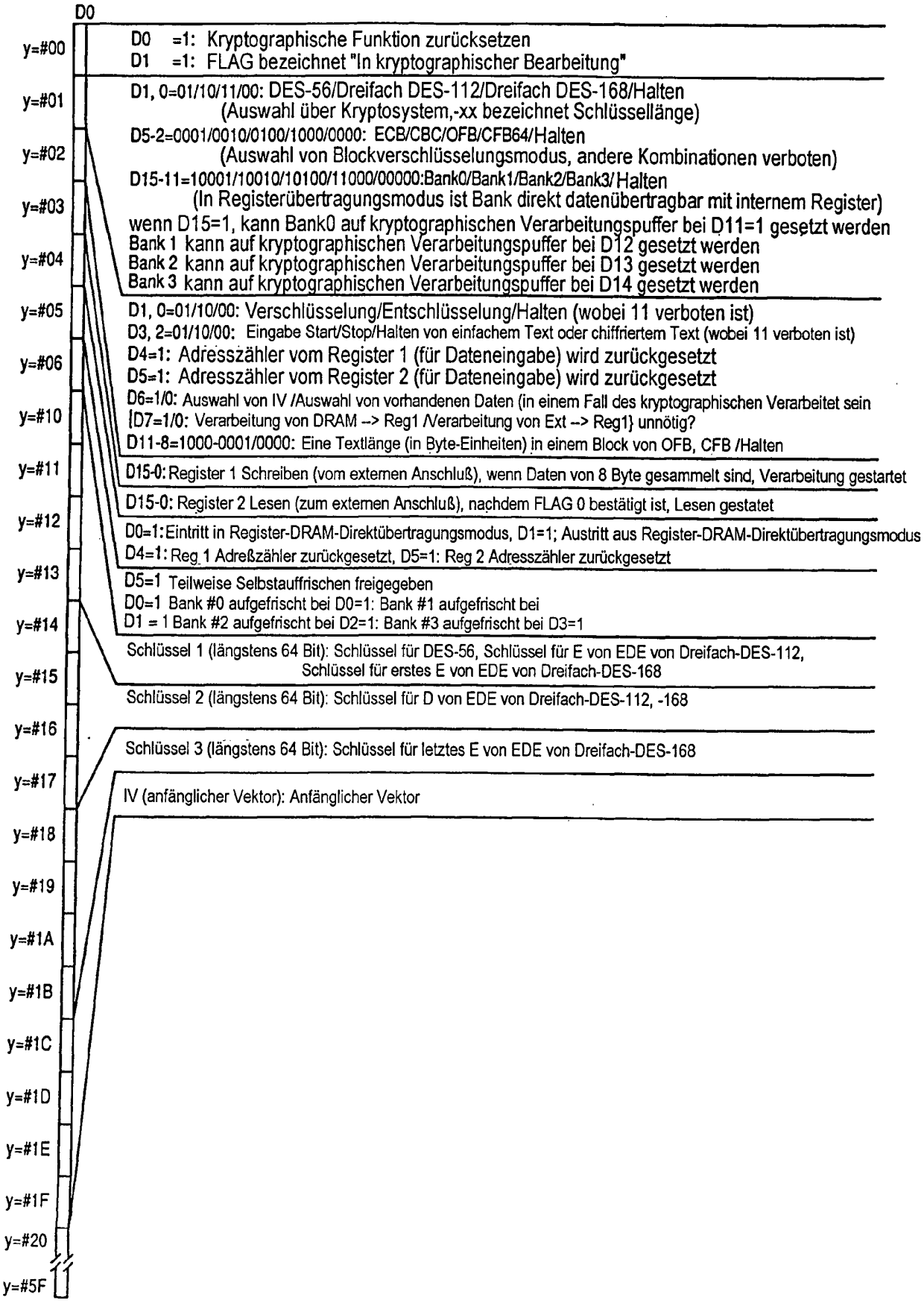

D0
y=#00
D0 =1: Kryptographische Funktion zurücksetzen
D1 =1: FLAG bezeichnet "In kryptographischer Bearbeitung"
y=#01
D1, 0=01/10/11/00: DES-56/Dreifach DES-112/Dreifach DES-168/Halten
(Auswahl über Kryptosystem,-xx bezeichnet Schlüssellänge)
y=#02
D5-2=0001/0010/0100/1000/0000: ECB/CBC/OFB/CFB64/Halten
(Auswahl von Blockverschlüsselungsmodus, andere Kombinationen verboten)
y=#03
D15-11=10001/10010/10100/11000/00000:Bank0/Bank1/Bank2/Bank3/Halten
(In Registerübertragungsmodus ist Bank direkt datenübertragbar mit internem Register)
y=#04
wenn D15=1, kann Bank0 auf kryptographischen Verarbeitungspuffer bei D11=1 gesetzt werden
Bank 1 kann auf kryptographischen Verarbeitungspuffer bei D12 gesetzt werden
Bank 2 kann auf kryptographischen Verarbeitungspuffer bei D13 gesetzt werden
Bank 3 kann auf kryptographischen Verarbeitungspuffer bei D14 gesetzt werden
y=#05
D1, 0=01/10/00: Verschlüsselung/Entschlüsselung/Halten (wobei 11 verboten ist)
D3, 2=01/10/00: Eingabe Start/Stop/Halten von einfachem Text oder chiffriertem Text (wobei 11 verboten ist)
y=#06
D4=1: Adresszähler vom Register 1 (für Dateneingabe) wird zurückgesetzt
D5=1: Adresszähler vom Register 2 (für Dateneingabe) wird zurückgesetzt
y=#10
D6=1/0: Auswahl von IV /Auswahl von vorhandenen Daten (in einem Fall des kryptographischen Verarbeitet sein
{D7=1/0: Verarbeitung von DRAM --> Reg1 /Verarbeitung von Ext --> Reg1} unnötig?
D11-8=1000-0001/0000: Eine Textlänge (in Byte-Einheiten) in einem Block von OFB, CFB /Halten
y=#11
D15-0: Register 1 Schreiben (vom externen Anschluß), wenn Daten von 8 Byte gesammelt sind, Verarbeitung gestartet
y=#12
D15-0: Register 2 Lesen (zum externen Anschluß), nachdem FLAG 0 bestätigt ist, Lesen gestatet
y=#13
D0=1: Eintritt in Register-DRAM-Direktübertragungsmodus, D1=1; Austritt aus Register-DRAM-Direktübertragungsmodus
D4=1: Reg 1 Adreßzähler zurückgesetzt, D5=1: Reg 2 Adresszähler zurückgesetzt
y=#14
D5=1 Teilweise Selbstauffrischen freigegeben
D0=1 Bank #0 aufgefrischt bei D0=1: Bank #1 aufgefrischt bei
D1 = 1 Bank #2 aufgefrischt bei D2=1: Bank #3 aufgefrischt bei D3=1
y=#15
Schlüssel 1 (längstens 64 Bit): Schlüssel für DES-56, Schlüssel für E von EDE von Dreifach-DES-112,
Schlüssel für erstes E von EDE von Dreifach-DES-168
Schlüssel 2 (längstens 64 Bit): Schlüssel für D von EDE von Dreifach-DES-112, -168
y=#16
Schlüssel 3 (längstens 64 Bit): Schlüssel für letztes E von EDE von Dreifach-DES-168
y=#17
IV (anfänglicher Vektor): Anfänglicher Vektor
y=#18
y=#19
y=#1A
y=#1B
y=#1C
y=#1D
y=#1E
y=#1F
y=#20
y=#5F

FIG.56

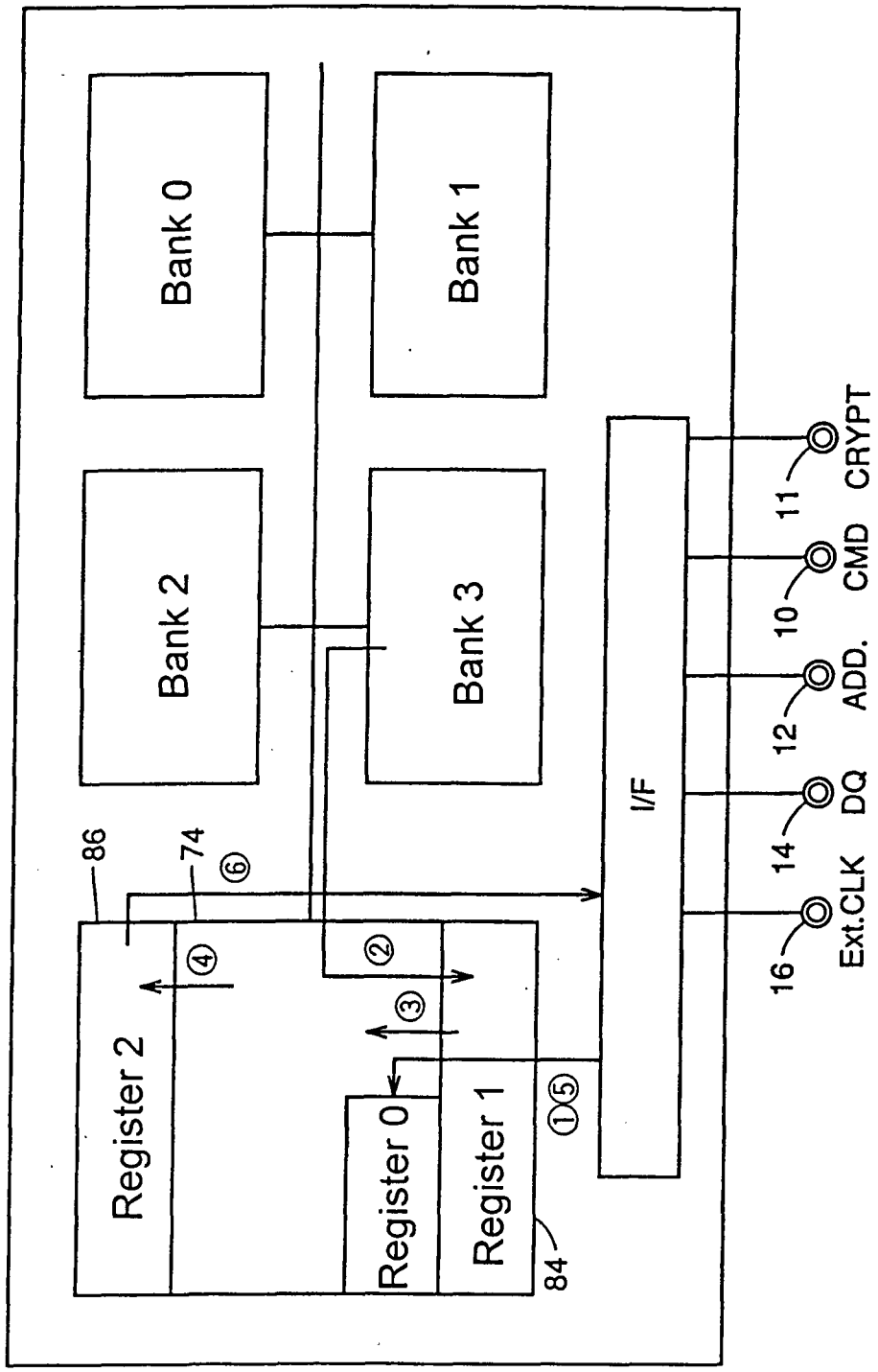

Bank 0
Bank 1
Bank 2
Bank 3
I/F
Register 2
Register 0
Register 1
84
86
74
⑥
④
②
③
①⑤
11
CRYPT
10
CMD
12
ADD.
14
DQ
16
Ext.CLK

FIG.57

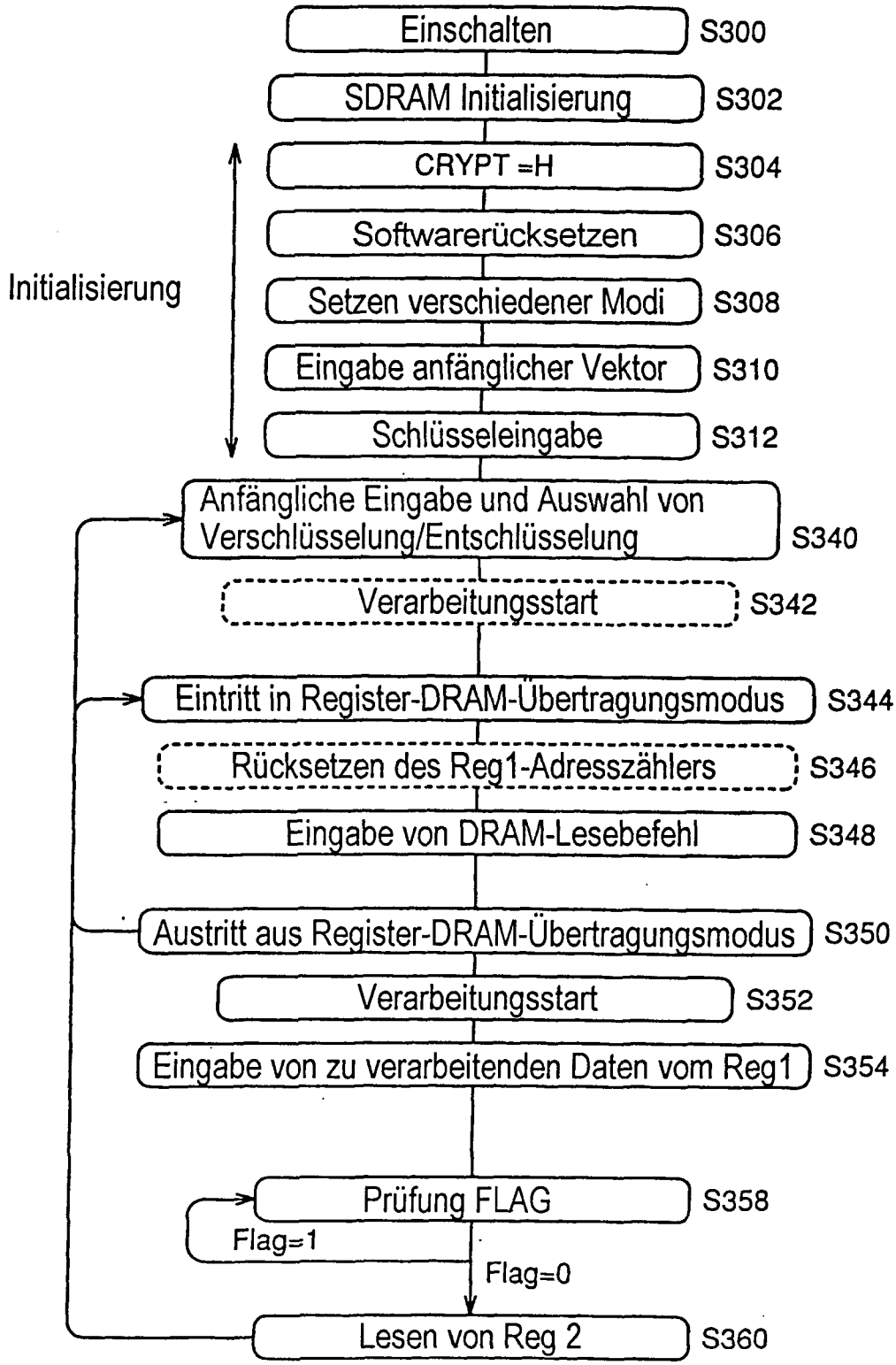
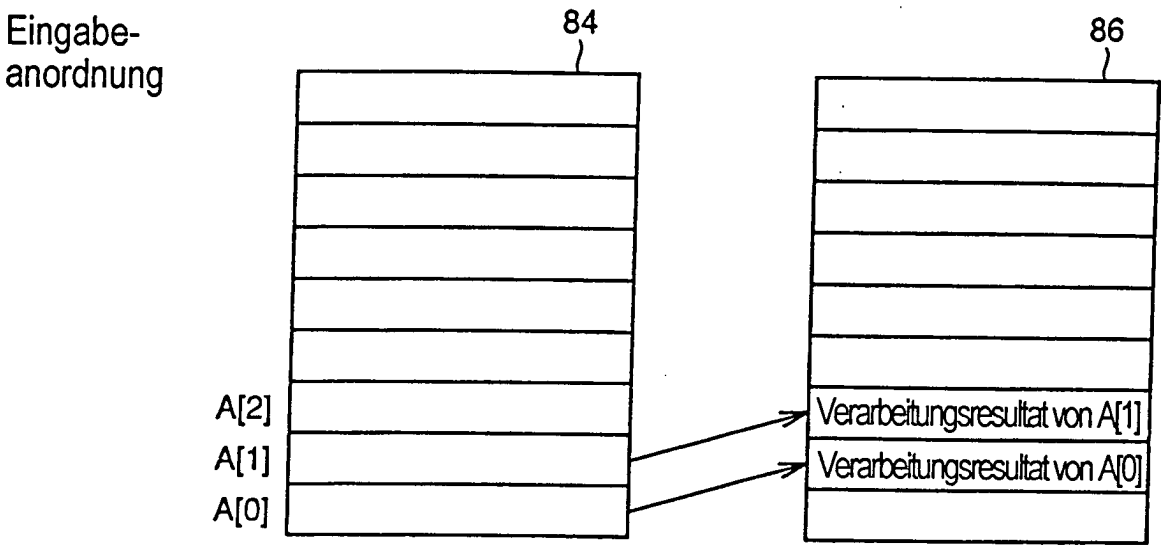

Einschalten
S300
SDRAM Initialisierung
S302
CRYPT =H
S304
Softwarerücksetzen
S306
Initialisierung
Setzen verschiedener Modi
S308
Eingabe anfänglicher Vektor
S310
Schlüsseleingabe
S312
Anfängliche Eingabe und Auswahl von Verschlüsselung/Entschlüsselung
S340
Verarbeitungsstart
S342
Eintritt in Register-DRAM-Übertragungsmodus
S344
Rücksetzen des Reg1-Adresszählers
S346
Eingabe von DRAM-Lesebefehl
S348
Austritt aus Register-DRAM-Übertragungsmodus
S350
Verarbeitungsstart
S352
Eingabe von zu verarbeitenden Daten vom Reg1
S354
Prüfung FLAG
S358
Flag=1
Flag=0
Lesen von Reg 2
S360

FIG.58



BL = 1 wenn auf Steuerbefehlsgebiet zugegriffen

FIG.59

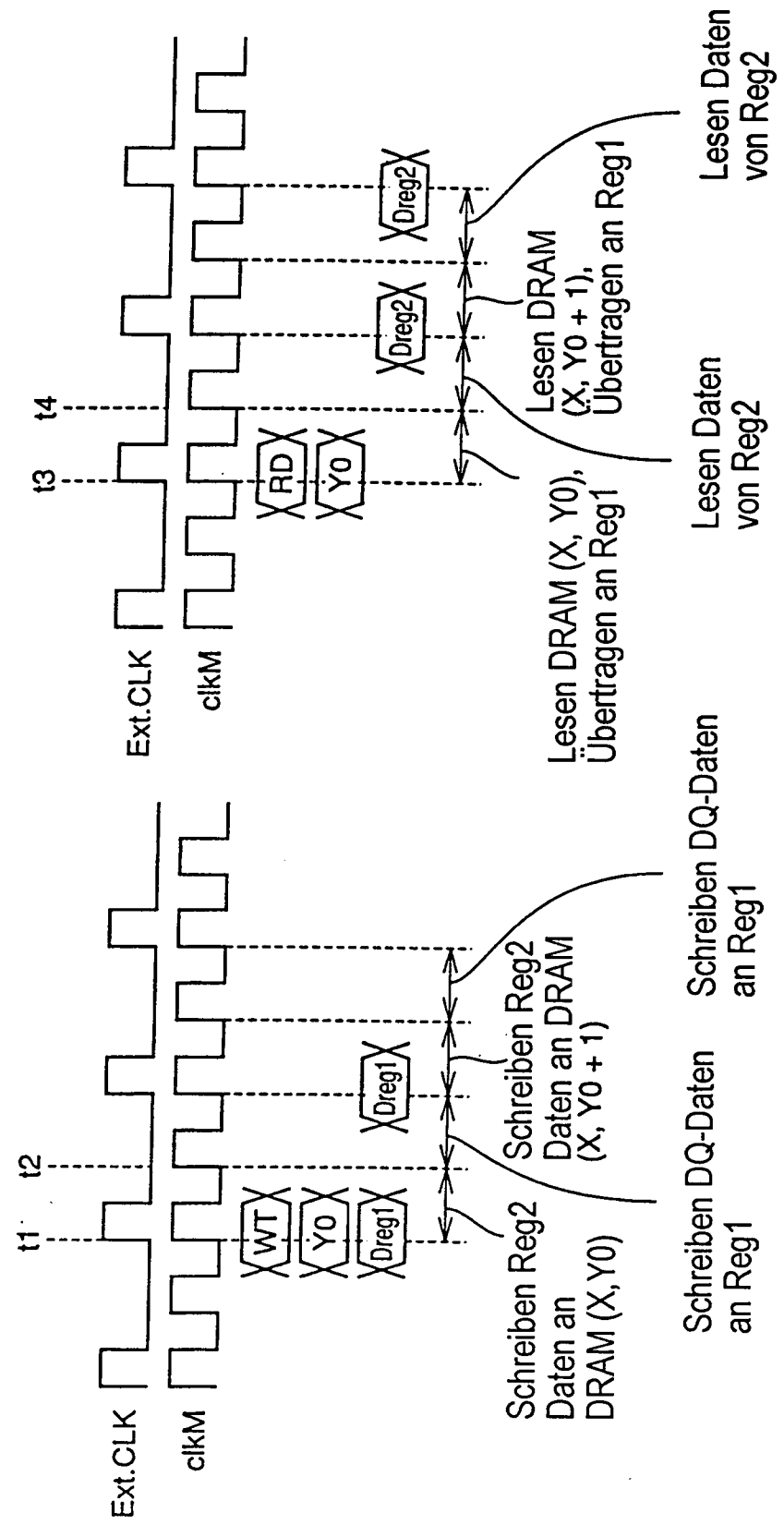

Ext.CLK
clkM
t1
t2
t3
t4
WT
Y0
Dreg1
RD
Dreg2
Schreiben Reg2 Daten an DRAM (X,Y0)
Schreiben Reg2 Daten an DRAM (X, Y0 + 1)
Schreiben DQ-Daten an Reg1
Lesen DRAM (X, Y0), Übertragen an Reg1
Lesen DRAM (X, Y0 + 1), Übertragen an Reg1
Lesen Daten von Reg2

FIG.60

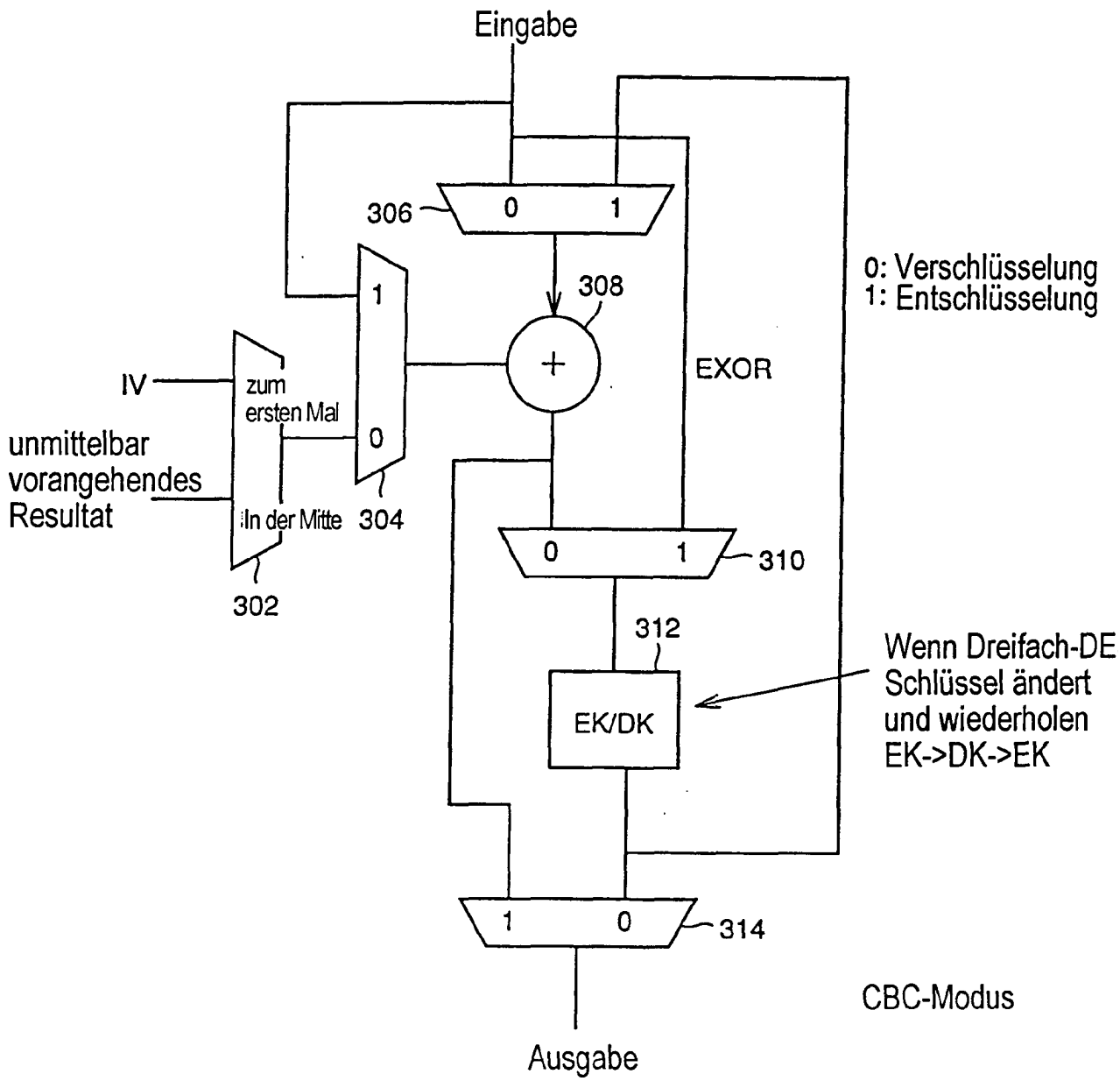

Eingabe
306
0
1
308
1
0
IV
zum ersten Mal
unmittelbar vorangehendes Resultat
In der Mitte
304
302
EXOR
0
1
310
312
EK/DK
0: Verschlüsselung
1: Entschlüsselung
Wenn Dreifach-DE Schlüssel ändert und wiederholen EK->DK->EK
1
0
314
CBC-Modus
Ausgabe

FIG.61

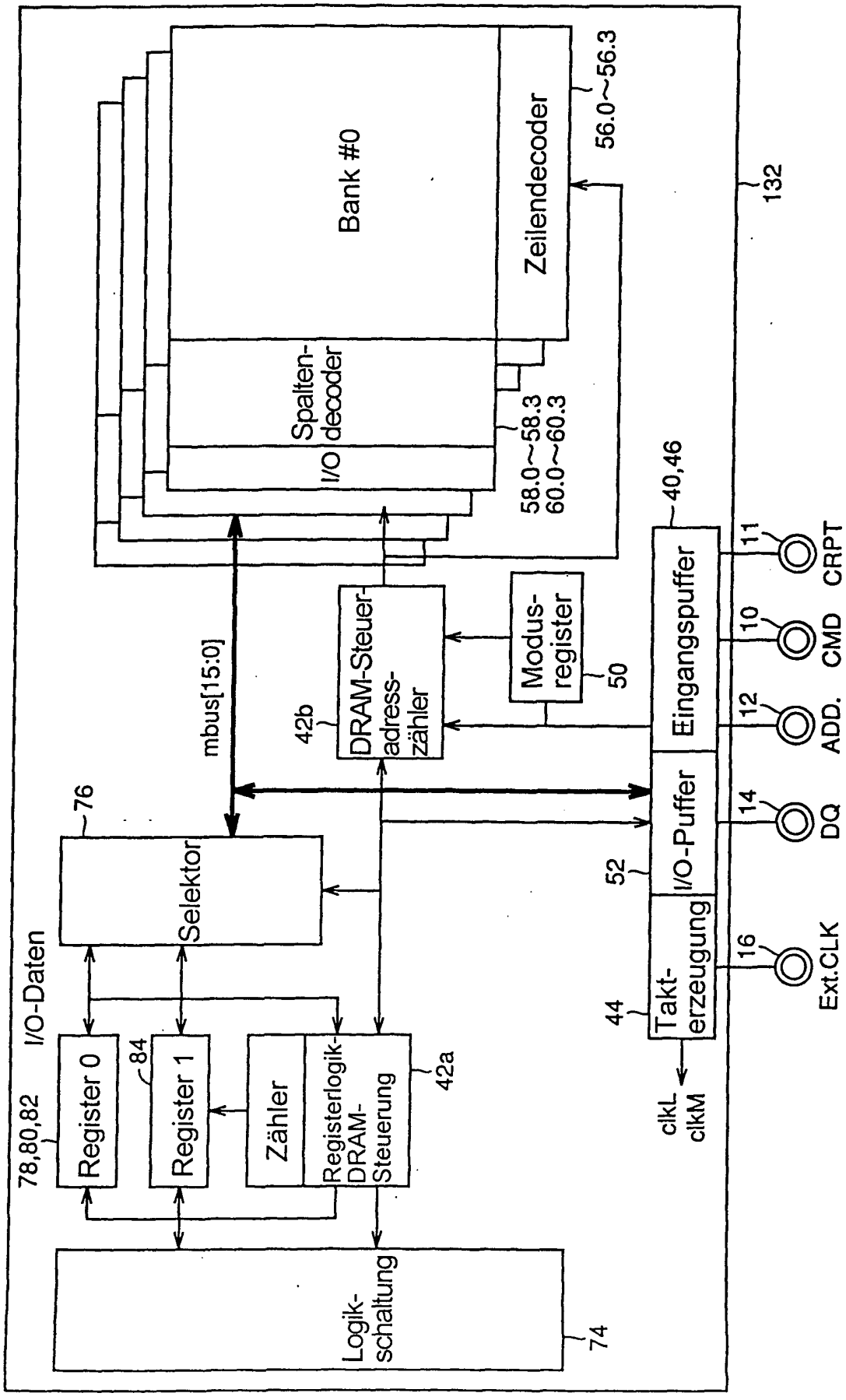

Bank #0
Zeilendecoder
56.0~56.3
132
Spalten-decoder
I/O
58.0~58.3
60.0~60.3
40,46
Eingangspuffer
11
CRPT
10
CMD
12
ADD.
DRAM-Steuer-adress-zähler
42b
Modus-register
50
mbus[15:0]
76
Selektor
I/O-Daten
I/O-Puffer
14
DQ
52
78,80,82
Register 0
84
Register 1
Zähler
Registerlogik-DRAM-Steuerung
42a
Takt-erzeugung
16
Ext.CLK
44
clkL
clkM
Logik-schaltung
74

FIG.62

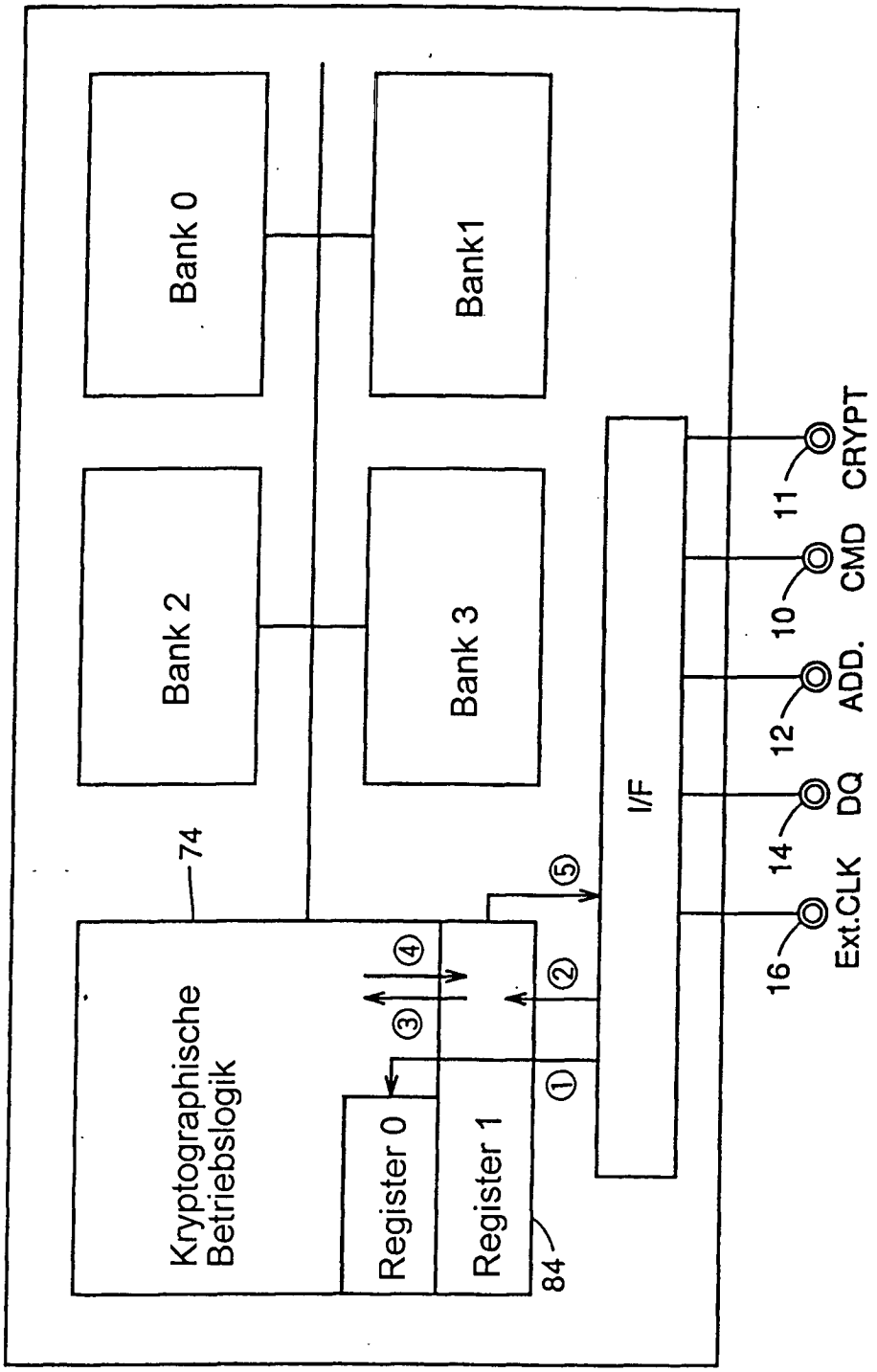
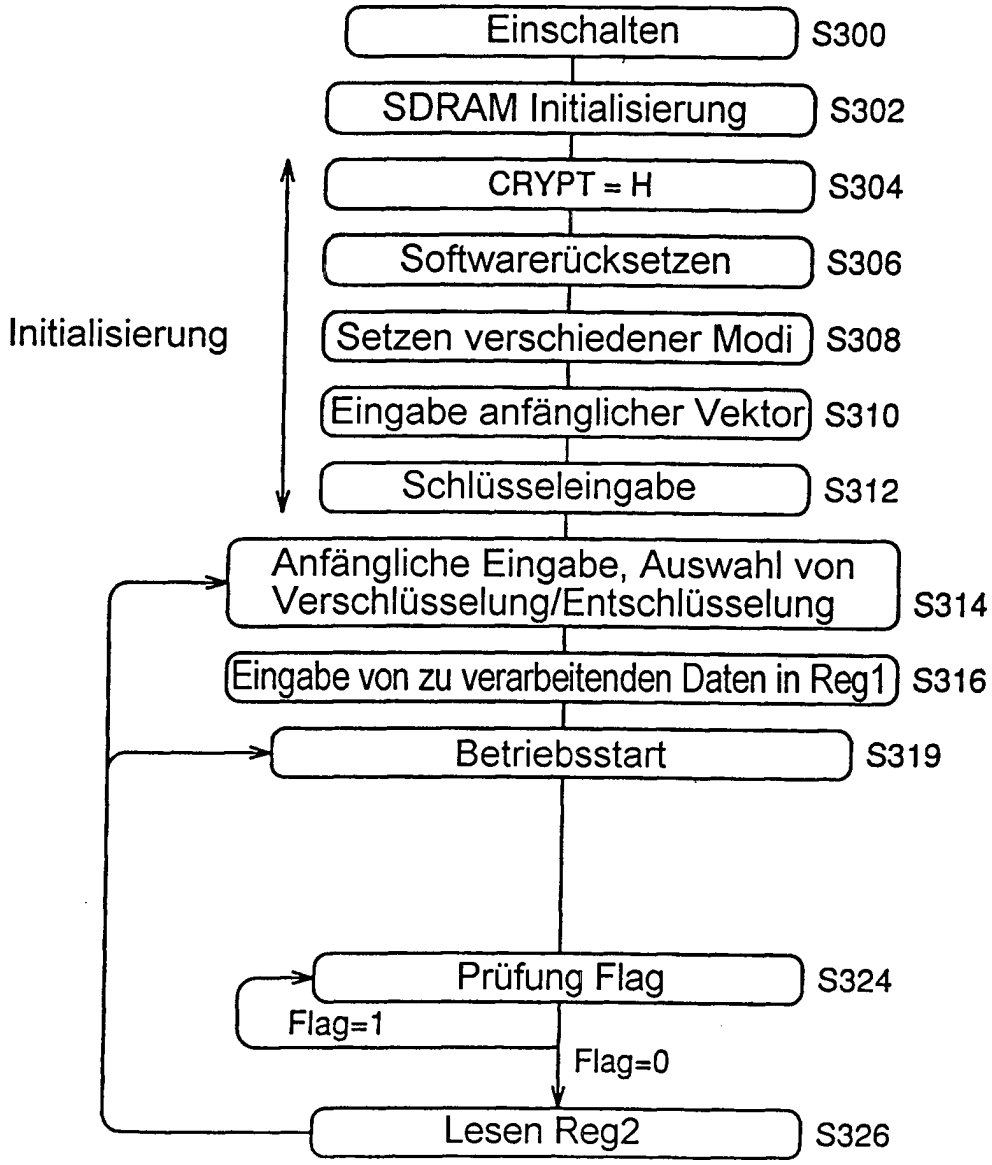

Bank 0
Bank1
Bank 2
Bank 3
74
Kryptographische Betriebslogik
Register 0
Register 1
84
I/F
①
②
③
④
⑤
11
CRYPT
10
CMD
12
ADD.
14
DQ
16
Ext.CLK

FIG.63


Einschalten
S300
SDRAM Initialisierung
S302
CRYPT = H
S304
Softwarerücksetzen
S306
Initialisierung
Setzen verschiedener Modi
S308
Eingabe anfänglicher Vektor
S310
Schlüsseleingabe
S312
Anfängliche Eingabe, Auswahl von Verschlüsselung/Entschlüsselung
S314
Eingabe von zu verarbeitenden Daten in Reg1
S316
Betriebsstart
S319
Prüfung Flag
S324
Flag=1
Flag=0
Lesen Reg2
S326

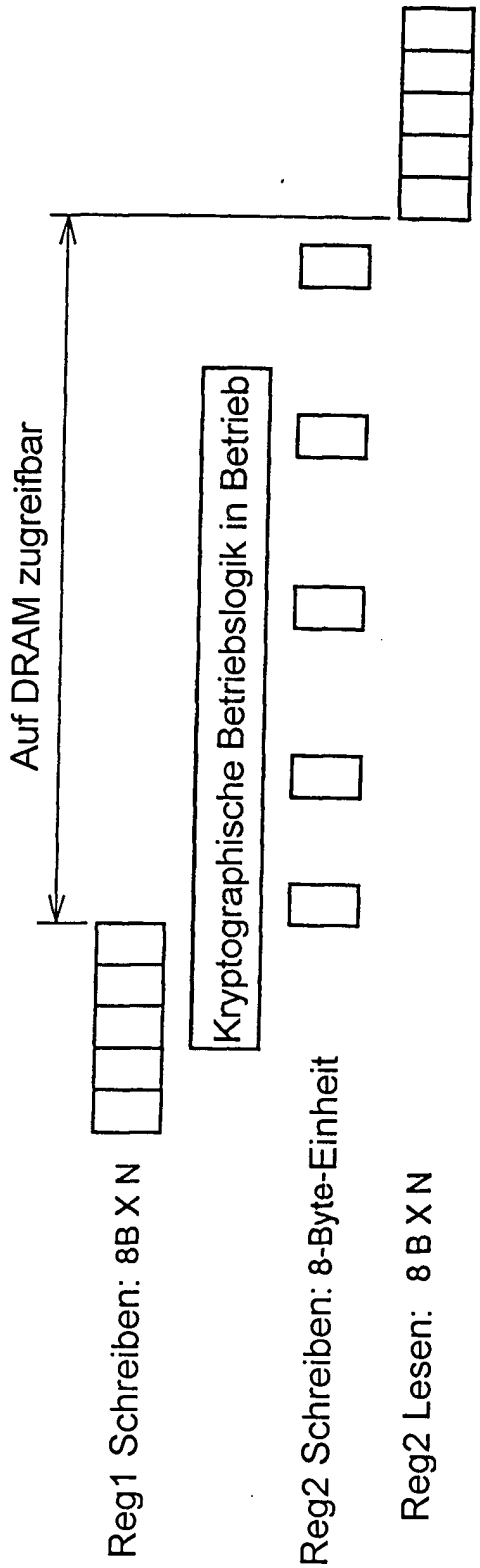

Auf DRAM zugreifbar
Kryptographische Betriebslogik in Betrieb
Reg1 Schreiben: 8B X N
Reg2 Schreiben: 8-Byte-Einheit
Reg2 Lesen: 8 B X N

FIG.64

FIG.65

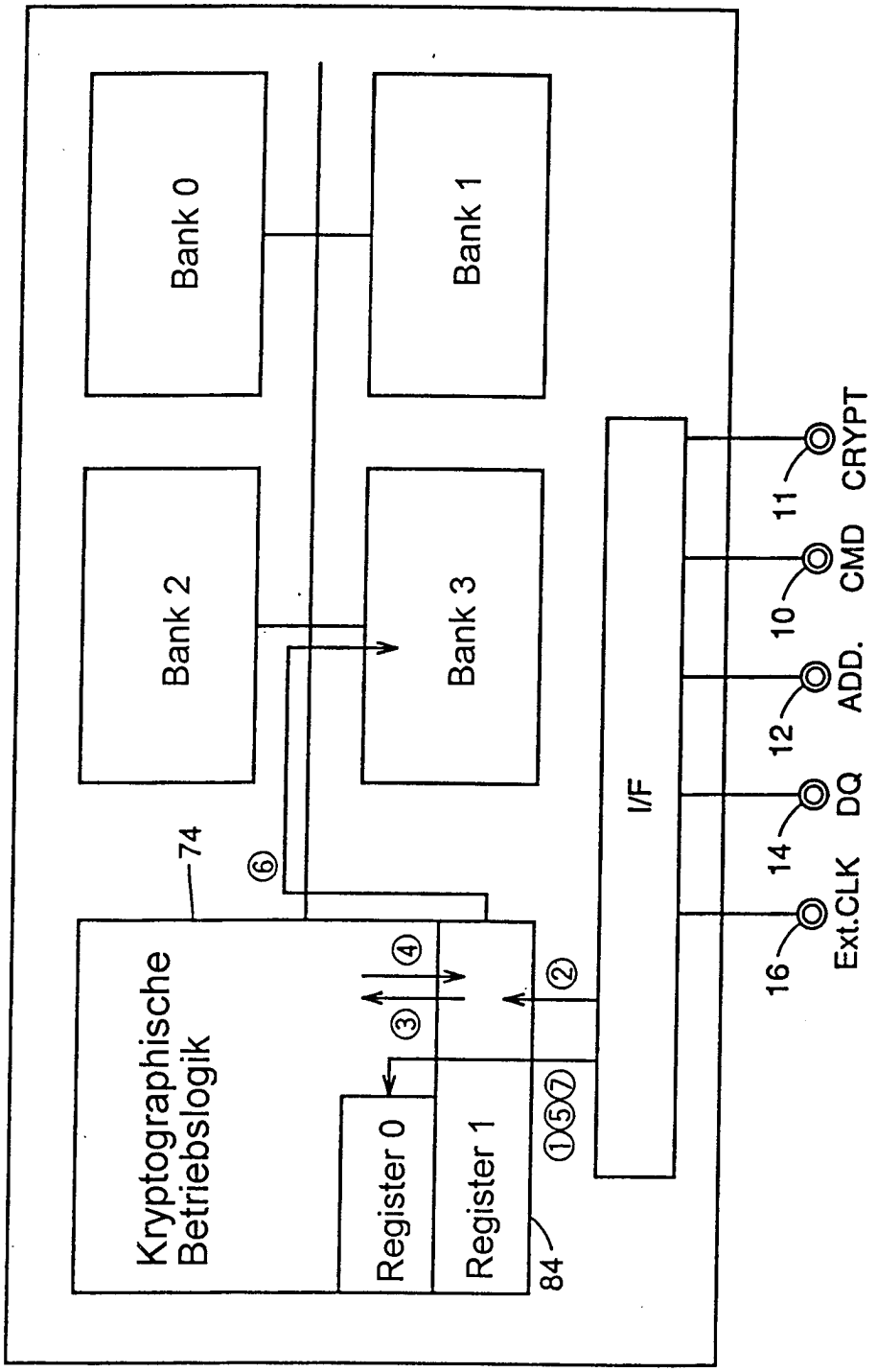

Bank 0
Bank 1
Bank 2
Bank 3
74
Kryptographische Betriebslogik
Register 0
Register 1
84
I/F
11
CRYPT
10
CMD
12
ADD.
14
DQ
16
Ext.CLK

FIG.66

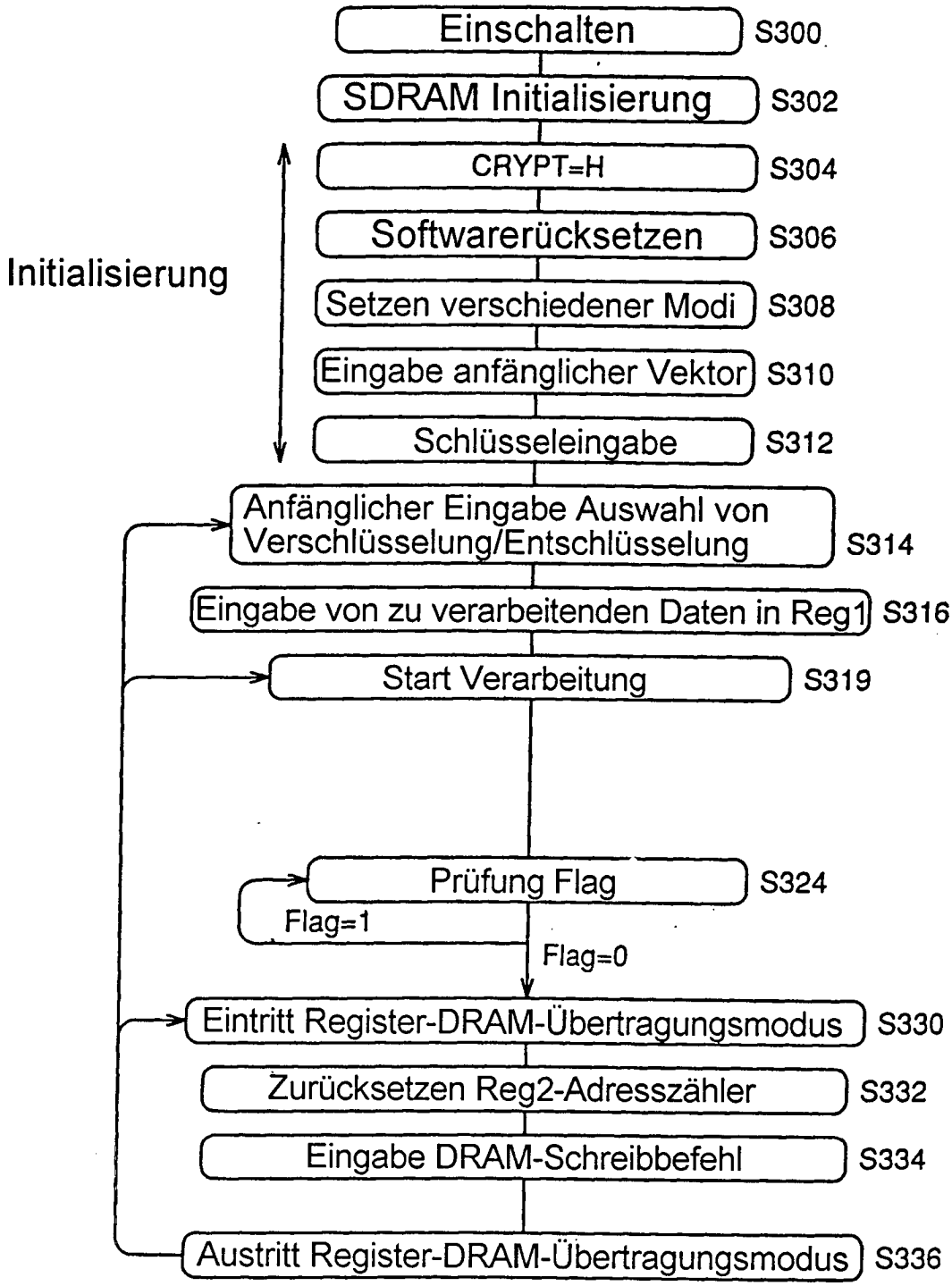

Einschalten
S300
SDRAM Initialisierung
S302
CRYPT=H
S304
Softwarerücksetzen
S306
Initialisierung
Setzen verschiedener Modi
S308
Eingabe anfänglicher Vektor
S310
Schlüsseleingabe
S312
Anfänglicher Eingabe Auswahl von Verschlüsselung/Entschlüsselung
S314
Eingabe von zu verarbeitenden Daten in Reg1
S316
Start Verarbeitung
S319
Prüfung Flag
S324
Flag=1
Flag=0
Eintritt Register-DRAM-Übertragungsmodus
S330
Zurücksetzen Reg2-Adresszähler
S332
Eingabe DRAM-Schreibbefehl
S334
Austritt Register-DRAM-Übertragungsmodus
S336

FIG.67

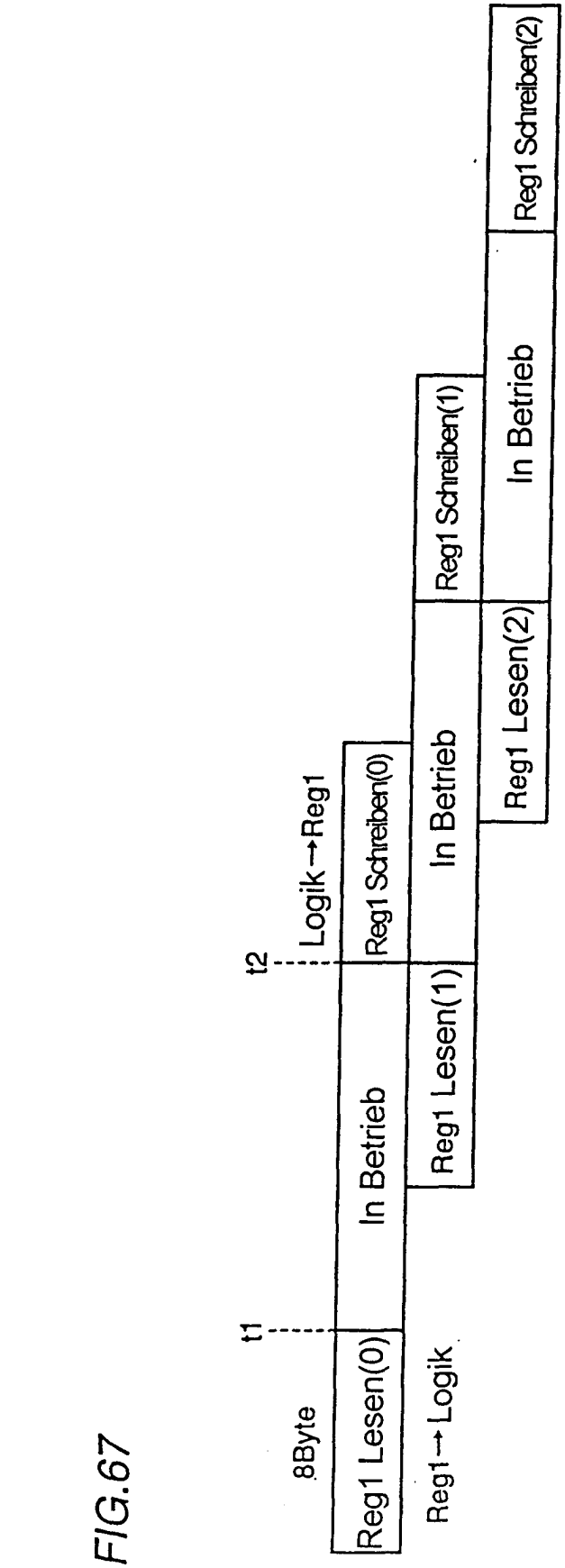

8Byte
t1
t2
Reg1 Lesen(0)
In Betrieb
Reg1 Schreiben(0)
Reg1 Lesen(1)
In Betrieb
Reg1 Schreiben(1)
Reg1 Lesen(2)
In Betrieb
Reg1 Schreiben(2)
Reg1→Logik
Logik→Reg1
Nächste Daten werden im Voraus aus Reg1 in Betrieb gelesen
<-- Schreiben des Betriebsresultates in Reg1 kann verborgen werden

FIG.68

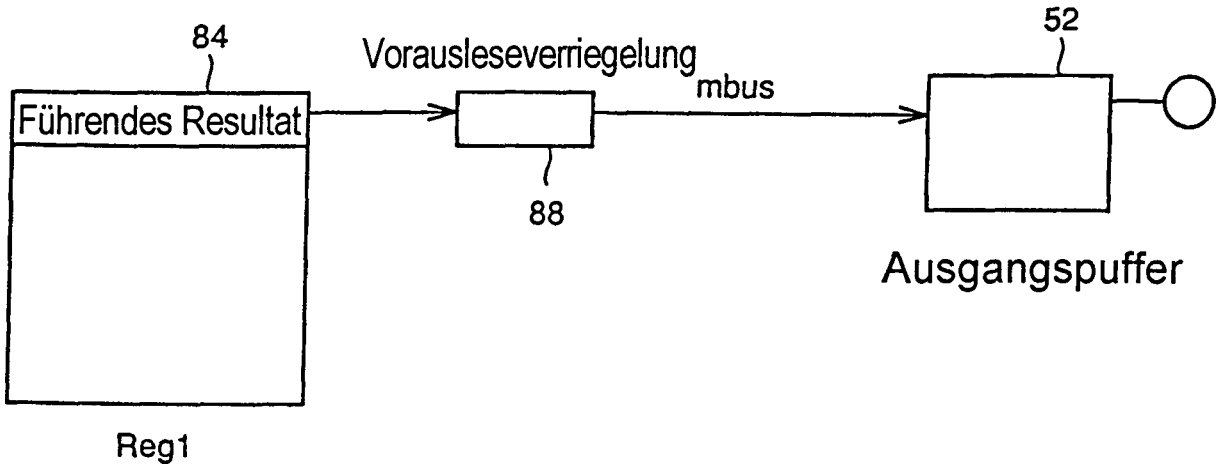

84
Führendes Resultat
Reg1
Vorausleseverriegelung
88
mbus
52
Ausgangspuffer

FIG.69

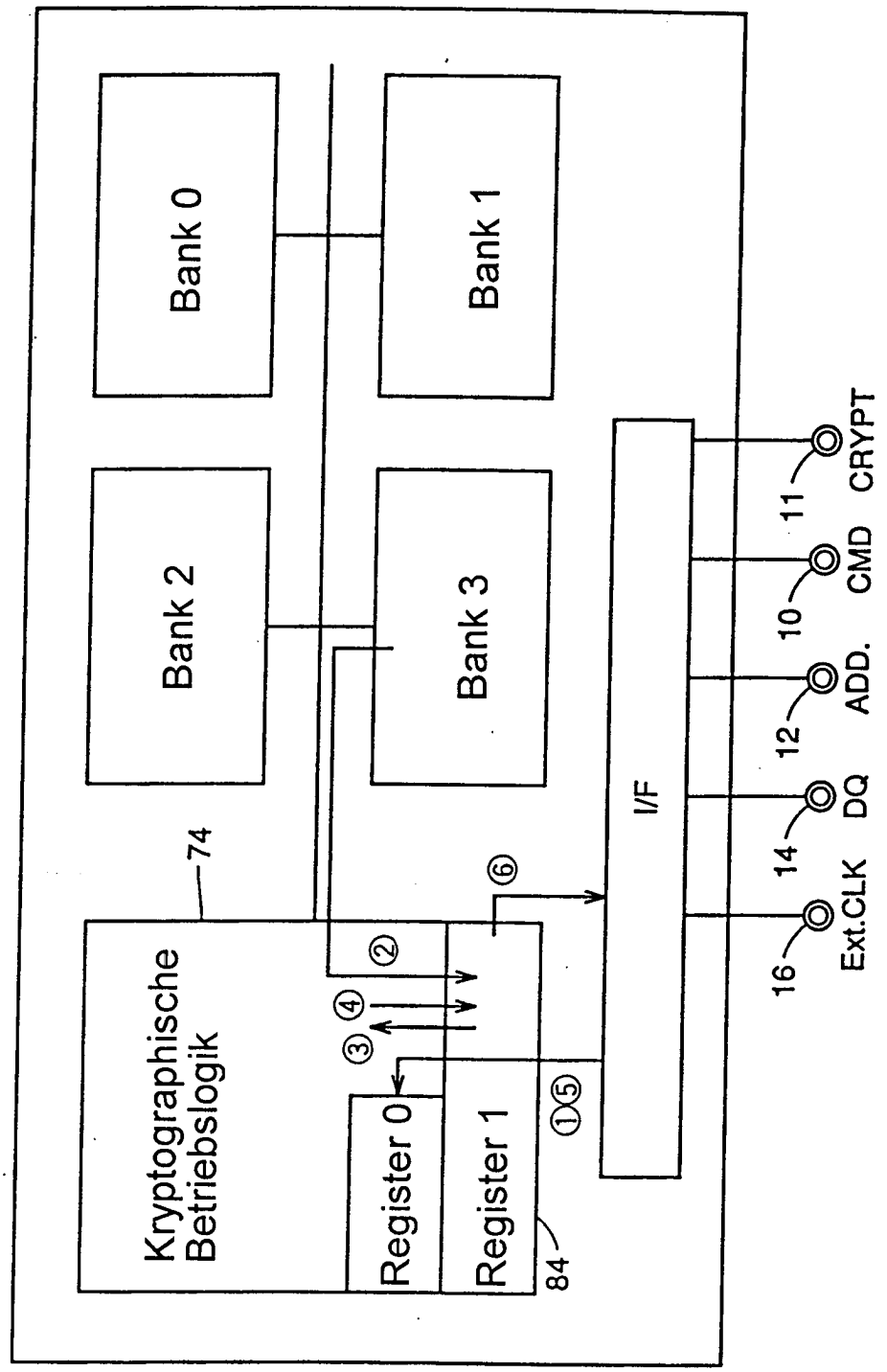

Bank 0
Bank 1
Bank 2
Bank 3
74
Kryptographische Betriebslogik
Register 0
Register 1
84
I/F
11
CRYPT
10
CMD
12
ADD.
14
DQ
16
Ext.CLK

FIG.70

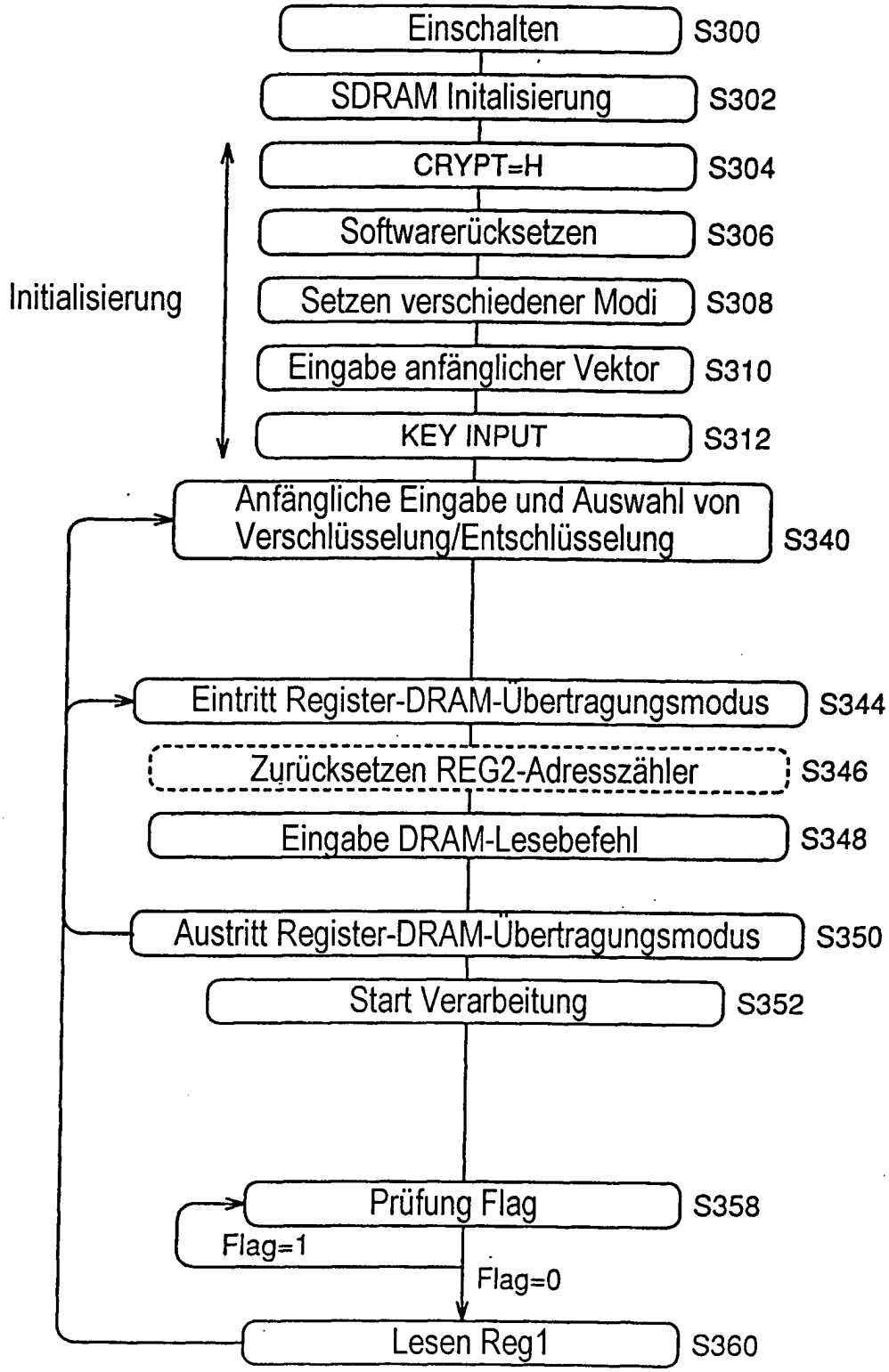

Einschalten
S300
SDRAM Initalisierung
S302
CRYPT=H
S304
Softwarerücksetzen
S306
Initialisierung
Setzen verschiedener Modi
S308
Eingabe anfänglicher Vektor
S310
KEY INPUT
S312
Anfängliche Eingabe und Auswahl von Verschlüsselung/Entschlüsselung
S340
Eintritt Register-DRAM-Übertragungsmodus
S344
Zurücksetzen REG2-Adresszähler
S346
Eingabe DRAM-Lesebefehl
S348
Austritt Register-DRAM-Übertragungsmodus
S350
Start Verarbeitung
S352
Prüfung Flag
S358
Flag=1
Flag=0
Lesen Reg1
S360

FIG.71

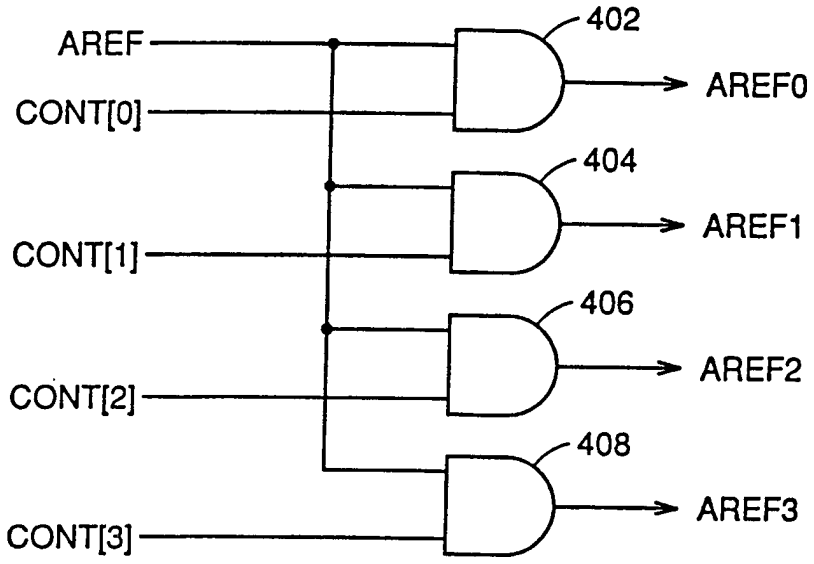

AREF
CONT[0]
CONT[1]
CONT[2]
CONT[3]
402
404
406
408
AREF0
AREF1
AREF2
AREF3

FIG.72

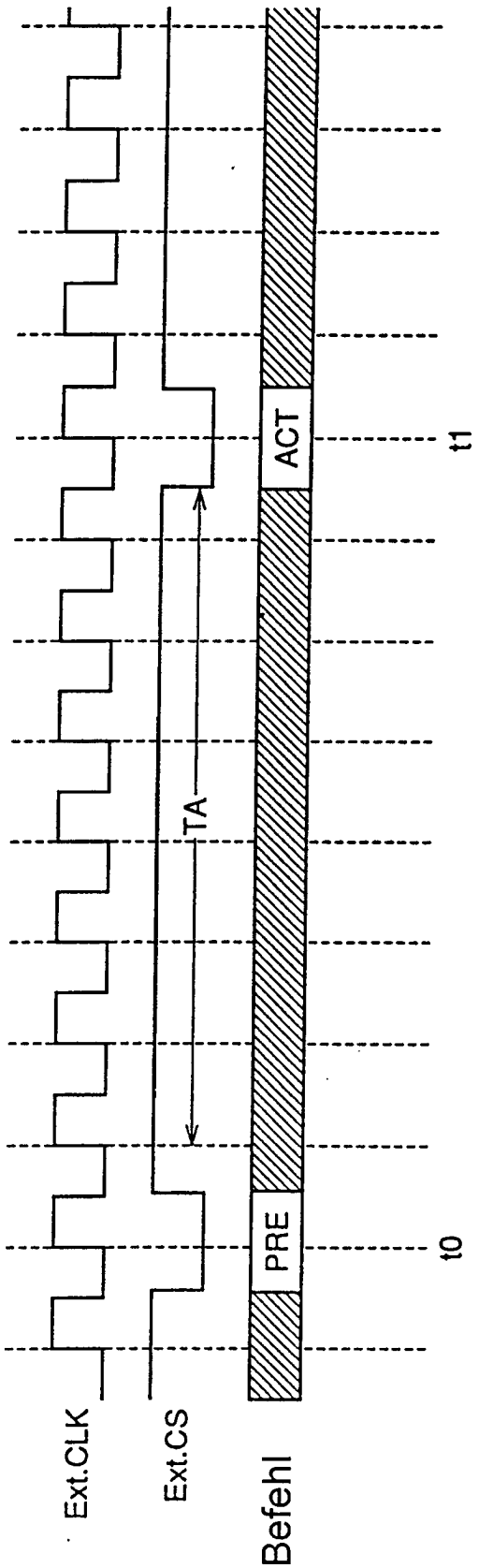
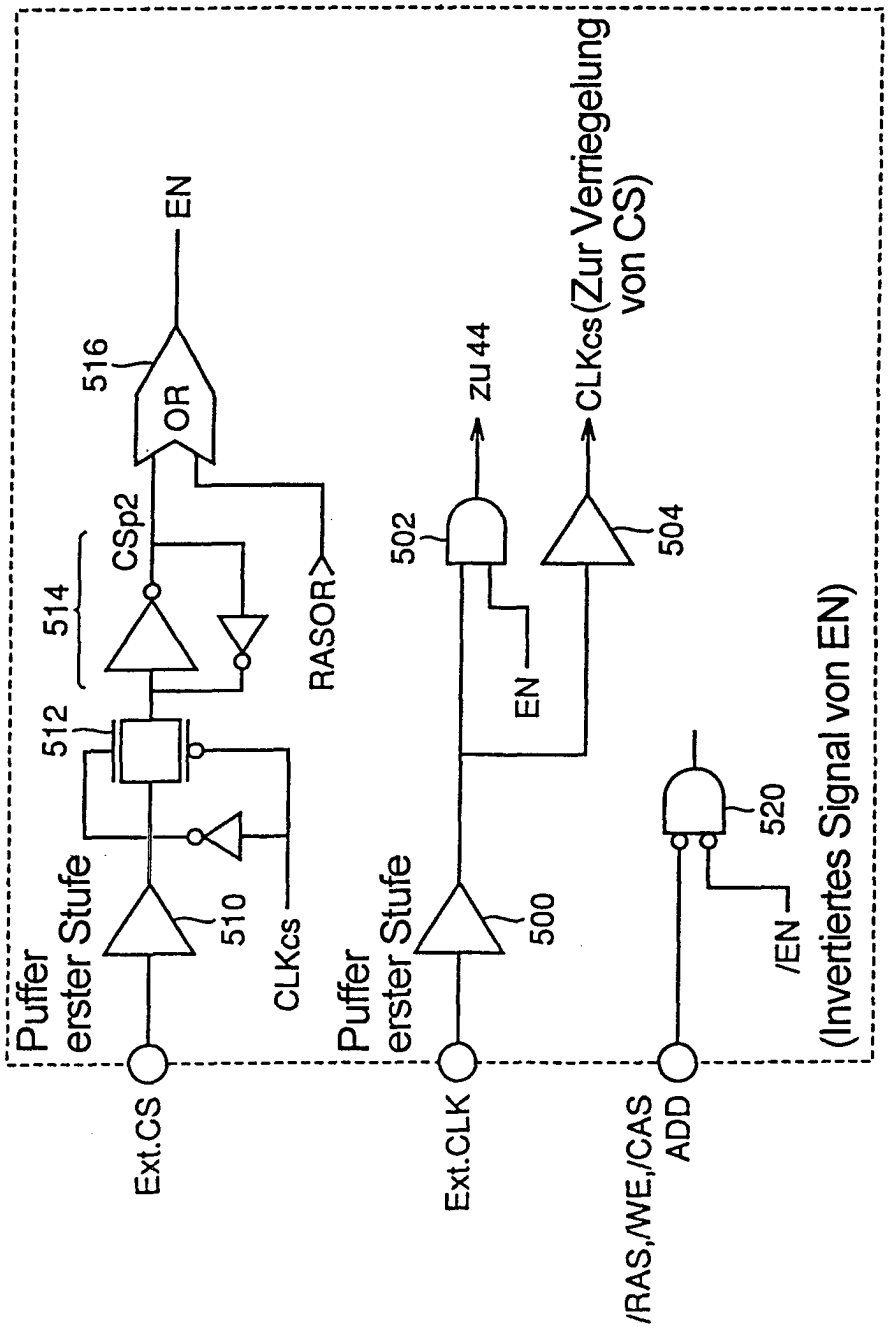


FIG.73


Puffer erster Stufe
510
512
514
CSp2
516
OR
EN
RASOR
CLKcs
Ext.CS
Puffer erster Stufe
500
502
EN
zu 44
CLKcs (Zur Verriegelung von CS)
504
Ext.CLK
520
/EN
(Invertiertes Signal von EN)
/RAS,/WE,/CAS ADD

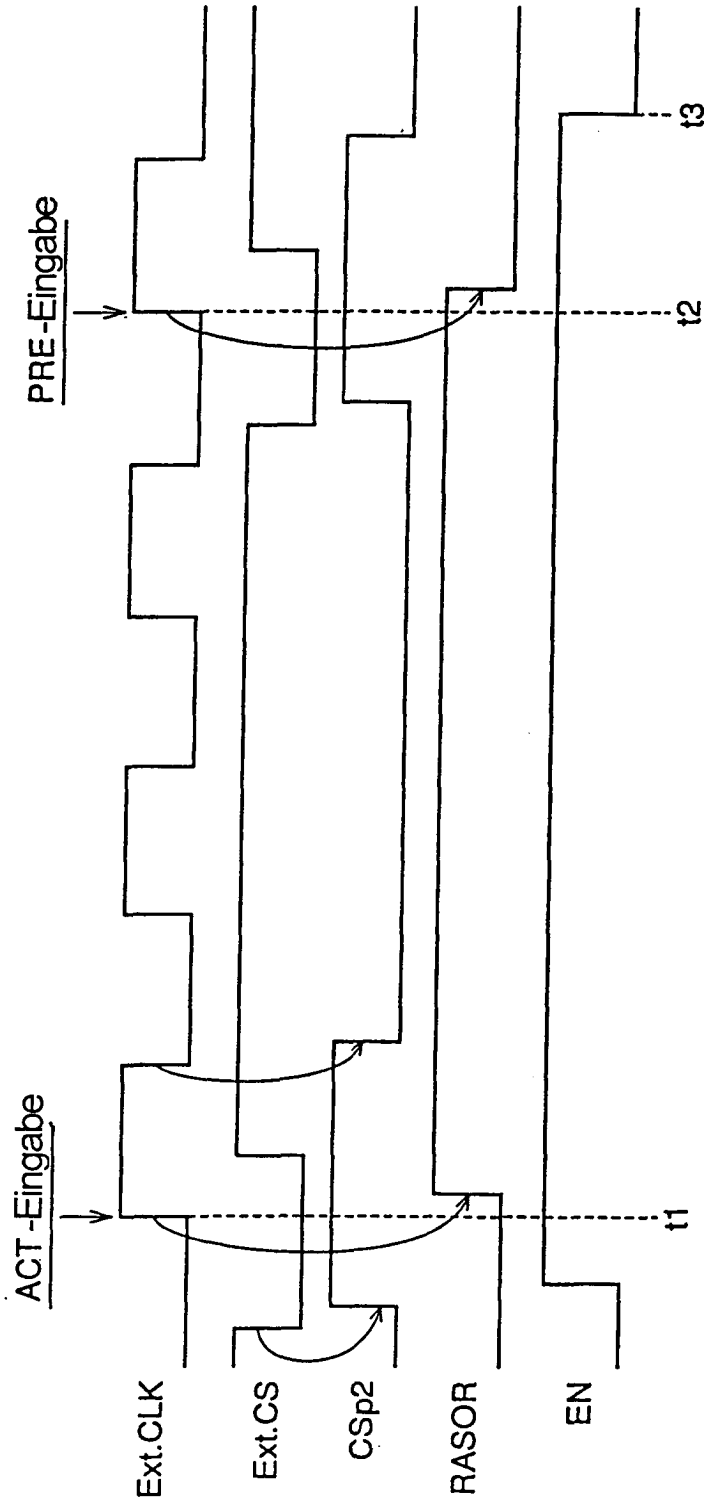
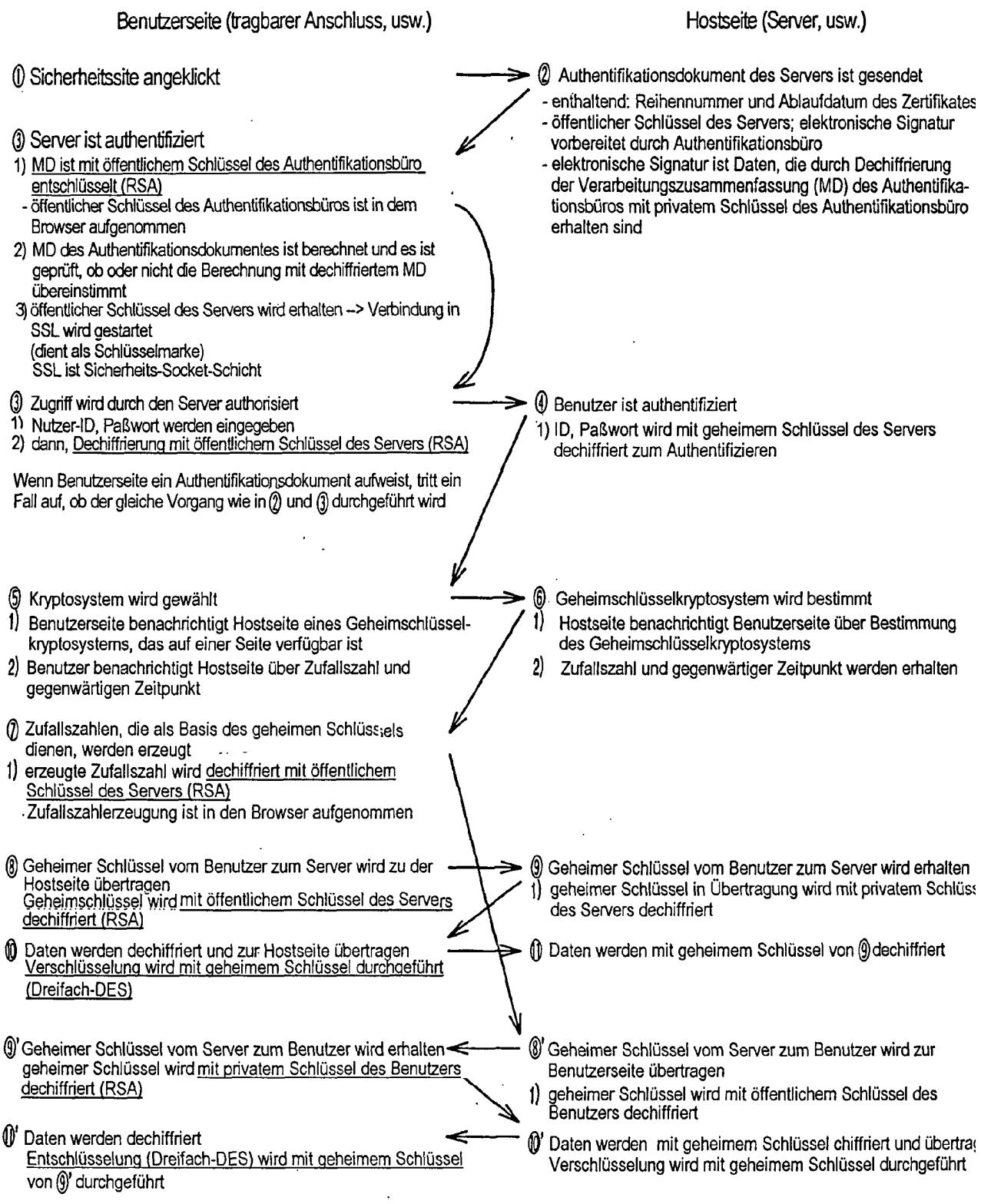


FIG.74

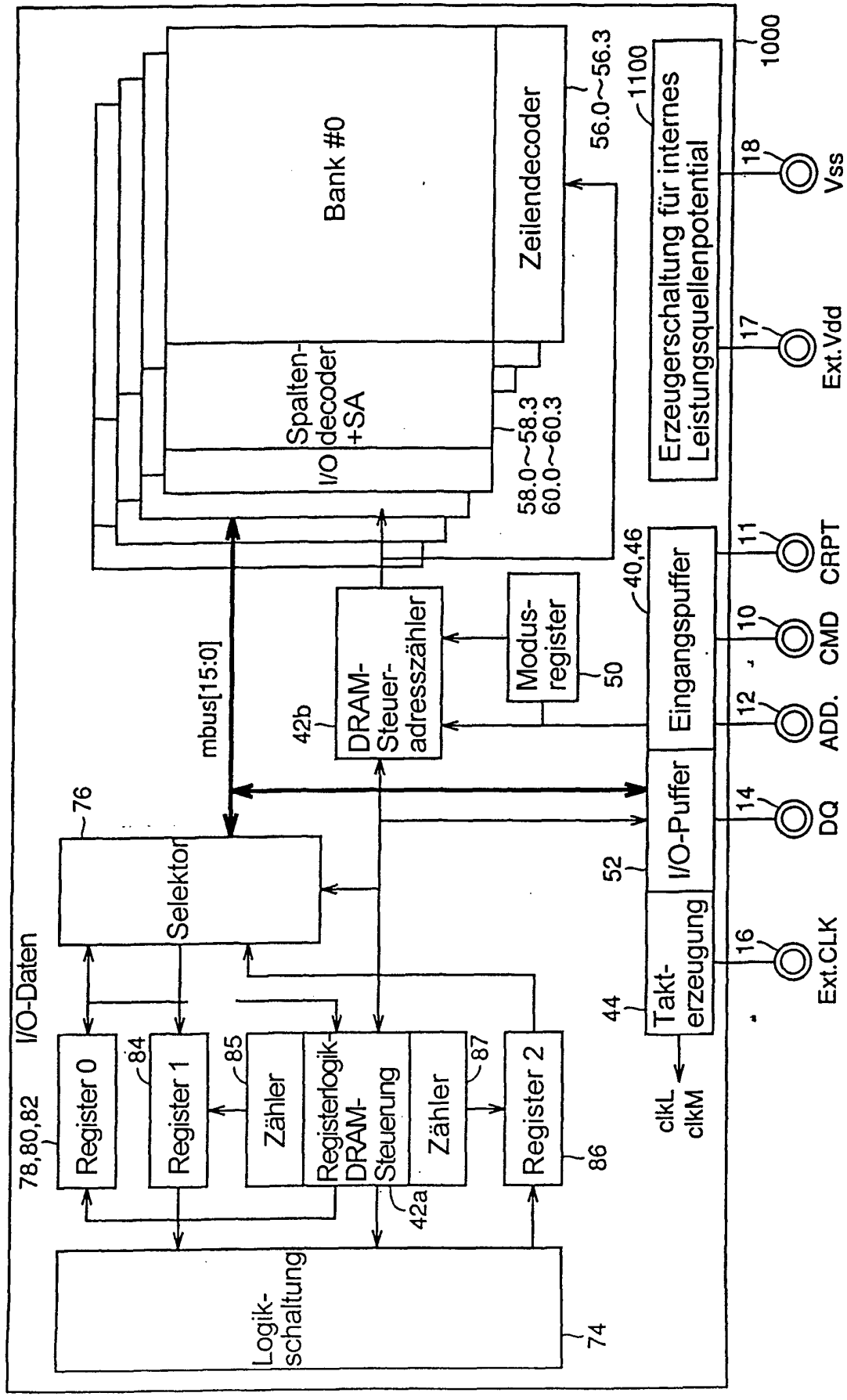
FIG.75

Geheime Datenkommunikation im Internet



Gemeiner Schlüssel wird zu vorgeschriebenen Intervallen während der Kommunikation geändert

FIG.76


1000
Bank #0
Zeilendecoder
56.0~56.3
1100
Erzeugerschaltung für internes Leistungsquellenpotential
18
Vss
17
Ext.Vdd
Spalten-decoder +SA
I/O
58.0~58.3
60.0~60.3
40,46
Eingangspuffer
11
CRPT
10
CMD
12
ADD.
DRAM-Steuer-adresszähler
42b
Modus-register
50
mbus[15:0]
76
Selektor
I/O-Daten
I/O-Puffer
14
DQ
52
Takt-erzeugung
16
Ext.CLK
44
clkL
clkM
78,80,82
Register 0
84
Register 1
85
Zähler
Registerlogik-DRAM-Steuerung
42a
Zähler
87
Register 2
86
Logik-schaltung
74

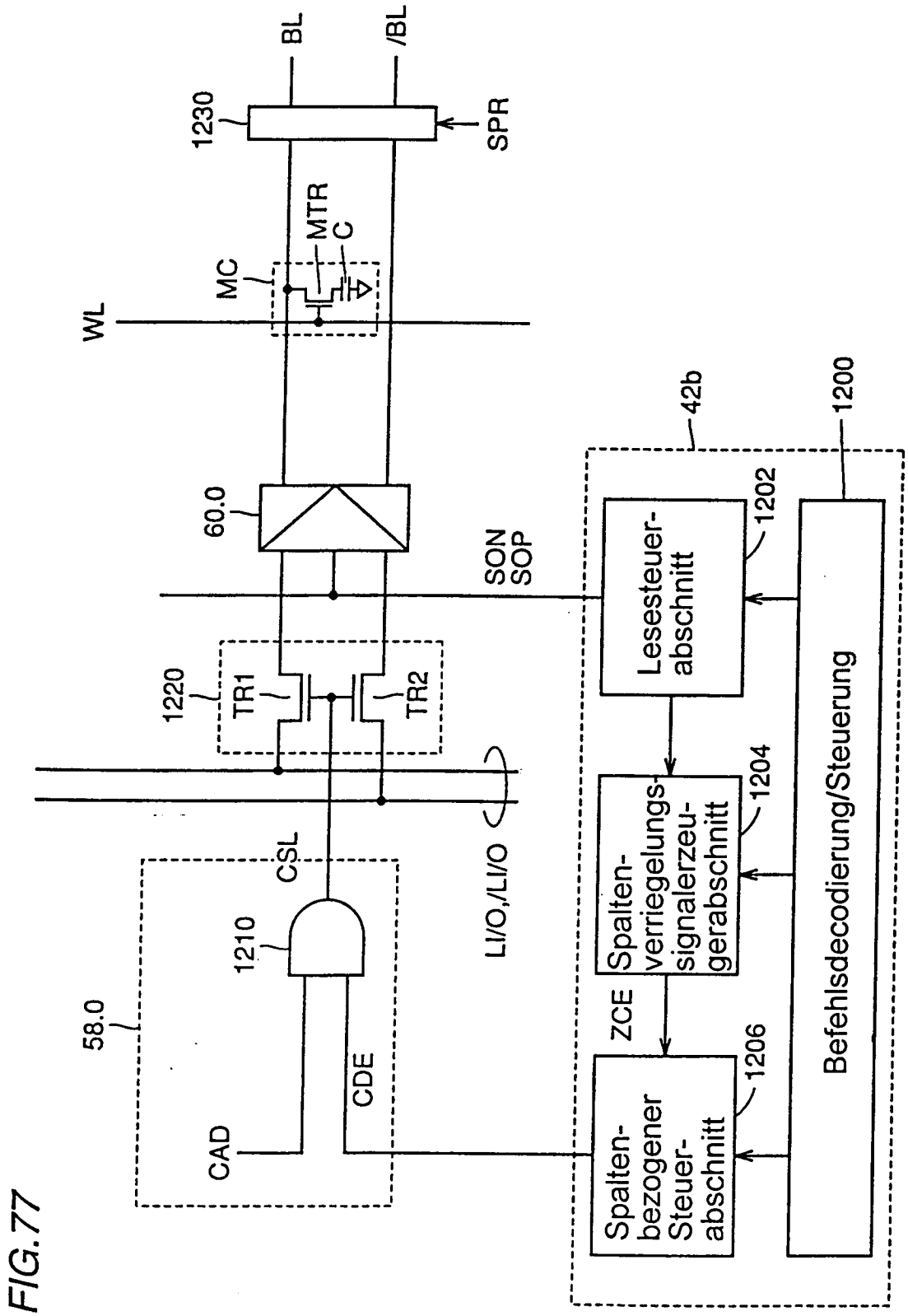

BL
/BL
1230
SPR
MTR
C
MC
WL
60.0
SON
SOP
1220
TR1
TR2
CSL
1210
LI/O,/LI/O
58.0
CAD
CDE
42b
1200
1202
Lesesteuer-abschnitt
1204
Spalten-verriegelungs-signalerzeu-gerabschnitt
ZCE
1206
Spalten-bezogener Steuer-abschnitt
Befehlsdecodierung/Steuerung

FIG.77

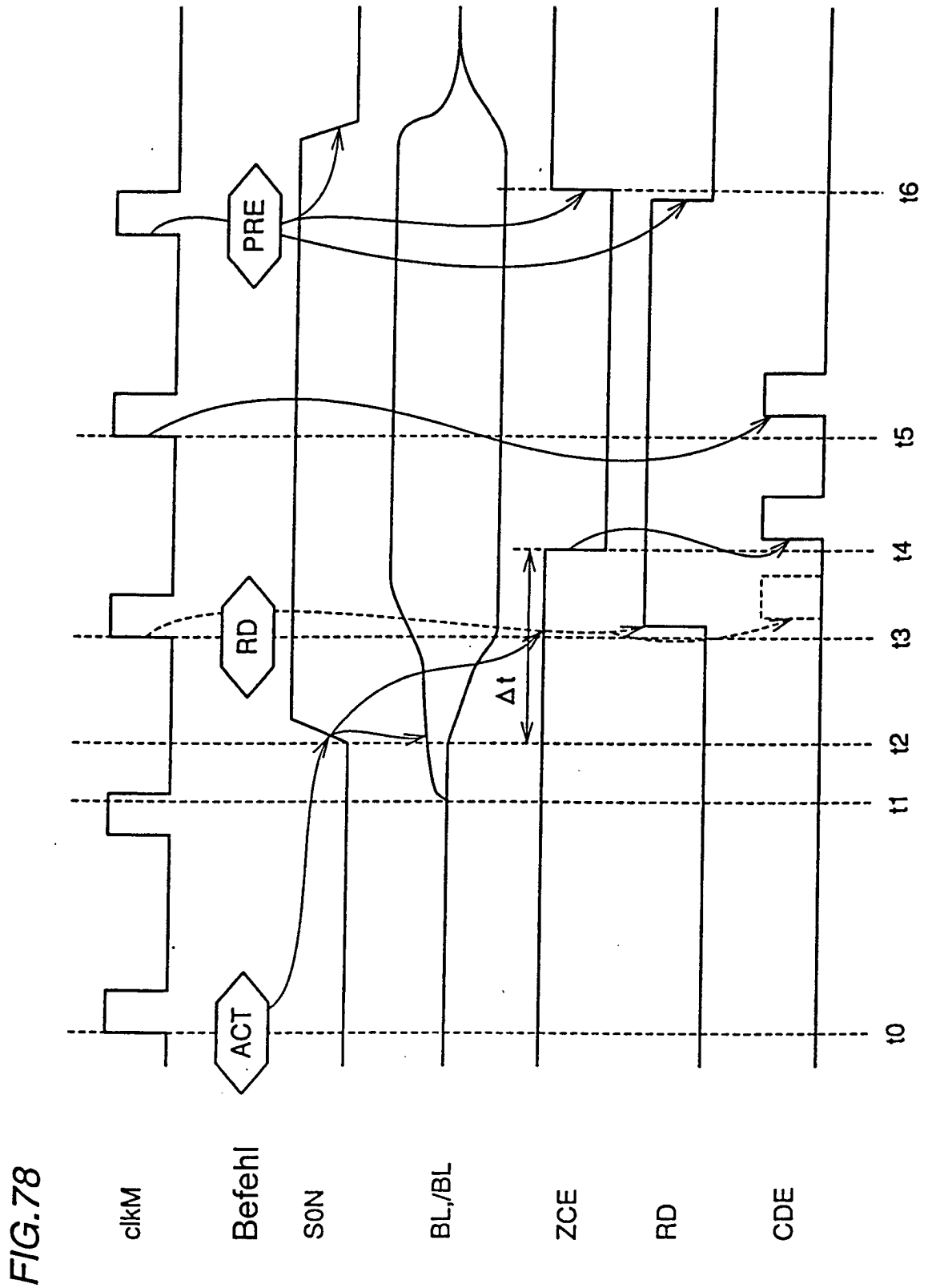
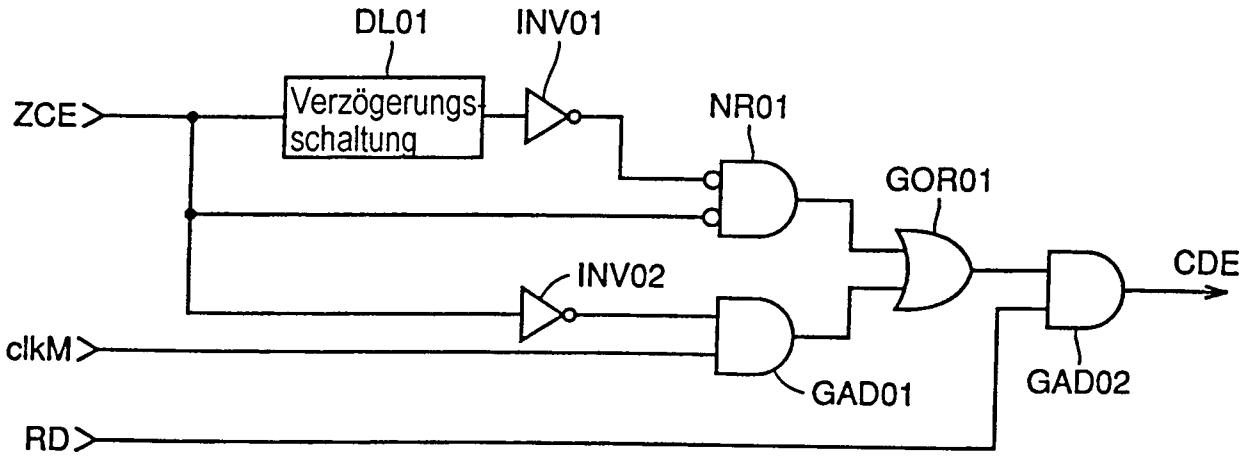


FIG.78

FIG.79


DL01
INV01
ZCE
Verzögerungs-
schaltung
NR01
GOR01
CDE
INV02
clkM
GAD01
GAD02
RD

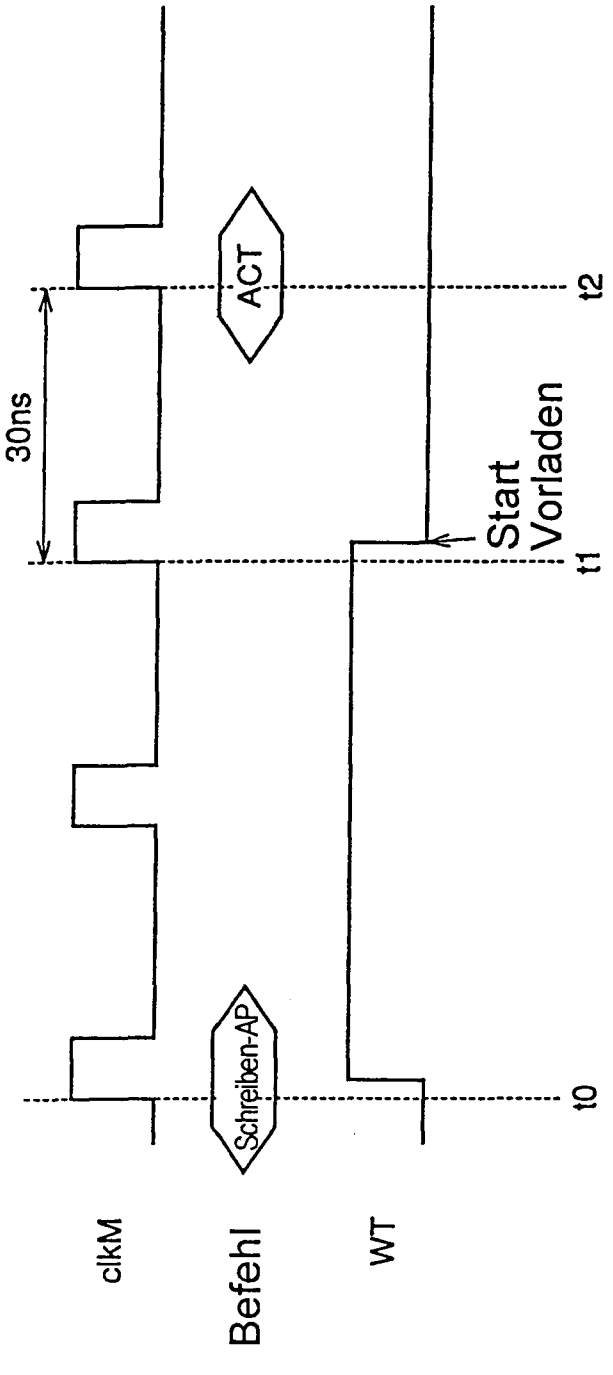
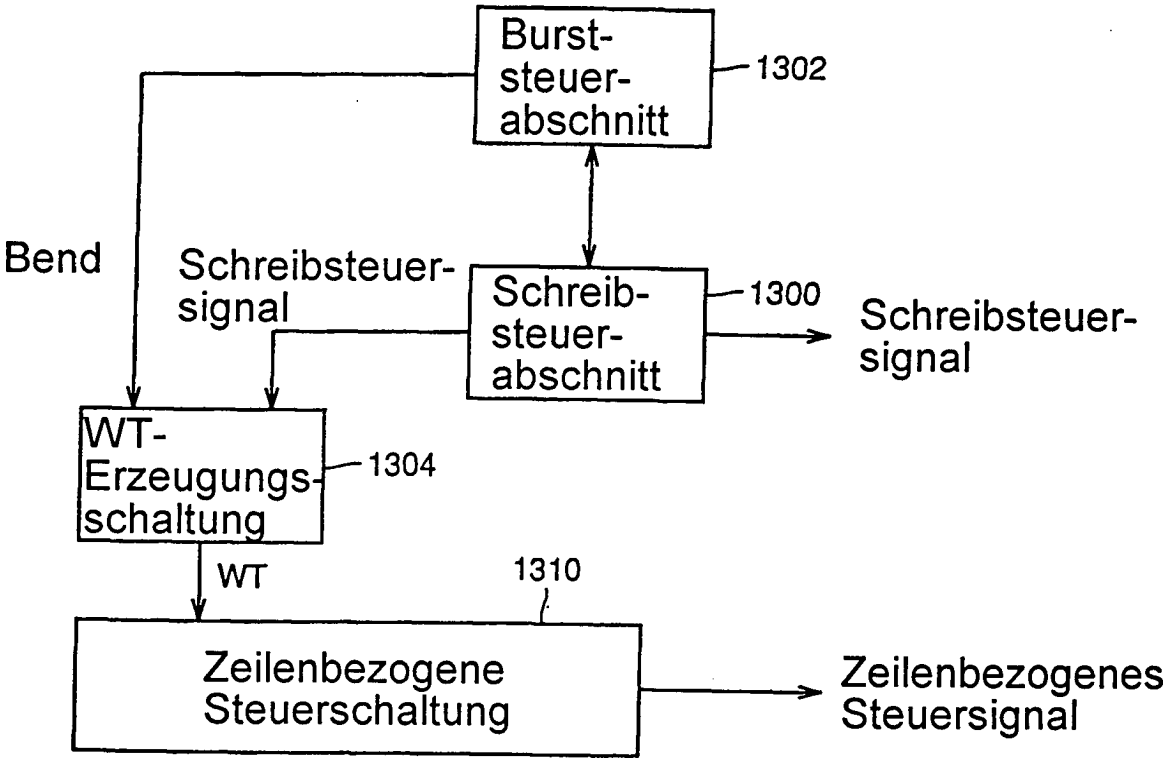


FIG.80

FIG.81


Burst-
steuer-
abschnitt
1302
Bend
Schreibsteuer-
signal
Schreib-
steuer-
abschnitt
1300
Schreibsteuer-
signal
WT-
Erzeugungs-
schaltung
1304
WT
1310
Zeilenbezogene
Steuerschaltung
Zeilenbezogenes
Steuersignal

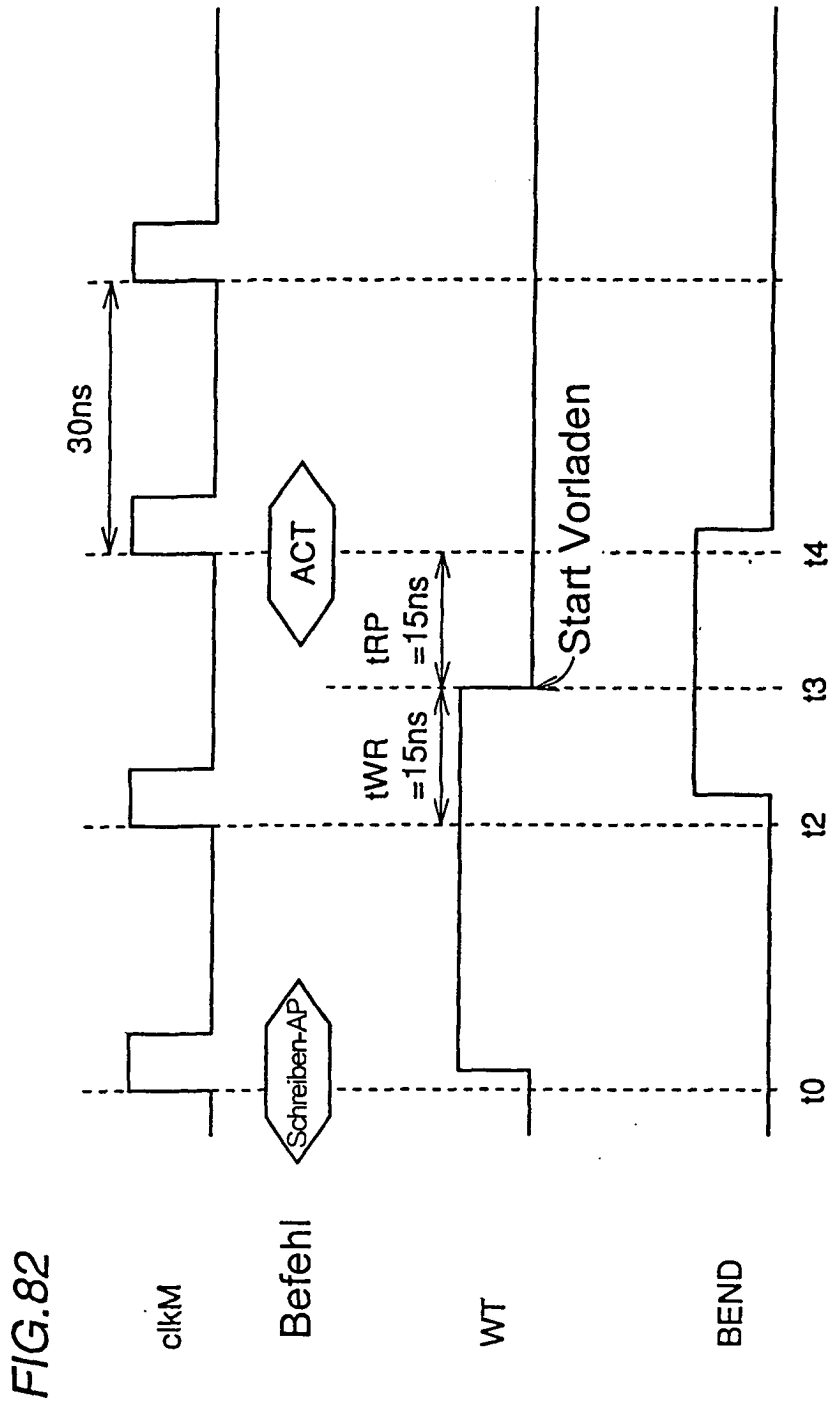
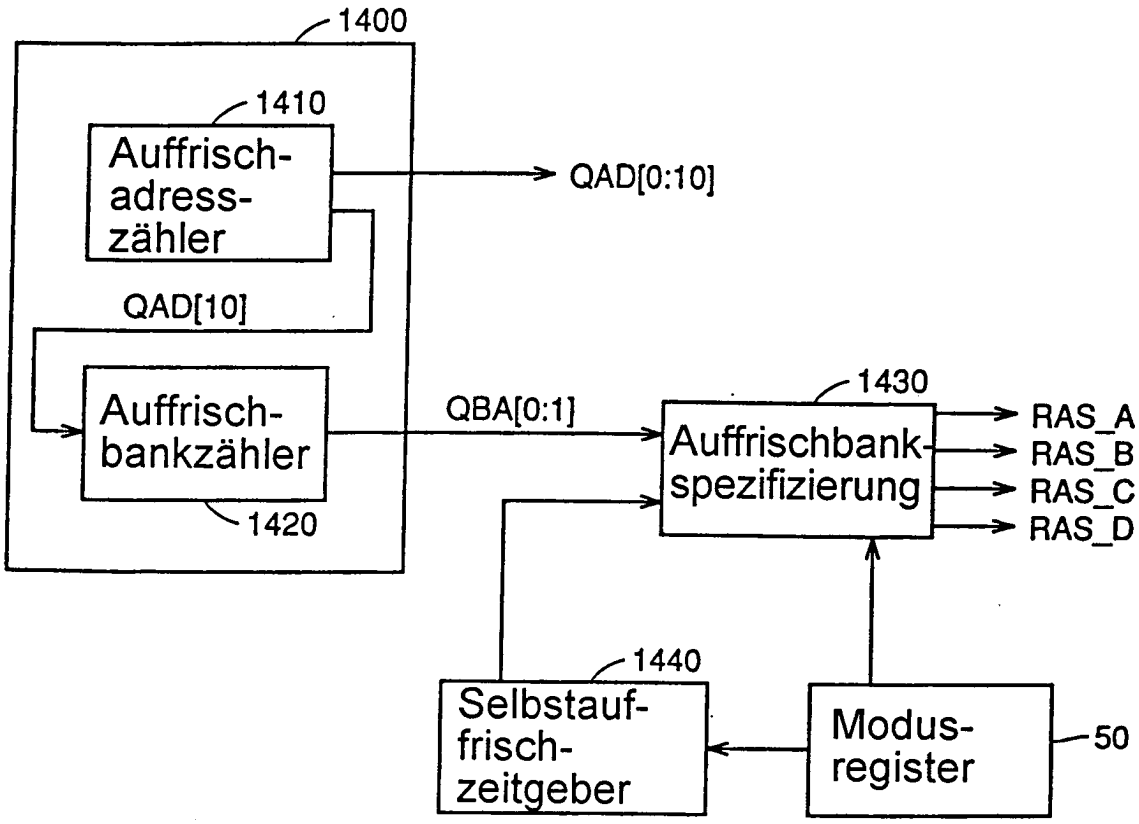

clkM
Befehl
WT
BEND
t0
t2
t3
t4
30ns
ACT
Schreiben-AP
tRP
=15ns
tWR
=15ns
Start Vorladen

FIG.82

FIG.83


1400
1410
Auffrisch-
adress-
zähler
QAD[0:10]
QAD[10]
Auffrisch-
bankzähler
1420
QBA[0:1]
1430
Auffrischbank-
spezifizierung
RAS_A
RAS_B
RAS_C
RAS_D
1440
Selbstauf-
frisch-
zeitgeber
Modus-
register
50

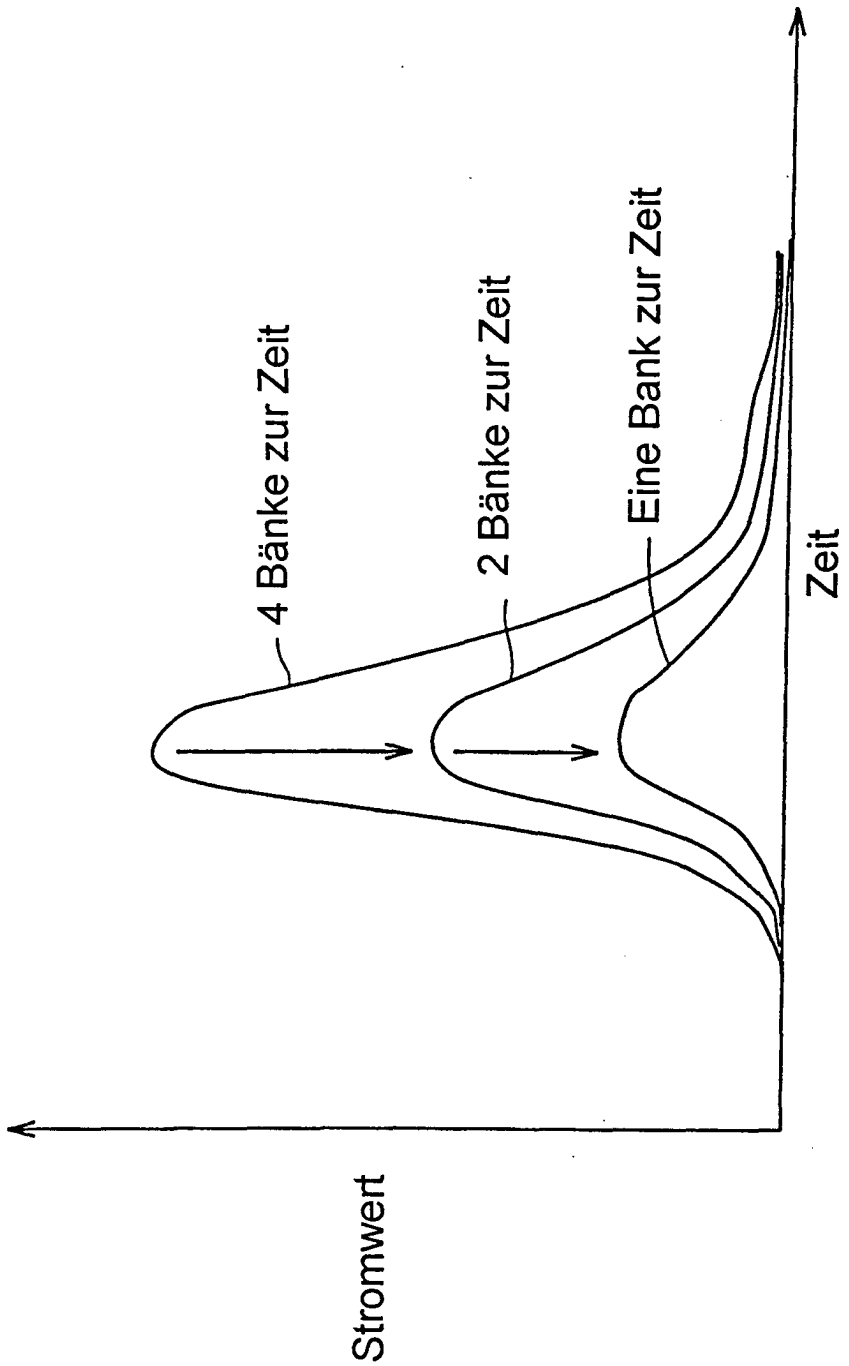

4 Bänke zur Zeit
2 Bänke zur Zeit
Eine Bank zur Zeit
Zeit
Stromwert

FIG.84

FIG.85

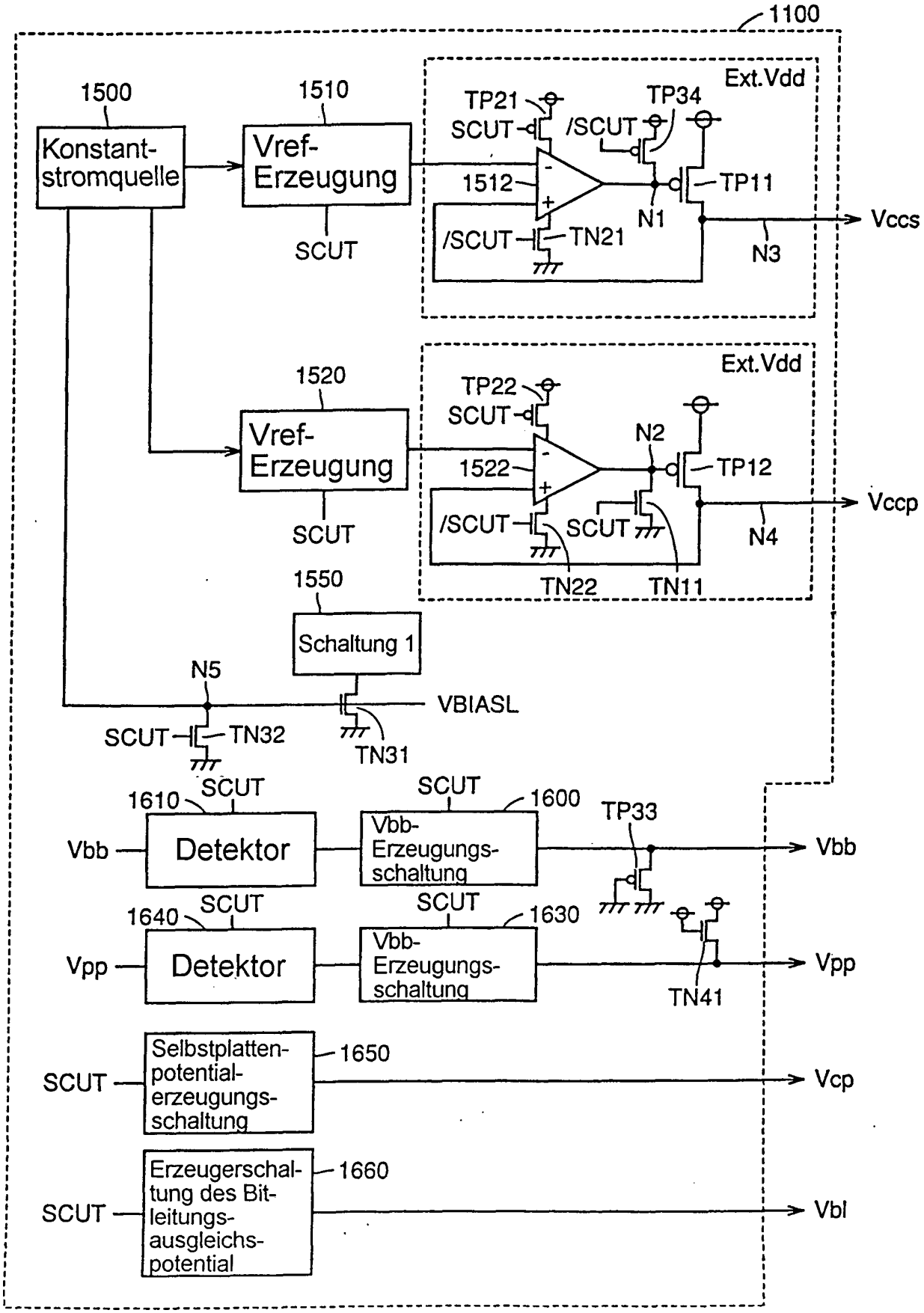

1100
1500
Konstant-stromquelle
1510
Vref-Erzeugung
SCUT
Ext.Vdd
TP21
SCUT
/SCUT
TP34
1512
TP11
N1
/SCUT
TN21
N3
Vccs
1520
Vref-Erzeugung
SCUT
Ext.Vdd
TP22
SCUT
N2
1522
TP12
/SCUT
SCUT
N4
Vccp
TN22
TN11
1550
Schaltung 1
N5
VBIASL
SCUT
TN32
TN31
1610
SCUT
Vbb
Detektor
SCUT
1600
Vbb-Erzeugungs-schaltung
TP33
Vbb
1640
SCUT
Vpp
Detektor
SCUT
1630
Vbb-Erzeugungs-schaltung
Vpp
TN41
SCUT
Selbstplatten-potential-erzeugungs-schaltung
1650
Vcp
SCUT
Erzeugerschal-tung des Bit-leitungs-ausgleichs-potential
1660
Vbl

FIG.86

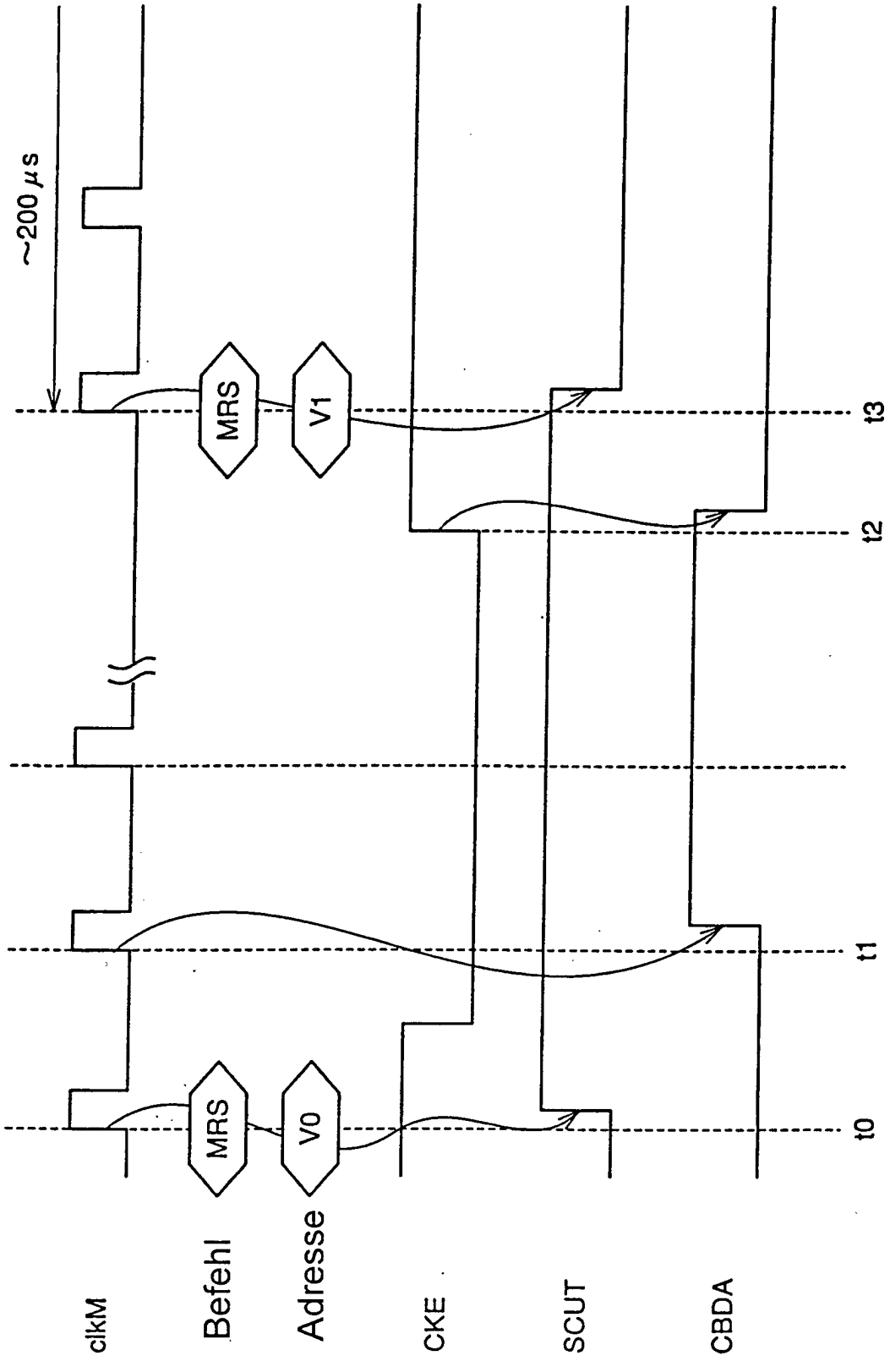

~200 µs
MRS
V1
MRS
V0
t3
t2
t1
t0
clkM
Befehl
Adresse
CKE
SCUT
CBDA

FIG.87

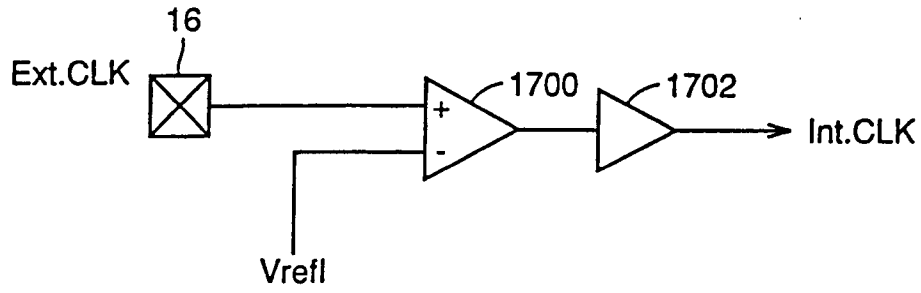

16
Ext.CLK
1700
1702
Int.CLK
Vrefl

FIG.88

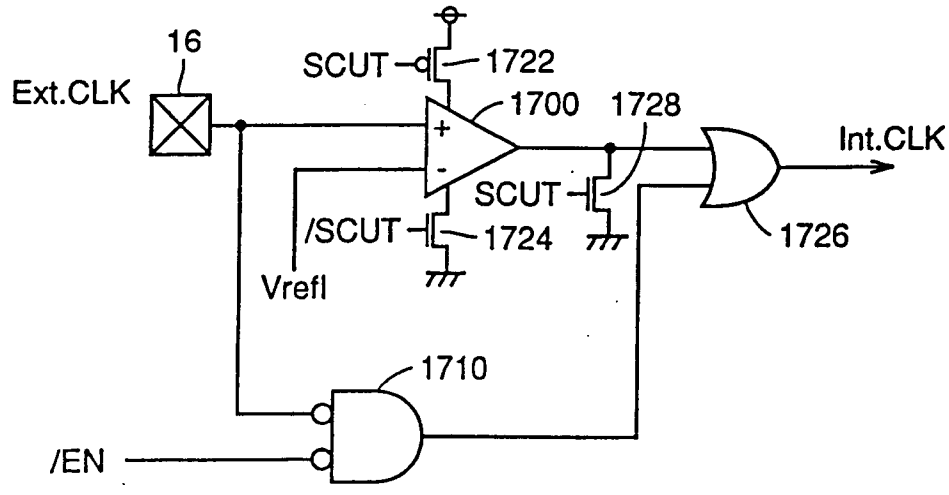

16
Ext.CLK
SCUT
1722
1700
1728
Int.CLK
SCUT
/SCUT
1724
1726
Vrefl
1710
/EN

FIG.89

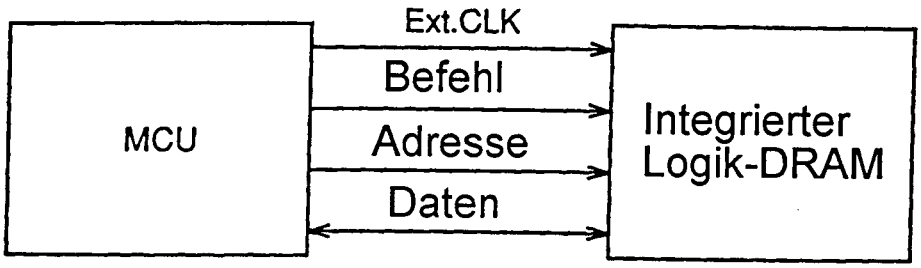

Ext.CLK
Befehl
Adresse
Daten
MCU
Integrierter
Logik-DRAM

FIG.90

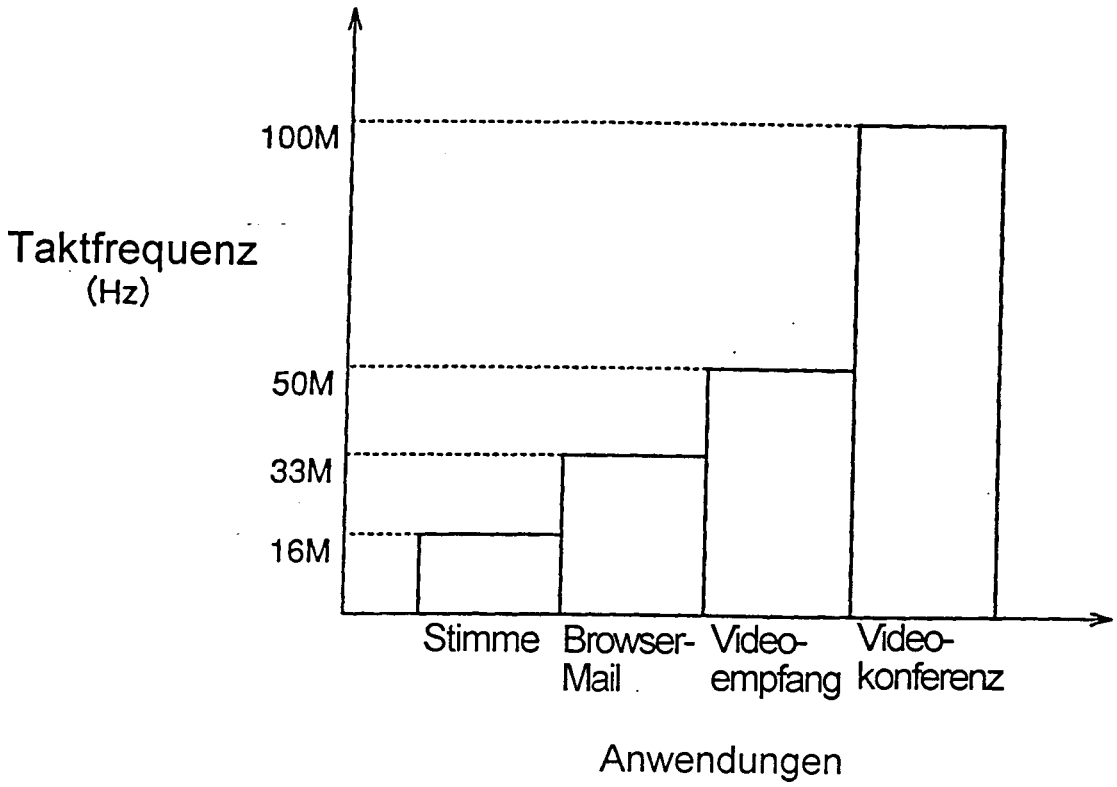

100M
50M
33M
16M
Taktfrequenz
(Hz)
Stimme
Browser-
Mail
Video-
empfang
Video-
konferenz
Anwendungen

FIG.91

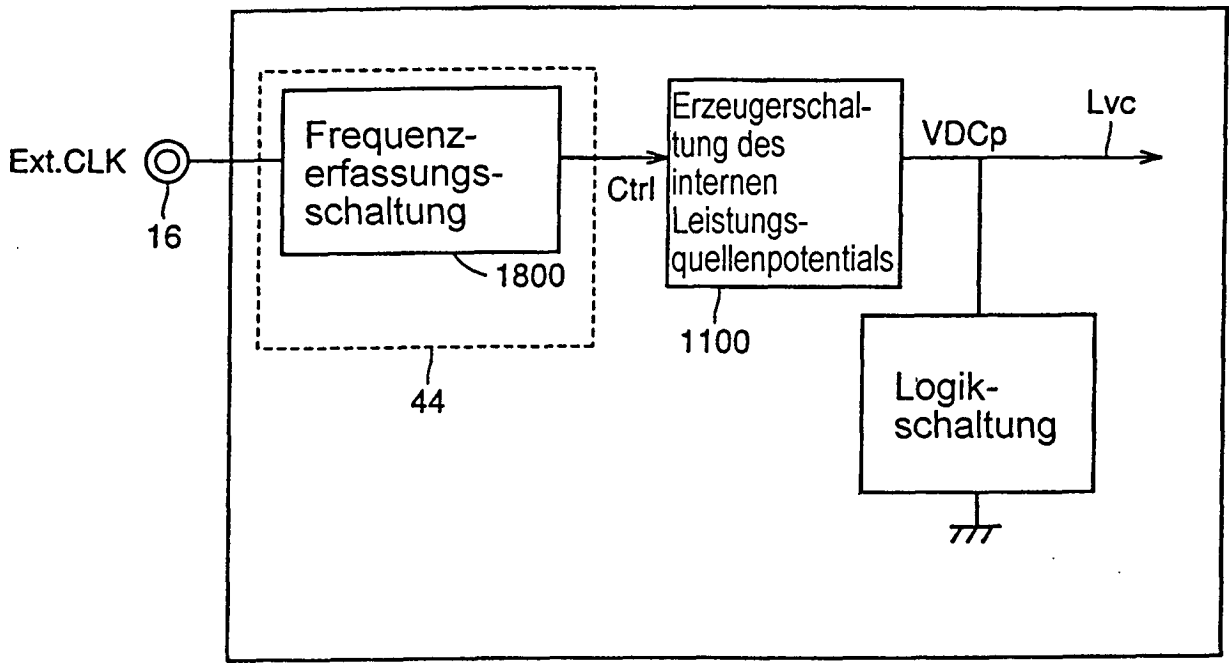
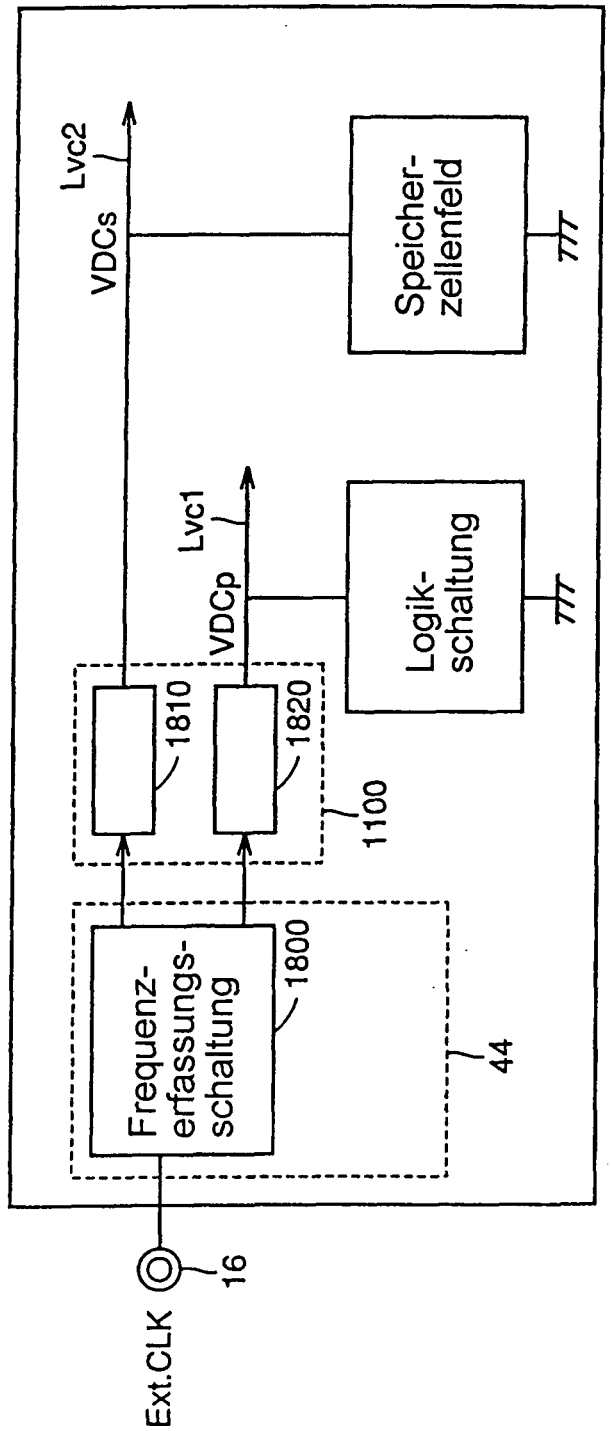

Ext.CLK
16
Frequenz-
erfassungs-
schaltung
1800
44
Ctrl
Erzeugerschal-
tung des
internen
Leistungs-
quellenpotentials
1100
VDCp
Lvc
Logik-
schaltung

FIG.92


Ext.CLK
16
Frequenz-erfassungs-schaltung
1800
44
1810
1820
1100
VDCp
Lvc1
VDCs
Lvc2
Logik-schaltung
Speicher-zellenfeld

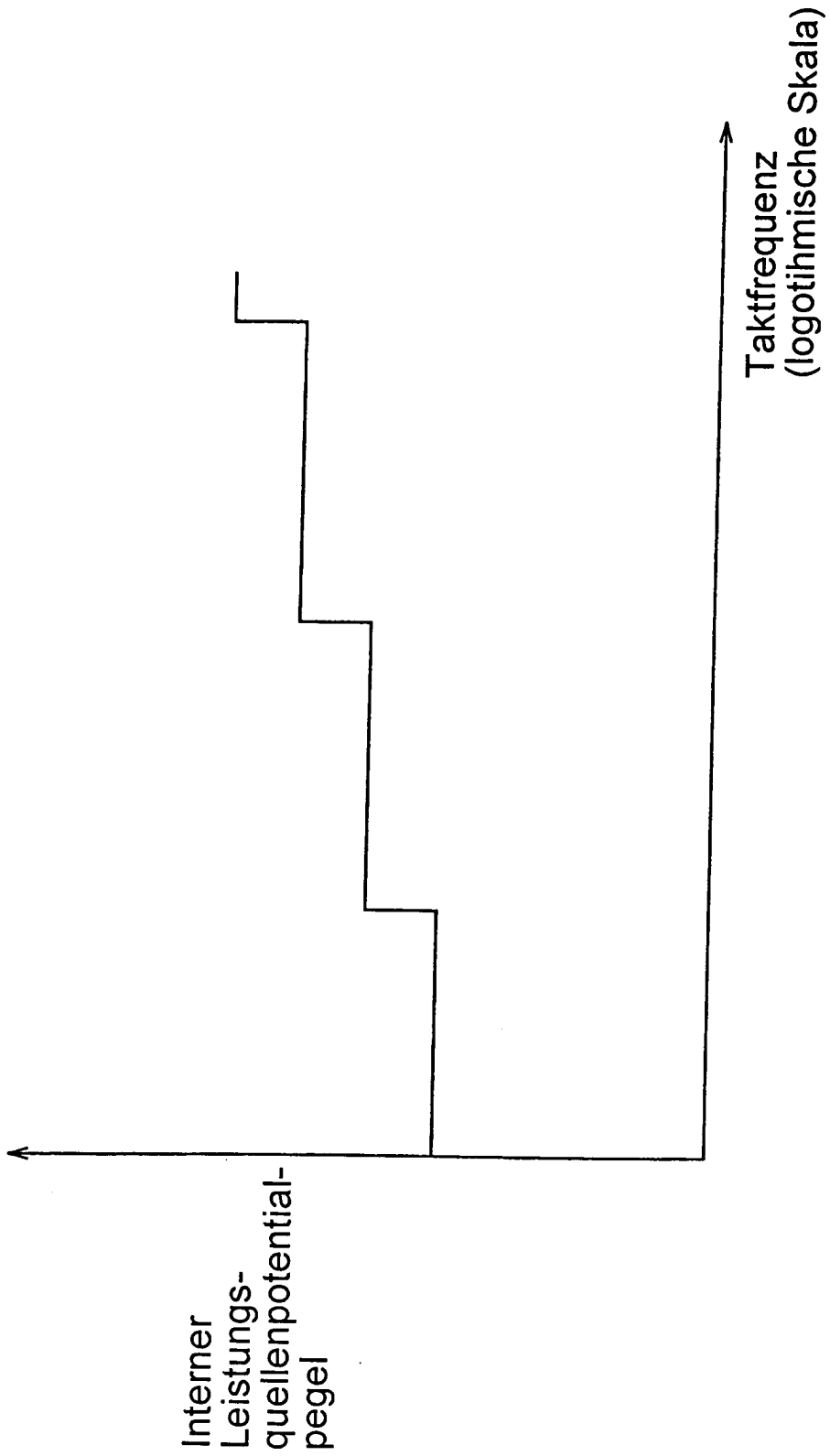
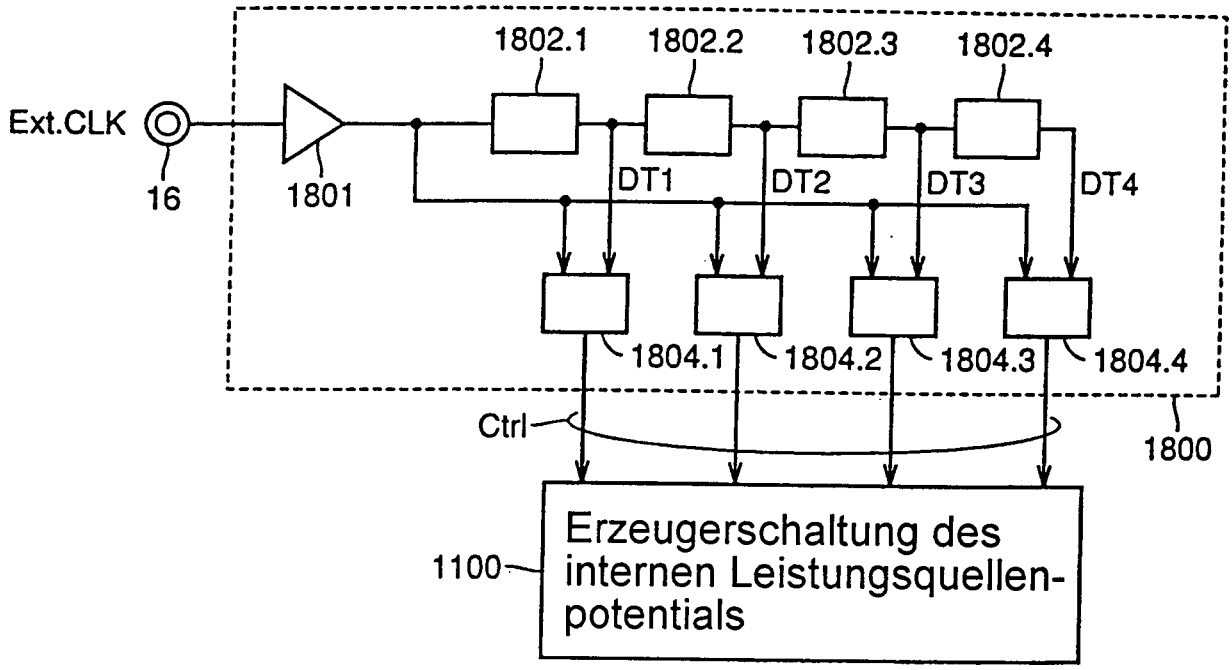

Taktfrequenz
(logothmische Skala)
Interner
Leistungs-
quellenpotential-
pegel

FIG.93

FIG.94


1802.1
1802.2
1802.3
1802.4
Ext.CLK
16
1801
DT1
DT2
DT3
DT4
1804.1
1804.2
1804.3
1804.4
Ctrl
1800
1100
Erzeugerschaltung des internen Leistungsquellen-potentials

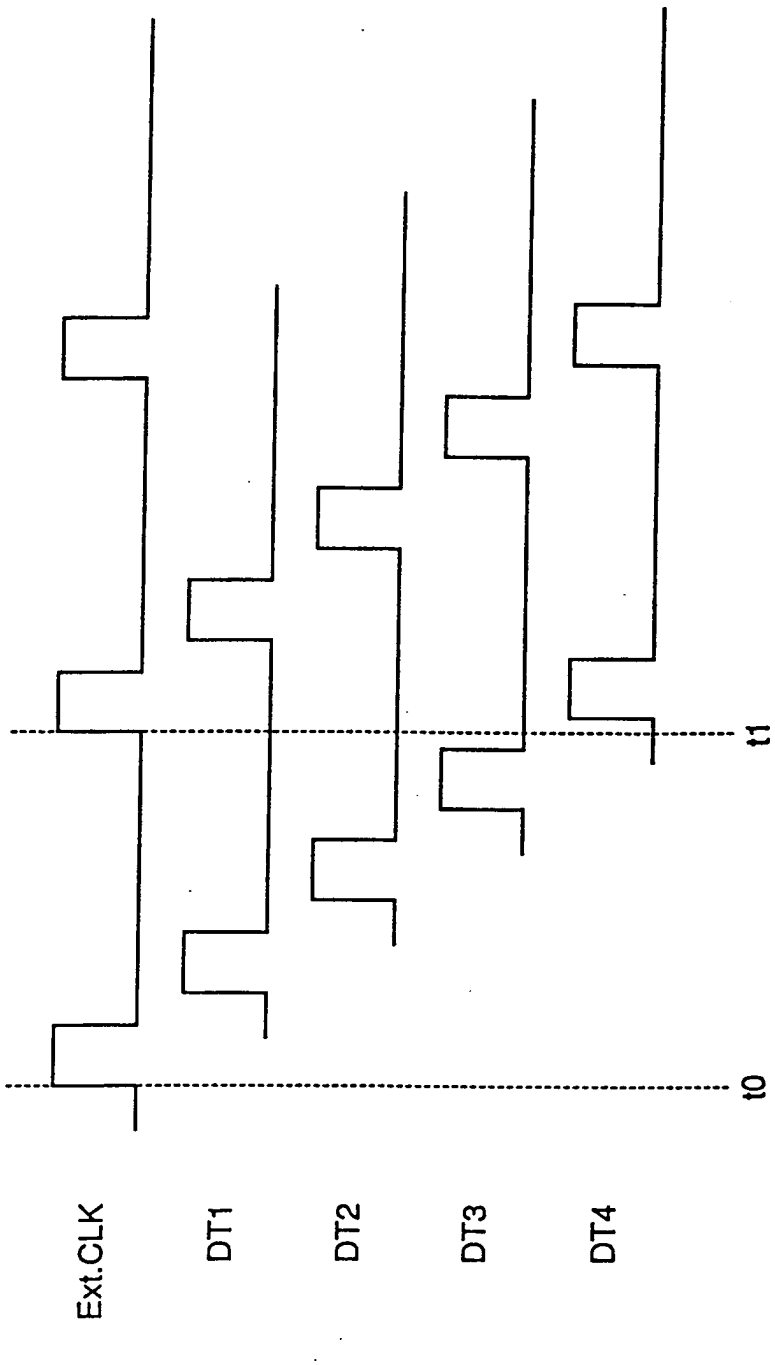


FIG.95

FIG.96

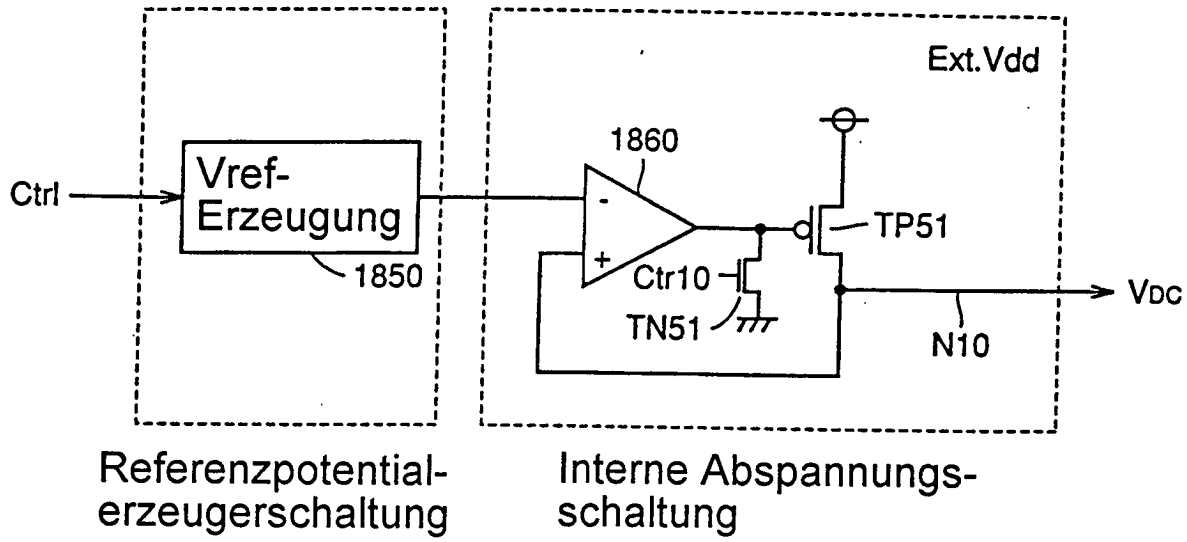

Ctrl
Vref-
Erzeugung
1850
1860
Ext.Vdd
TP51
Ctr10
TN51
VDC
N10
Referenzpotential-
erzeugerschaltung
Interne Abspannungs-
schaltung

FIG.97

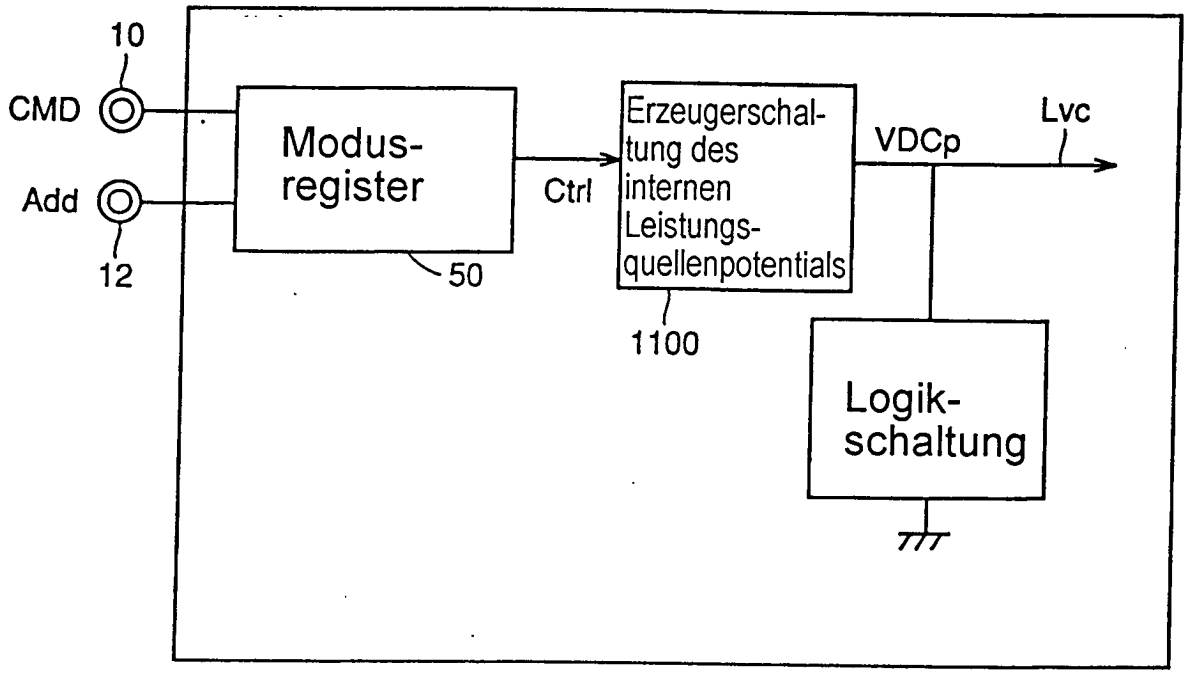

10
CMD
Add
12
Modus-
register
50
Ctrl
Erzeugerschal-
tung des
internen
Leistungs-
quellenpotentials
1100
VDCp
Lvc
Logik-
schaltung

FIG.98

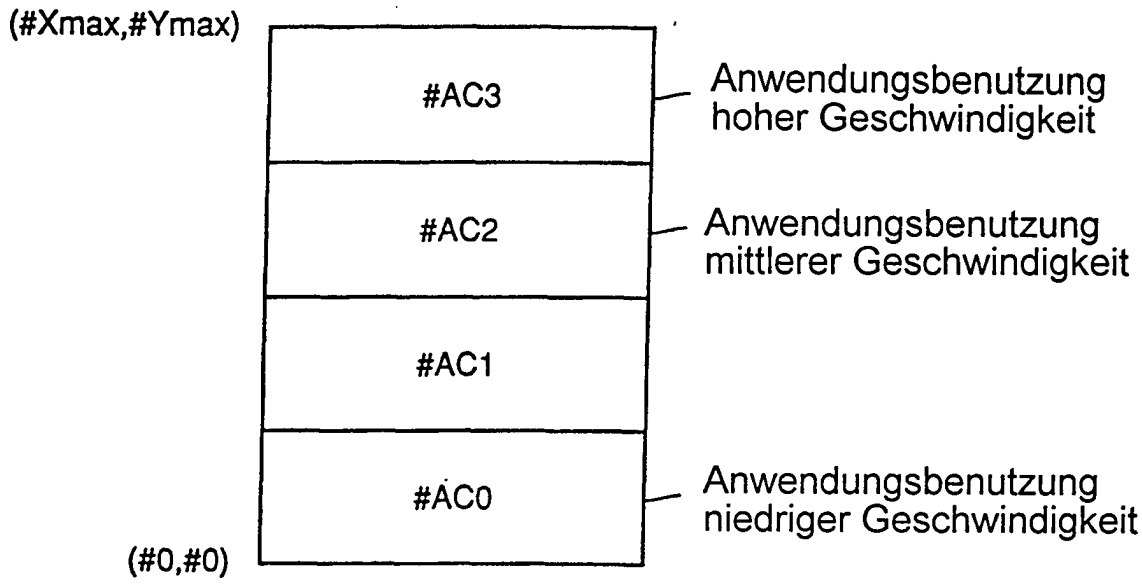

(#Xmax,#Ymax)
#AC3
#AC2
#AC1
#AC0
(#0,#0)
Anwendungsbenutzung hoher Geschwindigkeit
Anwendungsbenutzung mittlerer Geschwindigkeit
Anwendungsbenutzung niedriger Geschwindigkeit

FIG.99

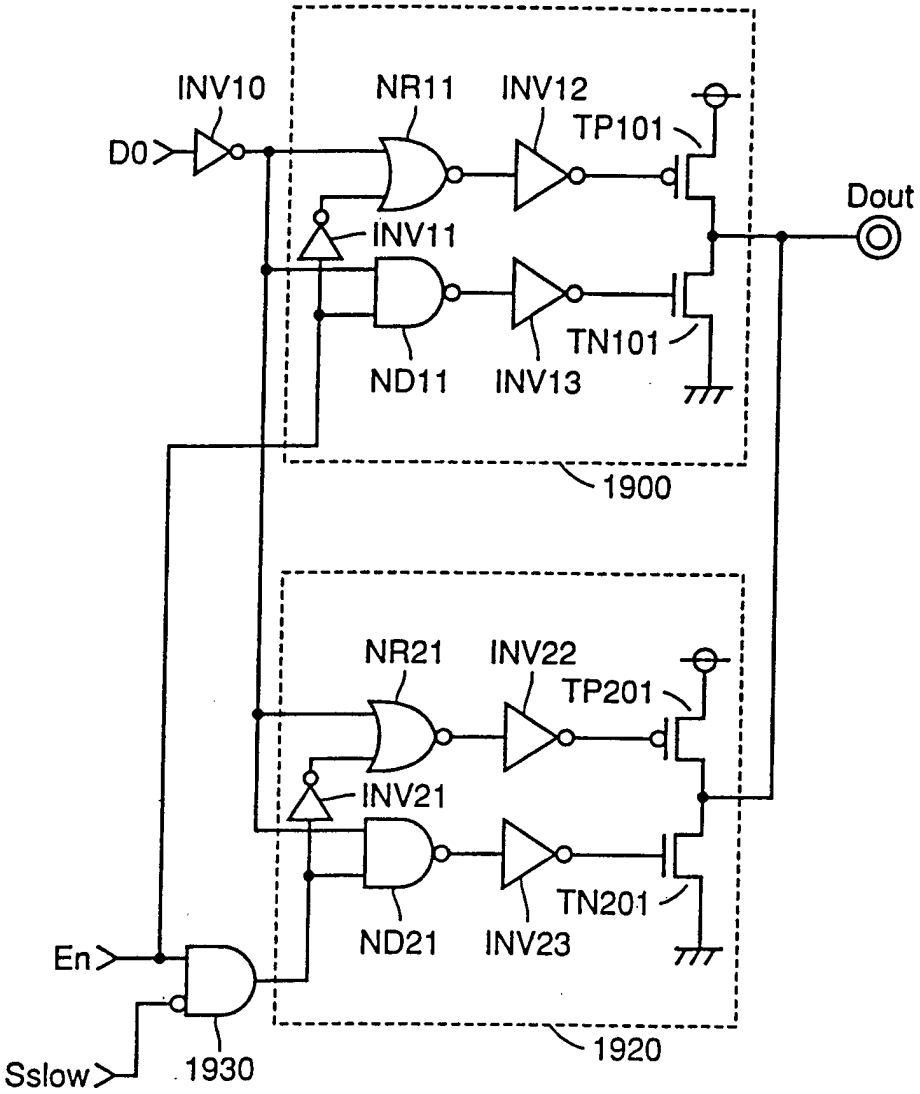

INV10
NR11
INV12
TP101
D0
Dout
INV11
TN101
ND11
INV13
1900
NR21
INV22
TP201
INV21
TN201
ND21
INV23
En
Sslow
1930
1920

FIG.100

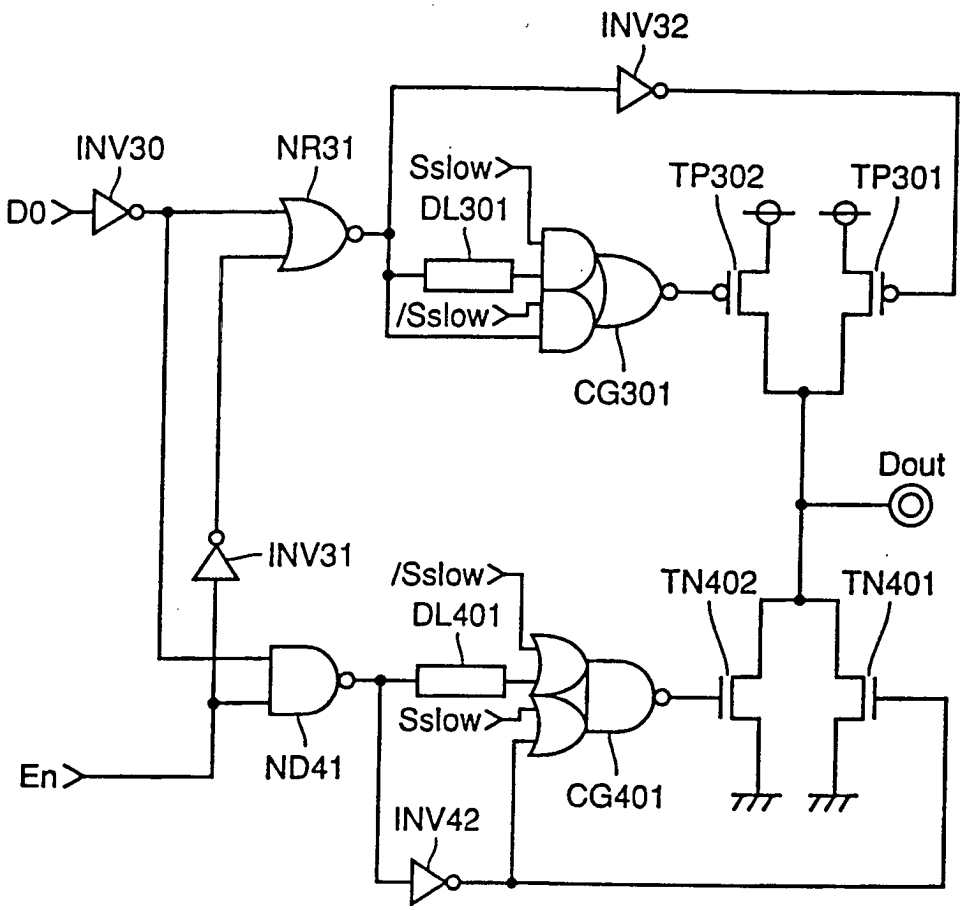

INV32
INV30
NR31
Sslow
DL301
/Sslow
TP302
TP301
D0
CG301
Dout
INV31
/Sslow
DL401
TN402
TN401
Sslow
En
ND41
CG401
INV42

FIG.101

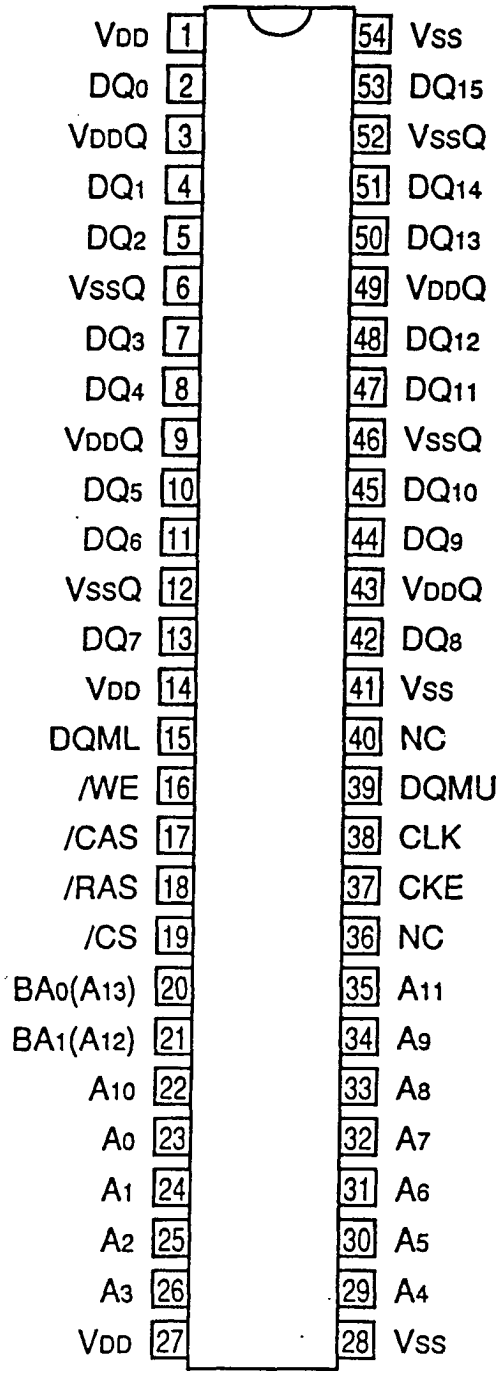

VDD 1
DQ0 2
VDDQ 3
DQ1 4
DQ2 5
VSSQ 6
DQ3 7
DQ4 8
VDDQ 9
DQ5 10
DQ6 11
VSSQ 12
DQ7 13
VDD 14
DQML 15
/WE 16
/CAS 17
/RAS 18
/CS 19
BA0(A13) 20
BA1(A12) 21
A10 22
A0 23
A1 24
A2 25
A3 26
VDD 27
54 VSS
53 DQ15
52 VSSQ
51 DQ14
50 DQ13
49 VDDQ
48 DQ12
47 DQ11
46 VSSQ
45 DQ10
44 DQ9
43 VDDQ
42 DQ8
41 VSS
40 NC
39 DQMU
38 CLK
37 CKE
36 NC
35 A11
34 A9
33 A8
32 A7
31 A6
30 A5
29 A4
28 VSS

FIG.102

Anschlussnamen	Funktion
CLK	Mastertakt
CKE	Taktfreigabe
/CS	Chipauswahl
/RAS	Zeilenadressstrobe
/CAS	Spaltenadressstrobe
/WE	Schreibfreigabe
$DQ_{0\sim15}$	Daten-I/O
DQM(U/L)	Ausgangssperre/Schreibmaskierung
$A_{0\sim11}$	Adresseingabe
$BA_{0,1}(A_{12,13})$	Bankadresse
V_{DD}	Leistungsquelle
$V_{DD}Q$	Ausgangsleistungsquelle
V_{SS}	Masse
$V_{SS}Q$	Ausgangsmasse

FIG.103