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

(10) **DE 603 08 076 T2** 2007.04.19

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 347 459 B1**
(21) Deutsches Aktenzeichen: **603 08 076.6**
(96) Europäisches Aktenzeichen: **03 006 070.1**
(96) Europäischer Anmeldetag: **19.03.2003**
(97) Erstveröffentlichung durch das EPA: **24.09.2003**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **06.09.2006**
(47) Veröffentlichungstag im Patentblatt: **19.04.2007**

(51) Int Cl.[8]: ***G11C 29/06*** (2006.01)
G11C 29/10 (2006.01)

(30) Unionspriorität:
101241 **19.03.2002** **US**

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Broadcom Corp., Irvine, Calif., US

(74) Vertreter:
Bosch, Graf von Stosch, Jehle Patentanwaltsgesellschaft mbH, 80639 München

(72) Erfinder:
Winograd, Gil I., Aliso Viejo, California 92656, US;
Terzioglu, Esin, Aliso Viejo, California 92656, US

(54) Bezeichnung: **Einbrennsystem und -verfahren für verbesserte Speicherzuverlässigkeit**

Beschreibung

[0001] Ein Ausführungsbeispiel der vorliegenden Erfindung betrifft ein System und ein Verfahren zum Testen hierarchischer Speicherarchitekturen. Genauer gesagt betrifft ein Ausführungsbeispiel der vorliegenden Erfindung ein System und ein Verfahren zum Anlegen einer Belastung an Speicherzellen in einer hierarchischen Speicherarchitektur in einer parallelen Art und Weise.

[0002] Hierarchische Speicherarchitekturen, wie zum Beispiel SRAM-Module, sind zu einem integralen Bestandteil moderner VLSI-Systeme geworden. Solche hoch integrierten Hochleistungskomponenten für VLSI-Systeme erfordern komplexe Fabrikations- und Verarbeitungsprozesse. Diese Prozesse können unvermeidbare Parameterfehler erfahren, die bei den SRAM-Modulen oder den größeren VLSI-Systemen zwangsweise zu unerwünschten Defekten führen können. In einem Ausführungsbeispiel ist eine Redundanz in die Speicherarchitektur eingebaut, die eine Eins-zu-Eins-Ersetzung für schadhafte Teile oder Subsysteme bereitstellt.

[0003] Während der Fabrikation und der Verarbeitung können elektrische Tests Defekte entdecken, die in den VLSI-Systemen oder ihren Komponenten, die die SRAM-Module einschließen, Schaltungsversagen bzw. Schaltungsstörungen verursachen können. Solche erfassten fehlerhaften Systeme oder Komponenten werden entweder repariert oder weggeworfen.

[0004] Aber es gibt eine Klasse von Defekten, die keine sofortige elektrische Störung verursacht, sondern eher zu einem Versagen während des Einsatzes an Ort und Stelle führt, nachdem das Teil bereits verpackt und verschickt worden ist. Störungen bzw. Versagen an Ort und Stelle während des Einsatzes sind kostenintensiv und schädigen den Ruf des Herstellers in Bezug auf die Zuverlässigkeit. Solche Defekte werden allgemein als "schwache Defekte" bezeichnet.

[0005] Als eine Folge davon werden die Systeme, die Subsysteme und ihre kleineren Komponenten während der Herstellung getestet, um solche schwachen Defekte zu erfassen. Die Systeme, Subsysteme und Komponenten werden einer Belastung ausgesetzt, die ein bevorstehendes Versagen beschleunigt, so dass die Teile, Subsysteme oder Komponenten vor dem Verpacken entweder repariert oder weggeworfen werden können. Logische Teile eines Chips können belastet werden, indem Testvektoren mit einer vorbestimmten Hochspannung durch die Chipschaltkreise geleitet werden, wodurch die Schaltungen einer vorbestimmten hohen Spannung und Temperatur ausgesetzt werden.

[0006] Das Belasten einer großen Speicherstruktur könnte potentiell einen unpraktischen Betrag an Zeit kosten, um diesen Vorgang während des Herstellungsprozesses komplett durchzuführen. Zum Beispiel weist ein 10 Megabit-Speicher 10 Millionen Einträge auf, von denen jeder getestet werden muss. Das sequentielle Testen von 10 Millionen Einträgen würde zu einer extrem langen Testzeit führen. Solche langen Testzeiten sind teuer.

[0007] Weitere Beschränkungen und Nachteile von herkömmlichen und traditionellen Lösungswegen werden einem Fachmann auf dem Gebiet offensichtlich werden, wenn er solche Systeme mit der vorliegenden Erfindung vergleicht, wie sie in dem restlichen Teil der vorliegenden Anmeldung unter Bezugnahme auf die Zeichnungen dargelegt ist.

[0008] Das Dokument US 5 995 427 offenbart ein DRAM, das einen Bitleitungs-Potentialeingabeknoten eines Ausgleichers (equalizer) bereitstellt, der für jedes ungeradzahlige Paar von Bitleitungen bereitgestellt wird. Dieser Bitleitungs-Potentialeingabeknoten ist separat von einem Bitleitungs-Potentialeingabeknoten eines Ausgleichers bereitgestellt, der für jedes geradzahlige Paar von Bitleitungen bereitgestellt wird.

[0009] Das Dokument US 5 852 581 offenbart einen Wafer-Burn-in-Speicher-Selbstbelastungsmodus für integrierte Schaltungen eines dynamischen Direktzugriffsspeichers (DRAM). Eine Burn-in-Stromversorgungsspannung und eine Erdspannung werden einem gemeinsamen Knoten einer Vielzahl von Speicherzellen-Speicherkondensatoren und einem Ausgleichknoten zugeführt, die mit den Bitleitungen gekoppelt sind.

[0010] Das Dokument US 6 327 682 offenbart ein Verfahren zum Burn-in-Testen (Durchbrenntesten) von entweder DRAMs oder FeRAMs auf der Wafer-Ebene. Eine Belastungsspannung wird quer durch alle Speicherkondensatoren eines DRAM oder eines FeRAM während eines Burn-in-Tests auf Wafer-Ebene angelegt, um Speicherchips mit schwachen Speicherzellen zu entfernen.

[0011] Das Dokument US 2001/0045570 offenbart eine Halbleiterspeicherungsvorrichtung, die eine verkettete Burn-in-Funktion von Hauptwortleitungen aufweist, wobei eine Belastungsspannung zwischen den Hauptwortleitungen in einem Wafer-Burn-in-Zustand angelegt wird.

[0012] Es ist Aufgabe der vorliegenden Erfindung, ein Verfahren zum Belasten einer hierarchischen Speicherstruktur bereitzustellen, wobei das Verfahren zu kurzen Testzeiten führt und somit die Kosten reduziert.

[0013] Diese Aufgabe wird von einem Verfahren erfüllt, wie dies in Anspruch 1 angegeben ist.

[0014] Vorteilhafte Ausführungsbeispiele der Erfindung sind in den Unteransprüchen definiert.

[0015] Ein Ausführungsbeispiel der vorliegenden Erfindung betrifft ein Verfahren zum Belasten einer Speicherstruktur. Das Verfahren umfasst das Initialisieren der Speicherstruktur in einen ersten Zustand und das Erzeugen einer Belastung durch wenigstens ein Paar von Leitungen, die mit wenigstens der Speicherstruktur gekoppelt sind. Diese Speicherstruktur wird dann in einen zweiten Zustand initialisiert und eine Belastung wird quer durch wenigstens ein Paar von Leitungen erzeugt.

[0016] Ein Ausführungsbeispiel der vorliegenden Erfindung betrifft ein System und ein Verfahren zum Anlegen einer Belastung an eine hierarchische Speicherstruktur in einer parallelen Art und Weise, wodurch die Speicherstruktur auf schwache Defekte getestet wird. Die vorliegende Erfindung umfasst das Schreiben einer logischen 0 in alle Speicherzellen einer Speicherstruktur. Alle vordecodierten Leitungen einer hohen Adresse und alternierende vordecodierte Leitungen für die niedrigste Adresse werden aktiviert. Ein Spannungsabfall zwischen benachbarten Wortleitungen und Bitleitungen wird beeinflusst. Eine logische 1 wird in alle Speicherzellen der Speicherstruktur geschrieben. Eine entgegengesetzte Spannungspolarität wird in den Bitleitungen aufgrund der logischen 1 in den Speicherzellen erzeugt. Eine umgekehrte Spannungspolaritätsbelastung wird in den Wortleitungen erreicht, indem der Zustand der niedrigsten vordecodierten Leitung gewechselt wird (z.B. durch das Ändern der Eingabeadresse, die dieser Leitung entspricht).

[0017] Weitere Ausführungsformen, Vorteile und neuartige Merkmale der vorliegenden Erfindung sowie auch Einzelheiten eines veranschaulichten Ausführungsbeispiels davon werden aus der nachfolgenden Beschreibung und den nachfolgenden Zeichnungen besser verständlich, in denen sich gleiche Bezugszeichen auf gleiche Teile beziehen.

KURZE BESCHREIBUNG MEHRERER ANSICHTEN DER ZEICHNUNGEN

[0018] Fig. 1 veranschaulicht ein Blockdiagramm eines beispielhaften SRAM-Moduls;

[0019] Fig. 2 veranschaulicht ein Blockdiagramm eines SRAM-Speicherkerns, der in Bänke aufgeteilt ist;

[0020] Fig. 3A und Fig. 3B veranschaulichen SRAM-Module gemäß einem Ausführungsbeispiel der vorliegenden Erfindung, die eine Blockstruktur oder ein Subsystem umfassen;

[0021] Fig. 4 veranschaulicht eine dimensionale Block-Array oder ein Subsystem, die/das in einem SRAM-Modul verwendet wird, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0022] Fig. 5 veranschaulicht eine Zell-Array gemäß einem Ausführungsbeispiel der vorliegenden Erfindung, die eine Vielzahl von Speicherzellen umfasst;

[0023] Fig. 6A veranschaulicht eine Speicherzelle, die gemäß einem Ausführungsbeispiel der vorliegenden Erfindung verwendet wird;

[0024] Fig. 6B veranschaulicht antiparallelgeschaltete Inverter, die die Speicherzelle von Fig. 6A gemäß einem Ausführungsbeispiel der vorliegenden Erfindung darstellen;

[0025] Fig. 7 veranschaulicht ein SRAM-Modul, das demjenigen ähnlich ist, das in den Fig. 3A und Fig. 3B veranschaulicht worden ist, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0026] Fig. 8 veranschaulicht einen lokalen Decoder gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0027] Fig. 9 veranschaulicht ein Schaltbild eines lokalen Decoders, der dem ähnlich ist, der in Fig. 8 veranschaulicht worden ist, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0028] Fig. 10 veranschaulicht ein Blockdiagramm der lokalen Leseverstärker und des 4:1-Multiplexens gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0029] Fig. 11 veranschaulicht ein Blockdiagramm der lokalen Leseverstärker und globalen Leseverstärker gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0030] Fig. 12A veranschaulicht eine schematische Darstellung der lokalen Leseverstärker und globalen Leseverstärker gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0031] Fig. 12B veranschaulicht ein Schaltbild eines Ausführungsbeispiels eines lokalen Leseverstärkers (der dem lokalen Leseverstärker von Fig. 12A ähnlich ist) gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0032] Fig. 12C veranschaulicht eine schematische Darstellung des Verstärkerkerns, der dem Verstärkerkern ähnlich ist, der in Fig. 12B veranschaulicht ist;

[0033] Fig. 13 veranschaulicht ein Blockdiagramm eines anderen Ausführungsbeispiels der lokalen Leseverstärker und der globalen Leseverstärker gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0034] Fig. 14 veranschaulicht ein Schaltbild, das ein Übertragungsgatter des 4:1-Mux umfasst, das dem ähnlich ist, das in den Fig. 10 und 12 veranschaulicht ist, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0035] Fig. 15 veranschaulicht Übertragungsgatter des 2:1-Mux, die mit den Invertern eines lokalen Leseverstärkers gekoppelt sind, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0036] Fig. 16 veranschaulicht die Vorlade- und Ausgleichsabschnitte und Übertragungsgatter des 2:1-Mux, die mit den Invertern eines lokalen Leseverstärkers gekoppelt sind, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0037] Fig. 17 veranschaulicht ein Schaltbild des lokalen Leseverstärkers gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0038] Fig. 18 veranschaulicht ein Blockdiagramm eines lokalen Controllers gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0039] Fig. 19 veranschaulicht ein Schaltbild des lokalen Controllers gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0040] Fig. 20 veranschaulicht die Zeitsteuerung (Timing) für einen LESE-Zyklus unter Verwendung eines SRAM-Speichermoduls gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0041] Fig. 21 veranschaulicht die Zeitsteuerung eines SCHREIB-Zyklus unter Verwendung eines SRAM-Speichermoduls gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0042] Fig. 22A veranschaulicht ein Blockdiagramm eines lokalen Leseverstärkers mit einem 4:1-Multiplexen und Vorladen, die darin enthalten sind, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0043] Fig. 22B veranschaulicht ein Beispiel eines 16:1-Multiplexens (das das globale 4:1-Multiplexen und das lokale 4:1-Multiplexen umfasst) gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0044] Fig. 22C veranschaulicht ein Beispiel eines 32:1-Multiplexens (das das globale 8:1-Multiplexen und das lokale 4:1-Multiplexen umfasst) gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0045] Fig. 23 veranschaulicht einen lokalen Leseverstärker, der mit einer Cluster-Schaltung verwendet wird, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0046] Fig. 24 veranschaulicht ein Flussdiagramm, das das Anlegen einer Belastung an eine hierarchische Speicherarchitektur gemäß einem Ausführungsbeispiel der vorliegenden Erfindung demonstriert;

[0047] Fig. 25 veranschaulicht eine Speicherzellen-Array, bei der eine volle Spannungsbelastung an einen ersten Satz von benachbarten Leitungen angelegt wird, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0048] Fig. 26 veranschaulicht eine Speicherzellen-Array, bei der eine volle Spannungsbelastung an einen zweiten Satz von benachbarten Leitungen angelegt wird, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0049] Fig. 27 veranschaulicht ein Schaltbild des lokalen Leseverstärkers, der dem von Fig. 17 ähnlich ist, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung, wobei veranschaulicht wird, wie die lokalen Bitleitungen mit den globalen Bitleitungen durch ein Durchgangsgatter verbunden sind;

[0050] Fig. 28 veranschaulicht eine bereits bekannte Taktvordecodiererschaltung des NOR-Typs (NICHT-ODER-Typs);

[0051] Fig. 29 veranschaulicht eine Vorcodiererschaltung mit Burn-in (Durchbrennen) gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0052] Fig. 30 veranschaulicht eine logische Vorcodiererschaltung mit Burn-in, die einen komplexen booleschen Ausdruck durchführt, gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0053] Fig. 31 veranschaulicht eine andere logische Vorcodiererschaltung mit einem Burn-in gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0054] Fig. 32 veranschaulicht eine Stapel-Schaltung gemäß einem Ausführungsbeispiel der vorliegenden Erfindung, die mit einer logischen Vorcodiererschaltung verwendet wird, die derjenigen ähnlichen ist, die in Fig. 31 veranschaulicht ist; und

[0055] Fig. 33 veranschaulicht noch eine andere logische Vorcodiererschaltung, die den Stapel enthält, der in Fig. 32 veranschaulicht ist, gemäß einem Aus-

führungsbeispiel der vorliegenden Erfindung.

AUSFÜHRLICHE BESCHREIBUNG DER ERFINDUNG

[0056] Wie einem Fachmann auf dem Gebiet bekannt sein wird, enthalten die meisten VLSI-Systeme, die Kommunikationssysteme und DSP-Geräte umfassen, VLSI-Speicher-Subsysteme. Moderne Anwendungen von VLSI-Speicher-Subsystemen erfordern beinahe gleich bleibend hochwirksame Hochleistungsimplementierungen, die die Designkompromisse zwischen Layout-Effizienz, Geschwindigkeit, Stromverbrauch, Skalierbarkeit, Designtoleranzen und dergleichen vergrößern. Die vorliegende Erfindung verbessert diese Kompromisse, indem sie eine neuartige synchrone selbstgetaktete hierarchische Architektur verwendet. Das Speichermodul der vorliegenden Erfindung kann auch ein oder mehrere neuartige Komponenten verwenden, die weiter zu der Effizienz und Robustheit des Speichermoduls beitragen.

[0057] Es sollte klar sein, dass es nützlich ist, die verschiedenen Ausführungsformen und Ausführungsbeispiele der Erfindung hier in dem Kontext einer SRAM-Speicherstruktur zu beschreiben, die CMOS-SRAM-Speicherzellen verwendet. Aber es sollte den Fachleuten auf dem Gebiet auch klar sein, dass die vorliegende Erfindung nicht auf CMOS-basierte Prozesse beschränkt ist, und dass diese Ausführungsformen und Ausführungsbeispiele in anderen Speicherprodukten als einer SRAM-Speicherstruktur verwendet werden können, die DRAM, ROM, PLA und dergleichen umfassen, aber nicht auf diese beschränkt sind, wobei es egal ist, ob diese in einem VLSI-System eingebettet sind oder eigenständige Speichervorrichtungen sind.

BEISPIELHAFTES SRAM-MODUL

[0058] Fig. 1 veranschaulicht ein Funktionsblockdiagramm eines ersten Beispiels einer SRAM-Speicherstruktur **100**, die die grundlegenden Merkmale von SRAM-Subsystemen bereitstellt. Das Modul **100** umfasst einen Speicherkern **102**, einen Wortleitungscontroller **104**, und Speicheradresseingänge **114**. In diesem beispielhaften Ausführungsbeispiel setzt sich der Speicherkern **102** aus einer zweidimensionalen Array von K Bits an Speicherzellen **103** zusammen, die so angeordnet sind, dass sie C Spalten und R Zeilen bzw. Reihen von Bitspeicherstellen aufweisen, wobei K = [C × R]. Die üblichste Konfiguration eines Speicherkerns **12** verwendet Einzelwortleitungen **106**, um die Zellen **106** mit den gepaarten differentiellen Bitleitungen **118** zu verbinden. Im Allgemeinen ist der Kern **102** als eine Array von 2^P Einträgen angeordnet, die auf einem Satz von darin enthaltenen P Speicheradressen basieren. Somit wird die p-Bit-Adresse von einem Zeilenadressdecoder **110** und einem Spaltenadressdecoder **122** decodiert. Ein Zugriff auf eine gegebene Speicherzelle **103** in einem solchen Einzelkernspeicher **102** wird erreicht, indem die Spalte **105** durch das Auswählen einer Bitleitung in der Spalte aktiviert wird, die der Zelle **103** entspricht.

[0059] Die spezielle Zeile, auf die zugegriffen werden soll, wird durch das selektive Aktivieren des Zeilenadress- oder Wortleitungsdecoders **110** ausgewählt, der normalerweise eindeutig mit einer gegebenen Zeile oder Wortleitung übereinstimmt, der alle Zellen **103** in dieser speziellen Zeile überspannt. Auch ein Wortleitungstreiber **108** kann eine ausgewählte Wortleitung **106** so ansteuern, dass in die ausgewählte Speicherzelle **103** auf einem bestimmten Paar von Bitleitungen **118** geschrieben werden kann oder aus dieser ausgelesen werden kann, und zwar in Übereinstimmung mit der Bitadresse, die den Speicheradresseingängen **114** zugeführt wird.

[0060] Ein Bitleitungscontroller **116** kann Vorladezellen (precharge cells) (nicht gezeigt), Spaltenmultiplexer oder -decoder **122**, Leseverstärker **124** und Ein-/Ausgabepuffer (nicht gezeigt) umfassen. Da typischerweise unterschiedliche LESE-/SCHREIB-Programme für die Speicherzellen verwendet werden, ist es wünschenswert, dass die Bitleitungen in einen wohldefinierten Zustand gesetzt werden, bevor auf diese zugegriffen wird. Vorladezellen können dazu verwendet werden, den Zustand der Bitleitungen **118** durch einen VORLADE-Zyklus gemäß einem vordefinierten Vorladeprogramm einzurichten. In einem statischen Vorladeprogramm können die Vorladezellen kontinuierlich eingeschaltet bleiben, ausgenommen dann, wenn auf einen bestimmten Block zugegriffen wird.

[0061] Zusätzlich zu dem Einrichten eines definierten Zustands in den Bitleitungen **118** können die Vorladezellen auch dazu verwendet werden, einen Ausgleich von Differenzspannungen in den Bitleitungen **118** vor einer LESE-Operation zu bewirken. Leseverstärker **124** ermöglichen es, dass die Größe der Speicherzelle **103** reduziert werden kann, indem die Differenzspannung in den Bitleitungen **118** abgefühlt wird, was ihren Zustand anzeigt, wodurch diese Differenzspannung in ein Logikpegelsignal umgewandelt wird.

[0062] In dem beispielhaften Ausführungsbeispiel wird eine LESE-Operation durchgeführt, indem ein Zeilendecoder **110** aktiviert wird, der eine bestimmte Zeile auswählt. Die Ladung in einer der Bitleitungen **118** von jedem Paar von Bitleitungen in jeder Spalte wird durch die aktivierte Speicherzelle **103** entladen, was den Zustand der aktiven Zellen **103** in dieser Spalte **105** darstellt. Der Spaltendecoder **122** aktiviert nur eine der Spalten, wobei die Bitleitungen **118** mit einem Ausgang gekoppelt werden. Die Leseverstärker **124** stellen die Ansteuerfähigkeit bereit, um

Strom für den Ausgang zu liefern, der Ein-/Ausgabepuffer umfasst. Wenn der Leseverstärker **124** aktiviert ist, werden die unsymmetrischen Bitleitungen **118** bewirken, dass der symmetrische Leseverstärker zu dem Zustand der Bitleitungen schaltet, und Daten werden ausgegeben werden.

[0063] Im Allgemeinen wird eine SCHREIB-Operation dadurch ausgeführt, dass Daten an einen Eingang angelegt werden, der E/A-Puffer (nicht gezeigt) umfasst. Vor der SCHREIB-Operation können die Bitleitungen **118** durch Vorladezellen auf einen vorbestimmten Wert vorgeladen werden. Das Anlegen von Eingabedaten an die Eingänge neigt dazu, die Vorladespannung in einer der Bitleitungen **118** zu entladen, wodurch eine Bitleitung mit einem logischen HIGH (HOCH) und eine Bitleitung mit einem logischen LOW (NIEDRIG) zurückgelassen wird. Der Spaltendecoder **122** wählt eine bestimmte Spalte **105** aus, verbindet die Bitleitungen **118** mit dem Eingang, wodurch eine der Bitleitungen **118** entladen wird. Der Zeilendecoder **110** wählt eine bestimmte Zeile aus, und die Informationen in den Bitleitungen **118** werden an der Schnittstelle zwischen der Spalte **105** und der Zeile **106** in die Zelle **103** geschrieben.

[0064] Zu Beginn eines typischen internen Zeitsteuerungszyklus wird das Vorladen deaktiviert. Das Vorladen wird nicht wieder aktiviert, bis die gesamte Operation vollständig durchgeführt worden ist. Der Spaltendecoder **122** und der Zeilendecoder **110** werden dann aktiviert, woraufhin die Aktivierung des Leseverstärkers **124** folgt. Bei Beendigung einer LESE- oder SCHREIB-Operation wird der Leseverstärker **124** deaktiviert. Daraufhin werden die Decoder **110**, **122** deaktiviert, zu welchem Zeitpunkt die Vorladezellen **120** während eines darauf folgenden VORLADE-Zyklus wieder aktiv werden.

STROMREDUZIERUNG UND GESCHWINDIGKEITSVERBESSERUNG

[0065] Unter Bezugnahme auf Fig. 1 wird der Inhalt der Speicherzelle **103** des Speicherblocks **100** in dem Leseverstärker **124** erfasst, wobei eine differentielle Leitung zwischen den gepaarten Bitleitungen **118** verwendet wird. Es sollte klar sein, dass diese Architektur nicht skalierbar ist. Auch das Vergrößern des Speicherblocks **100** kann die praktischen Beschränkungen des Leseverstärkers **124**, um ein adäquates Signal in einer zeitgerechten Art und Weise an den Bitleitungen **118** zu empfangen, überschreiten. Das Vergrößern der Länge der Bitleitungen **118** steigert die assoziierte Bitleitungskapazität, und somit steigert dies die Zeit, die für das Entwickeln einer Spannung darin benötigt wird. Es muss den Leitungen **104**, **106** mehr Strom zugeführt werden, um diese zusätzliche Kapazität zu überwinden.

[0066] Außerdem benötigt es mehr Zeit, die langen Bitleitungen in den Architekturen des bestehenden Standes der Technik vorzuladen, wodurch die effektive Vorrichtungsgeschwindigkeit reduziert wird. In ähnlicher Weise erfordert das Schreiben in längere Bitleitungen **118**, wie dies im bestehenden Stand der Technik vorgefunden wird, einen extensiveren Strom. Dies steigert die Stromanforderungen der Schaltung und reduziert auch die effektive Vorrichtungsgeschwindigkeit.

[0067] Im Allgemeinen kann ein reduzierter Stromverbrauch in Speichervorrichtungen wie etwa der Struktur **100** in Fig. 1 zum Beispiel dadurch erreicht werden, dass die gesamte Schaltkapazität verringert wird und die Spannungshübe minimiert werden. Die Vorteile der Leistungsreduzierungsaspekte bestimmter Ausführungsbeispiele der vorliegenden Erfindung können im Kontext der Schaltkapazitätsverringerung und der Spannungshubbegrenzung weiter erkannt werden.

REDUZIERUNG DER SCHALTKAPAZITÄT

[0068] Wenn die Bitdichte von Speicherstrukturen ansteigt, ist beobachtet worden, dass Einzelkern-Speicherstrukturen unakzeptabel große Schaltungskapazitäten aufweisen können, die mit jedem Speicherzugriff verbunden sind. Ein Zugriff auf irgendeine Bitstelle innerhalb eines solchen Einzelkernspeichers erfordert das Aktivieren der gesamten Zeile oder Wortleitung **106**, in der die Daten gespeichert sind, und das Schalten aller Bitleitungen **118** in der Struktur. Deshalb ist es wünschenswert, Hochleistungs-Speicherstrukturen zu entwerfen, um die gesamte Schaltungskapazität während eines gegebenen Zugriffs zu reduzieren.

[0069] Zwei allgemein bekannte Lösungswege zur Reduzierung der gesamten Schaltungskapazität während eines Speicherstrukturzugriffs umfassen das Teilen einer Einzelkern-Speicherstruktur in eine mit Bänken versehene bzw. verschachtelte Speicherstruktur und das Verwenden geteilter Wortleitungsstrukturen. Bei dem ersten Lösungsweg ist es notwendig, nur die bestimmte Speicherbank zu aktivieren, die mit der Speicherzelle von Interesse verbunden ist. Bei dem letzteren Lösungsweg reduziert die festgelegte Wortleitungsaktivierung in dem größten praktizierbaren Ausmaß die gesamte Schaltungskapazität.

GETEILTER SPEICHERKERN ODER SPEICHERKERN MIT BÄNKEN

[0070] Ein Lösungsweg zur Reduzierung von Schaltungskapazitäten liegt darin, den Speicherkern in separat schaltbare Bänke von Speicherzellen aufzuteilen.

[0071] Ein Beispiel eines Speicherkerns **200**, der in

Bänke aufgeteilt ist, ist in Fig. 2 veranschaulicht. In dem veranschaulichten Ausführungsbeispiel umfasst der Speicherkern zwei Bänke von Speicherzellen, nämlich Bank #0 und Bank #1, die jeweils allgemein mit **202** und **204** bezeichnet sind. Der Speicherkern **200** umfasst zwei lokale Decoder **206**, die kommunikativ miteinander und mit einem globalen Decoder **208** über eine Wortleitung High **210** gekoppelt sind. Jeder lokale Decoder **206** umfasst eine lokale Wortleitung High **210**, die den Decoder **206** kommunikativ mit seiner zugeordneten Bank koppelt. Außerdem sind die beiden Bankleitungen **214** so dargestellt, dass sie kommunikativ mit den lokalen Decodern **206** gekoppelt sind oder über eine Schnittstelle mit diesen verbunden sind. Es sollte klar sein, dass in einem Ausführungsbeispiel mit jeder Bank eine Bankleitung **214** assoziiert ist.

[0072] Typischerweise ist die gesamte Schaltungskapazität während eines gegebenen Speicherzugriffs für Speicherkerne, die mit Bänken versehen sind, umgekehrt proportional zu der Anzahl an verwendeten Bänken. Durch ein kluges Auswählen der Anzahl und der Platzierung der Bankeinheiten innerhalb eines gegebenen Speicherkerndesigns sowie auch des Typs von verwendeter Decodierung kann die gesamte Schaltungskapazität und somit der gesamte Strom, der von dem Speicherkern verbraucht wird, zu einem großen Teil reduziert werden. Entwürfe mit Bänken können auch einen höheren Produktertrag realisieren. Die Speicherbänke können so angeordnet werden, dass eine defekte Bank inoperabel und unzugänglich gemacht wird, während die restlichen betriebsfähigen Bänke des Speicherkerns **200** in ein Produkt mit einer niedrigeren Kapazität gepackt werden können.

[0073] Aber Designs mit Banken können für bestimmte Anwendungen nicht geeignet sein. Geteilte Speicherkerne erfordern zusätzliche Decodierschaltungen, um einen selektiven Zugriff auf die einzelnen Bänke zu erlauben. Mit anderen Worten, solche geteilten Speicherkerne können zum Beispiel zusätzlich einen lokalen Decoder **206**, eine lokale Bankleitung **214** und eine lokale Wortleitung High **210** erfordern. Als eine Folge davon kann eine Verzögerung auftreten. Auch verwenden viele Designs mit Bänken Speichersegmente, die lediglich verkleinerte Versionen von traditionellen monolithischen Kernspeicherdesigns sind, wobei jedes Segment eine zugeordnete Steuerungs-, Vorlade-, Decodier-, Lese- und Ansteuerungsschaltung aufweist. Diese Schaltungen neigen dazu, sowohl im Standby-Modus als auch im Betriebsmodus viel mehr Strom zu verbrauchen als ihre assoziierten Speicherzellen. Solche Strukturen mit Bänken können einfach zu entwerfen sein, aber die zusätzliche Komplexität und der zusätzliche Stromverbrauch können die Gesamtspeicherkomponentenperformanz reduzieren.

[0074] Durch ihre nämliche Beschaffenheit sind Entwürfe mit Bänken nicht für das Aufwärtsskalieren bzw. Vergrößern geeignet, um eine Anpassung an große Designanforderungen vorzunehmen. Auch sind traditionelle Designs mit Bänken nicht ohne weiteres an Anwendungen anpassbar, die eine Speicherkernkonfiguration benötigen, die im Wesentlichen anders als die darunter liegende Bankarchitektur ist (z.B. eine Speicherstruktur, die relativ wenige Zeilen mit Langwortlängen benötigt). Traditionelle Bankdesigns sind im Allgemeinen nicht ohne weiteres an eine Speicherstruktur anpassbar, die relativ wenige Zeilen von sehr langen Wortlängen benötigt.

[0075] Anstatt zu einer Top-Down-Teilung der Basisspeicherstruktur Zuflucht zu nehmen, die Speicherdesigns mit Bänken verwendet, stellen ein oder mehrere Ausführungsbeispiele der vorliegenden Erfindung eine hierarchische Speicherstruktur bereit, die unter Verwendung eines Bottom-Up-Ansatzes synthetisiert wird. Das hierarchische Koppeln von Basisspeichermodulen mit lokalisierten entscheidungstreffenden Merkmalen, die synergistisch kooperieren, reduziert den gesamten Strombedarf der Struktur dramatisch und verbessert die Betriebsgeschwindigkeit der Struktur. Als ein Minimum kann ein solches grundlegendes hierarchisches Modul ein lokalisiertes Bitleitungsabfühlen umfassen.

GETEILTE WORTLEITUNG

[0076] Oftmals ist die Bitbreite einer Speicherkomponente größenmäßig so bemessen, dass sie eine bestimmte Wortlänge unterbringt. Wenn die Wortlänge für ein spezielles Design größer wird, vergrößern sich auch die assoziierten Wortleitungsverzögerung, die Schaltungskapazität, der Stromverbrauch, usw. Um sehr lange Wortleitungen unterzubringen, kann es wünschenswert sein, kernüberspannende globale Wortleitungen in lokale Wortleitungen zu unterteilen, wobei jede aus kleineren Gruppe von benachbarten wortorientierten Speicherzellen besteht. Jede lokale Gruppe verwendet lokale Decodier- und Ansteuerungskomponenten, um die lokalen Wortleitungen zu erzeugen, wenn die globale Wortleitung, mit der sie gekoppelt sind, aktiviert wird. Bei Anwendungen mit langen Wortlängen kann der zusätzliche Overhead, der sich durch die geteilten Wortleitungen ergibt, durch reduzierte Wortleitungsverzögerungen ausgeglichen werden.

[0077] Anstatt zu der traditionellen Top-Down-Teilung der Wortleitungen Zuflucht zu nehmen, umfassen gewisse Ausführungsbeispiele der hier beschriebenen Erfindung das Bereitstellen einer lokalen Wortleitung für das oben erwähnte Basisspeichermodul, was die lokalen entscheidungstreffenden Merkmale des Moduls weiter verbessert. Wie zuvor können, indem ein Bottom-Up-Ansatz für die hierarchischen gekoppelten Basisspeichermodule verwendet wird, wie

dies vorher beschrieben worden ist, mit den hinzugefügten lokalisierten entscheidungstreffenden Merkmalen von lokalen Wortleitungen gemäß der vorliegenden Erfindung zusätzliche Synergien realisiert werden, was den gesamten Stromverbrauch und die Signallaufzeiten weiter reduziert.

MULTIPLEXING

[0078] Eine Alternative zu einem Speicherkerndesign mit Bänken liegt darin, die Speicherzellen einem Multiplexen oder Muxen zu unterziehen. Mit anderen Worten, Bits aus unterschiedlichen Worten werden nicht sequentiell gespeichert. So werden zum Beispiel bei dem 2:1-Muxen Bits von zwei Worten in einem alternierenden Muster gespeichert. Wenn die Zahl 1 zum Beispiel Bits von einem ersten Wort repräsentiert, repräsentiert die Zahl 2 die Bits von einem zweiten Wort. Während einer LESE- oder SCHREIB-Operation wählt der Mux aus, welche Spalte er betrachtet (d.h. das linke Bit oder das rechte Bit). Es sollte klar sein, dass das Muxen bzw. Multiplexen Platz sparen kann. Designs mit Bänken ohne Muxen erfordern einen Leseverstärker für jeweils zwei Leitungen. In einem 2:1-Muxen wird zum Beispiel ein Leseverstärker für jeweils vier Leitungen verwendet (d.h., ein Leseverstärker verbindet zwei Sätze von Bitleitungen). Das Muxen ermöglicht es, dass die Leseverstärker von gemuxten Zellen gemeinsam benutzt werden können, was den Layout-Abstand (layout pitch) und die Flächeneffizienz vergrößern kann.

[0079] Im Allgemeinen verbraucht das Muxen mehr Strom als das Speicherkerndesign mit Bänken. Um zum Beispiel ein gespeichertes Wort zu lesen, greift der Mux auf eine gesamte Zeile in der Zell-Array zu oder aktiviert diese, wobei er alle darin gespeicherten Daten liest, nur die benötigten Daten abfühlt und den Rest unbeachtet lässt.

[0080] Wenn man einen Bottom-Up-Ansatz bei hierarchisch gekoppelten Basisspeichermodulen mit Multiplexing gemäß einem Ausführungsbeispiel der vorliegenden Erfindung verwendet, werden zusätzliche Synergien realisiert, wodurch der Stromverbrauch und die Signallaufzeiten reduziert werden.

SPANNUNGSHUB-REDUKTIONSTECHNIKEN

[0081] Die Stromreduzierung kann auch dadurch erreicht werden, dass die Spannungshübe reduziert werden, die in der gesamten Struktur erfahren werden. Durch das Begrenzen der Spannungshübe ist es möglich, den Betrag an Strom zu reduzieren, der abgegeben wird, wenn die Spannung an einem Knoten oder in einer Leitung während eines bestimmten Ereignisses oder einer bestimmten Operation ausschwingt bzw. abklingt, sowie auch den Betrag an Strom zu reduzieren, der notwendig ist, um die verschiedenen abgeklungenen Spannungen nach dem bestimmten Ereignis oder der bestimmten Operation oder vor dem nächsten Zugriff in den gewünschten Zustand zurückzubringen. Zwei Techniken zu diesem Zweck umfassen die Verwendung von gepulsten Wortleitungen und die Leseverstärker-Spannungshubreduktion.

GEPULSTE WORTLEITUNGEN

[0082] Durch das Bereitstellen einer Wortleitung mit einer Länge, die gerade lang genug ist, um die Differenzspannung quer durch eine ausgewählte Speicherzelle zu erfassen, ist es möglich, die Bitleitungs-Spannungsentladung zu reduzieren, die einer LESE-Operation der ausgewählten Zelle entspricht. In einigen Designs wird durch das Anlegen eines gepulsten Signals an die zugehörige Wortleitung über ein ausgewähltes Zeitintervall ein Leseverstärker nur während dieses Intervalls aktiviert, wodurch die Dauer des Bitleitungs-Spannungsabklingens reduziert wird. Diese Designs verwenden typischerweise einige der Impulsgeneratoren, die einen Impuls mit festgelegter Dauer erzeugen. Wenn die Dauer des Impulses darauf abzielt, die Zeitsteuerungsszenarien des schlimmsten Falles zufrieden zu stellen, wird der zusätzliche Spielraum zu einem unnötigen Abziehen von Bitleitungsstrom während nomineller Operationen führen.

[0083] Deshalb kann es wünschenswert sein, eine selbstgetaktete, selbstbegrenzende Wortleitungsvorrichtung zu verwenden, die auf die tatsächliche Dauer einer gegebenen LESE-Operation in einer ausgewählten Zelle anspricht, und die die Wortleitungsaktivierung im Wesentlichen auf diese Dauer beschränkt. Des Weiteren kann es dann, wenn ein Leseverstärker eine LESE-Operation in weniger als einem Speichersystem-Taktzyklus erfolgreich beendet, auch wünschenswert sein, eine asynchrone Impulsbreitenaktivierung relativ zu dem Speichersystemtakt zu haben. Gewisse Ausführungsformen der vorliegenden Erfindung können ein gepulstes Wortleitungssignal zum Beispiel unter Verwendung einer kooperativen Interaktion zwischen einem lokalen Decoder und einem lokalen Controller bereitstellen.

LESEVERSTÄRKER-SPANNUNGSHUBREDUZIERUNG

[0084] Um große Speicher-Arrays herzustellen, ist es äußerst wünschenswert, die Größe einer individuellen Speicherzelle auf einem Minimum zu halten. Als eine Folge davon sind einzelne Speicherzellen im Allgemeinen nicht in der Lage, einen Ansteuerstrom für assoziierte Eingangs-/Ausgangs-Bitleitungen zu liefern. Leseverstärker werden typischerweise dazu verwendet, den Wert der Daten zu erfassen, die in einer bestimmten Speicherzelle gespeichert sind, und den Strom bereitzustellen, der benötigt wird, um die

E/A-Leitungen anzusteuern.

[0085] In einem Leseverstärkerdesign gibt es typischerweise einen Kompromiss zwischen Energie und Geschwindigkeit, wobei schnellere Antwortzeiten für Gewöhnlich größere Energieanforderungen diktieren. Schnellere Leseverstärker können auch dazu neigen, physisch zwar größere Vorrichtungen, aber relativ dazu Vorrichtungen mit einer geringen Geschwindigkeit und niedrigen Energie zu sein. Des Weiteren kann die analoge Beschaffenheit der Leseverstärker dazu führen, dass sie einen beträchtlichen Teil der gesamten Energie verbrauchen. Obwohl eine Möglichkeit zur Verbesserung der Ansprechfähigkeit eines Leseverstärkers darin liegt, einen empfindlicheren Leseverstärker zu verwenden, wird jeder gewonnene Nutzen von der damit verbundenen Schaltungskomplexität aufgehoben, die nichtsdestoweniger an einer erhöhten Rauschempfindlichkeit leidet. Es ist dann wünschenswert, die Bitleitungs-Spannungshübe zu beschränken und die Energie zu reduzieren, die von dem Leseverstärker verbraucht wird.

[0086] In einem typischen Design erfasst der Leseverstärker die kleinen differentiellen Signale quer durch eine Speicherzelle, die sich in einem unsymmetrischen Zustand befindet, der repräsentativ ist für den Datenwert, der in der Zelle gespeichert ist, und verstärkt das sich ergebende Signal auf einen Logikpegel. Vor einer LESE-Operation werden die Bitleitungen, die mit einer bestimmten Speicherspalte assoziiert sind, auf einen gewählten Wert vorgeladen. Wenn eine bestimmte Speicherzelle aktiviert wird, werden eine bestimmte Zeile, in der sich die Speicherzelle befindet, und ein Leseverstärker, der mit der bestimmten Spalte assoziiert ist, ausgewählt. Die Ladung in einer dieser Bitleitungen, die mit der Speicherzelle assoziiert sind, wird durch die aktivierte Speicherzelle entladen, und zwar in einer Art und Weise, die dem Wert der Daten entspricht, die in der Speicherzelle gespeichert sind. Dies erzeugt ein Ungleichgewicht zwischen den Signalen in den gepaarten Bitleitungen, wodurch ein Bitleitungs-Spannungshub erzeugt wird.

[0087] Wenn der Leseverstärker aktiviert ist, erfasst er das unsymmetrische Signal, und im Ansprechen darauf ändert sich der für Gewöhnlich symmetrisch belastete Leseverstärkerzustand in einen Zustand, der repräsentativ für den Wert der Daten ist. Diese Zustandserfassung und das Ansprechen darauf treten in einer endlichen Periode auf, während der ein spezieller Betrag an Energie abgegeben wird. In einem Ausführungsbeispiel geben Leseverstärker des Latch-Typs Energie nur während der Aktivierung ab, bis der Leseverstärker die Daten auflöst. Energie wird abgegeben, während sich die Spannung in den Bitleitungen entwickelt. Je größer das Spannungsabklingen in den vorgeladenen Bitleitungen ist, desto mehr Energie wird während der LESE-Operation abgegeben.

[0088] Es wird in Betracht gezogen, dass die Verwendung von Leseverstärkern, die automatisch abschalten, wenn eine Abfühloperation beendet ist, die Energie reduzieren kann. Ein selbstsperrender Leseverstärker schaltet sich zum Beispiel ab, sobald der Leseverstärker den abgefühlten Datenzustand anzeigt. Leseverstärker des Latch-Typs benötigen ein Aktivierungssignal, das in einem Ausführungsbeispiel von einer Schein-Spaltenzeitschaltung erzeugt wird. Der Leseverstärker steuert ein Signal eines begrenzten Hubs aus den globalen Bitleitungen heraus, um Energie zu sparen.

REDUNDANZ

[0089] Speicherdesigner balancieren typischerweise Energie- und Vorrichtungsflächenerwägungen gegen Geschwindigkeit aus. Hochleistungsspeicherkomponenten stellen eine ernsthafte Belastung für die Energie- und Flächen-Budgets von assoziierten Systemen dar, vor allen dann, wenn solche Komponenten innerhalb eines VLSI-Systems wie etwa einem digitalen Signalverarbeitungssystem eingebettet sind. Deshalb ist es äußerst wünschenswert, Speichersubsysteme bereitzustellen, die schnell, aber dennoch energie- und flächeneffizient sind.

[0090] In einem hohen Grade integrierte Hochleistungskomponenten erfordern komplexe Fabrikations- und Verarbeitungsprozesse. Diese Prozesse können unvermeidbare Parameterschwankungen erfahren, die bei den Einheiten, die erzeugt werden, zwangsweise unerwünschte physikalische Defekte erzeugen können oder Designschwächen ausnützen können, und zwar bis zu dem Ausmaß, dass sie die betroffenen Einheiten unbrauchbar machen oder minderwertig machen.

[0091] In einer Speicherstruktur kann die Redundanz wichtig sein, weil ein Fabrikationsfehler oder ein Betriebsausfall sogar einer einzigen Bitzelle zum Beispiel zu dem Versagen des Systems führen kann, das sich auf diesen Speicher verlässt. In ähnlicher Weise können prozessinvariante Merkmale benötigt werden, um zu gewährleisten, dass die internen Operationen der Struktur genauen Zeitsteuerungs- und parametrischen Spezifikationen entsprechen. Wenn Redundanz und Prozessinvarianzmerkmale fehlen, sind die tatsächlichen Herstellungserträge für einen bestimmten Speicher insbesondere dann inakzeptabel, wenn diese in komplexeren Systemen eingebettet sind, die inhärent mehr Fabrikations- und Verarbeitungsschwächen aufweisen. Ein höherer Herstellungsertrag lässt sich in niedrigere Kosten pro Einheit übertragen, während sich ein robustes Design in zuverlässige Produkte übertragen lässt, die niedrigere Betriebskosten aufweisen. Somit ist es äußerst wünschenswert, Komponenten zu entwerfen, die Redun-

danz und prozessinvariante Merkmale aufweisen, wo auch immer dies möglich ist.

[0092] Redundanzvorrichtungen und -techniken bilden andere bestimmte bevorzugte Ausführungsformen der hier beschriebenen Erfindung, die allein oder zusammen die Funktionalität der hierarchischen Speicherstruktur verbessern. Die vorher diskutierten Redundanzaspekte der vorliegenden Erfindung können dazu führen, dass die hierarchische Speicherstruktur weniger anfällig ist für die Disqualifizierung aufgrund von Defekten während der Herstellung oder des Betriebs, wodurch vorteilhafterweise ein Speicherprodukt bereitgestellt wird, das sofort leichter herzustellen ist und kosteneffizienter ist und beim Betrieb robuster ist.

[0093] Redundanz innerhalb eines hierarchischen Speichermoduls kann dadurch realisiert werden, dass eine oder mehrere redundante Zeilen, Spalten oder beides der Basismodulstruktur hinzugefügt werden. Darüber hinaus kann eine Speicherstruktur, die sich aus hierarchischen Speichermodulen zusammensetzt, eine oder mehrere redundante Module für das Mapping in schadhafte Speicherschaltungen verwenden. Ein redundantes Modul kann eine Eins-zu-Eins-Ersetzung eines schadhaften Moduls bereitstellen, oder es kann eine oder mehrere Speicherzellenschaltungen für ein oder mehrere primäre Speichermodule bereitstellen.

SPEICHERMODUL MIT HIERARCHISCHER FUNKTIONALITÄT

[0094] Die modulare hierarchische Speicherarchitektur gemäß einem Ausführungsbeispiel der vorliegenden Erfindung stellt ein kompaktes, robustes, energieeffizientes Hochleistungs-Speichersystem bereit, das vorteilhafterweise eine flexible und extensiv skalierbare Architektur aufweist. Die hierarchische Speicherstruktur setzt sich zusammen aus fundamentalen Speichermodulen oder -blöcken, die kooperativ gekoppelt sein können und in mehreren hierarchischen Stufen angeordnet sein können, um ein zusammengesetztes Speicherprodukt zu bewerkstelligen, das eine willkürliche Spaltenhöhe oder Zeilenlänge aufweist. Dieser modulare Bottom-Up-Ansatz legt Zeitsteuerungserwägungen, das Entscheidungstreffen und den Stromverbrauch für die spezielle(n) Einheiten) fest, in der/denen die gewünschten Daten gespeichert sind.

[0095] Innerhalb einer definierten Designhierarchie können die fundamentalen Speichersubsysteme oder -blöcke gruppiert werden, um eine größere Speicherstruktur zu bilden, die wiederum selbst mit ähnlichen Speicherstrukturen gekoppelt sein kann, um noch größere Speicherstrukturen zu bilden. Diese größeren Strukturen können wiederum so angeordnet werden, dass sie eine komplexe Struktur, die ein SRAM-Modul umfasst, auf der höchsten Stufe der Hierarchie erschaffen. Bei dem hierarchischen Abfühlen ist es erwünscht, zwei oder mehr Stufen der Bitabfühlung bereitzustellen, wodurch die LESE- und SCHREIB-Zeit der Vorrichtung verringert wird, d.h. die effektive Vorrichtungsgeschwindigkeit gesteigert wird, während die gesamten Vorrichtungsenergieanforderungen reduziert werden. In einem hierarchischen Design werden das Umschalten und der Speicherzellen-Stromverbrauch während einer LESE-/SCHREIB-Operation auf die direkte Nachbarschaft der Speicherzellen beschränkt, die ausgewertet werden oder in die geschrieben wird, d.h., auf die Speicherzellen in ausgewählten Speichersubsystemen oder -blöcken, mit der Ausnahme einer begrenzten Anzahl an globalen Wortleitungsselektoren, Leseverstärkern und Unterstützungsschaltungen. Die Mehrzahl der Subsysteme oder Blöcke, die nicht die Speicherzellen enthalten, die ausgewertet werden oder in die geschrieben wird, bleiben im Allgemeinen inaktiv.

[0096] Alternative Ausführungsbeispiele der vorliegenden Erfindung stellen ein hierarchisches Speichermodul bereit, das die lokale Bitleitungsabfühlung, die lokale Bitleitungsdecodierung oder beides verwendet, was intrinsisch den gesamten Stromverbrauch und die Signalausbreitung reduziert und die Gesamtgeschwindigkeit erhöht, sowie auch die Designflexibilität und die Skalierbarkeit erhöht. Ausführungsformen der vorliegenden Erfindung ziehen Vorrichtungen und Verfahren in Betracht, die die gesamte Energieabgabe der hierarchischen Speicherstruktur begrenzen, während sie den Einfluss einer mehrstufigen Hierarchie minimieren. Gewisse Ausführungsformen der vorliegenden Erfindung sind auf die Abschwächung funktioneller Verletzbarkeiten ausgerichtet, die sich aus Schwankungen der Betriebsparameter entwickeln können oder die sich auf den Fabrikationsprozess beziehen.

HIERARCHISCHE SPEICHERMODULE

[0097] In Speicherentwürfen aus dem Stand der Technik, wie etwa die oben erwähnten Designs mit Bänken, werden große logische Speicherblöcke in kleinere physikalische Module aufgeteilt, wobei jedes den begleitenden Overhead eines gesamten Speicherblocks aufweist, der Vordecodierer, Leseverstärker, Multiplexer und dergleichen umfasst. Insgesamt würden sich solche Speicherblöcke wie ein einzelner Speicherblock verhalten. Aber durch die Verwendung der vorliegenden Erfindung können SRAM-Speichermodule einer vergleichbaren oder viel größeren Größe bereitgestellt werden, indem hierarchische Funktionssubsysteme oder -blöcke zu größeren physischen Speichermodulen einer beliebigen Anzahl von Worten und Wortlängen gekoppelt werden. Zum Beispiel benötigen bestehende Designs, die kleinere Speichermodule zu einem einzi-

gen logischen Modul zusammenfassen, für gewöhnlich die Replikation der Vordecodierer, Leseverstärker und anderer Overhead-Schaltungen, die mit einem einzigen Speichermodul assoziiert sein würden.

[0098] Gemäß der vorliegenden Erfindung ist diese Replikation nicht notwendig und unerwünscht. Ein Ausführungsbeispiel der vorliegenden Erfindung umfasst die lokale Bitleitungsabfühlung, bei der eine begrenzte Anzahl von Speicherzellen mit einem einzigen lokalen Leseverstärker gekoppelt ist, wodurch ein Basisspeichermodul gebildet wird. Ähnliche Speichermodule werden gruppiert und angeordnet, um Blöcke zu bilden, die zusammen mit der entsprechenden Schaltung das lokale Leseverstärkersignal an den globalen Leseverstärker ausgeben. Somit sind die Bitleitungen, die mit den Speicherzellen in dem Block assoziiert sind, nicht direkt mit einem globalen Leseverstärker gekoppelt, wodurch die Signalausbreitungsverzögerung und der Stromverbrauch, die typischerweise mit einer globalen Bitleitungsabfühlung verbunden sind, abgeschwächt werden. Bei diesem Lösungsweg fühlt der lokale Bitleitungs-Leseverstärker den Zustand einer ausgewählten Speicherzelle in einem Block schnell und wirtschaftlich ab und berichtet den Zustand dem globalen Leseverstärker.

[0099] In einem anderen Ausführungsbeispiel der hier beschriebenen Erfindung wird ein Speicherblock, eine begrenzte Anzahl an Speicherzellen, neben anderen Einheiten bereitgestellt. Die Verwendung einer lokalen Wortleitungsdecodierung schwächt die Verzögerungen und den Stromverbrauch der globalen Wortleitungsdecodierung ab. Ähnlich wie bei dem lokalen Bitleitungsabfühllösungsweg kann ein einziger globaler Wortleitungsdecoder mit den jeweiligen lokalen Wortleitungsdecodern der vielfachen Blöcke gekoppelt sein. Wenn der globale Decoder mit einer Adresse aktiviert wird, antwortet nur der lokale Wortleitungsdecoder, der mit der gewünschten Speicherzelle eines gewünschten Blocks assoziiert ist, wobei er die Speicherzelle aktiviert. Diese Ausführungsform ist ebenfalls insbesondere stromsparend und schnell, weil das Laden in der globalen Leitung auf die zugehörigen lokalen Wortleitungsdecoder begrenzt ist, und das globale Wortleitungssignal muss nur solange vorhanden sein, wie dies benötigt wird, um die relevante lokale Wortleitung zu aktivieren. In noch einem anderen Ausführungsbeispiel der vorliegenden Erfindung ist ein hierarchischer Speicherblock, der sowohl die lokale Bitleitungsabfühlung als auch die lokale Wortleitungsdecodierung verwendet, bereitgestellt, der die Vorteile beider Ausführungsformen realisiert. Jedes der oben genannten Ausführungsbeispiele wird unten neben anderen erläutert.

SYNCHRONES GESTEUERTES SELBSTGETAKTETES SRAM

[0100] Ein Ausführungsbeispiel eines 0,13 µm SRAM-Moduls, das allgemein mit **300** bezeichnet ist, ist in den Fig. 3A und Fig. 3B veranschaulicht. Es sollte klar sein, dass, obwohl ein 0,13 µm SRAM-Modul veranschaulicht ist, andere SRAM-Module mit anderen Größen in Betracht gezogen werden. Das veranschaulichte SRAM-Ausführungsbeispiel umfasst einen hierarchischen Speicher, der einen großen Speicher in eine zweidimensionale Array von Blöcken aufspaltet. In diesem Ausführungsbeispiel wird eine Zeile von Blöcken als ein Zeilenblock bezeichnet, während eine Spalte von Blöcken als ein Spaltenblock bezeichnet wird. Ein Paar von benachbarten Zeilenblöcken **302** und Spaltenblöcken **304** ist veranschaulicht.

[0101] Es sollte klar sein, dass die Begriffe Zeilenblöcke und Blockspalten willkürliche Bezeichnungen sind, die zugeordnet worden sind, um die Blöcke, die sich in einer Richtung erstrecken, von den Blöcken zu unterscheiden, die sich senkrecht dazu erstrecken, und dass diese Begriffe unabhängig von der Ausrichtung des SRAM **300** sind. Es sollte ebenso klar sein, dass zwar vier Blöcke dargestellt sind, aber jede Anzahl von Spalten- und Zeilenblöcken in Betracht gezogen werden kann. Die Anzahl von Blöcken in einem Zeilenblock kann im Allgemeinen in einem Bereich von irgendwo zwischen 1 bis 16 liegen, während die Anzahl von Blöcken in einem Spaltenblock allgemein in einem Bereich von irgendwo zwischen 1 bis 16 liegen kann, obwohl größere Zeilen- und Spaltenblöcke in Betracht kommen.

[0102] In einem Ausführungsbeispiel umfasst ein Block **306** wenigstens vier Entitäten: (1) eine oder mehrere Zell-Arrays **308**; (2) einen oder mehrere lokale Decoder **310** (die alternativ als "LxDEC **710**" bezeichnet werden); (3) einen oder mehrere lokale Leseverstärker **312** (die alternativ als "LSA **712**" bezeichnet werden); und (4) einen oder mehrere lokale Controller **314** (die alternativ als "LxCTRL **714**" bezeichnet werden). In einem alternativen Ausführungsbeispiel kann der Block **306** Cluster umfassen, wie dies unten beschrieben wird.

[0103] Das SRAM **300**, das in den Fig. 3A und Fig. 3B veranschaulicht ist, umfasst zwei lokale Vordecodierer **316** (die alternativ als "LxPRED" bezeichnet werden), drei globale Decoder **318** (die alternativ als "GxDEC" bezeichnet werden), einen globalen Vordecodierer **320** (der alternativ als "GxPRED" bezeichnet wird), zwei globale Controller **322** (die alternativ als "GxCTR" bezeichnet werden) und zwei globale Leseverstärker **324** (die alternativ als "GSA **724**" bezeichnet werden) zusätzlich zu dem veranschaulichten Block **306**, der acht Zell-Arrays **308**, sechs lokale Decoder **310**, acht lokale Leseverstärker **312**

und zwei lokale Controller **314** umfasst. Es sollte klar sein, dass ein Ausführungsbeispiel einen lokalen Leseverstärker (und in einem Ausführungsbeispiel einen 4:1-Mux) für jeweils vier Spalten der Speicherzelle umfasst, jeder veranschaulichte globale Controller eine Vielzahl von globalen Controllern umfasst, nämlich einen globalen Controller für jeden lokalen Controller, und jeder veranschaulichte lokale Controller eine Vielzahl von lokalen Controllern umfasst, nämlich einen für jede Zeile der Speicherzellen.

[0104] Ein alternatives Ausführungsbeispiel von Block **306**, der nur vier Zell-Arrays **308**, zwei lokale Decoder **310**, zwei lokale Leseverstärker **312** und einen lokalen Controller **314** umfasst, ist in Fig. 4 veranschaulicht. Typischerweise befinden sich die Blöcke in einem Größenbereich von etwa 2 Kbit bis zu etwa 150 Kbit.

[0105] In einem Ausführungsbeispiel können die Blöcke **306** weiter in kleinere Entitäten aufgeteilt werden. Ein Ausführungsbeispiel umfasst eine Array von Leseverstärkern, die in der Mitte der Zell-Arrays **308** angeordnet ist, wodurch sie die Zell-Arrays in obere und untere Subblöcke unterteilen, wie dies unten erläutert wird.

[0106] Es wird in Betracht gezogen, dass in einem Ausführungsbeispiel die externen Signale, die jeden Block **300** steuern, alle synchron sind. Das heißt, die Impulsdauer der Steuersignale ist gleich der Takt-Hoch-Periode (clock high period) des SRAM-Moduls. Des Weiteren ist die interne Zeitsteuerung jedes Blocks **300** selbstgetaktet. Mit anderen Worten, die Impulsdauer der Signale ist abhängig von einer Bitleitungs-Abklingzeit und ist unabhängig von der Taktperiode. Diese Konfiguration ist global robust gegenüber RC-Effekten, lokal schnell und energieeffizient, wie unten bereitgestellt wird.

SPEICHERZELLE

[0107] In einem Ausführungsbeispiel umfassen die Zell-Arrays **308** des SRAM **300** eine Vielzahl von Speicherzellen, wie dies in Fig. 5 veranschaulicht ist, wobei die Größe der Array (gemessen in Zelleinheiten) durch Zeilen mal Spalten bestimmt wird. Zum Beispiel umfasst eine Megabit-Speicherzellen-Array 1024 × 1024 Speicherzellen. Ein Ausführungsbeispiel einer Speicherzelle, die in der SRAM-Zell-Array verwendet wird, umfasst eine Sechs-Transistor-CMOS-Zelle **600A** (die alternativ als "6T-Zelle" bezeichnet wird), die in Fig. 6A veranschaulicht ist. In dem veranschaulichten Ausführungsbeispiel umfasst die 6T-Zelle **600** die Transistoren **601a**, **601b**, **601c** und **601d**.

[0108] Jede 6T-Zelle **600** ist mit einer lokalen Wortleitung **626** (die alternativ als lwlH bezeichnet wird) verbunden, die von allen anderen 6T-Zellen in der gleichen Zeile einer Zell-Array gemeinsam genutzt wird. Ein Paar von lokalen Bitleitungen, die als "bit" und "bit-n" bezeichnet werden und jeweils mit den Nummern **628** und **630** versehen sind, werden von allen anderen 6T-Zellen **600** in der gleichen Spalte in der Zell-Array gemeinsam genutzt. In einem Ausführungsbeispiel betritt das lokale Wortleitungssignal jede 6T-Zelle **600** direkt in einer Poly-Leitung, die das Gate der Zellzugriffstransistoren **632** und **630** bildet, wie dies veranschaulicht ist. Eine Überbrückungsdraht-Metallleitung (jumper metal line) überträgt auch das gleiche lokale Wortleitungssignal. Die Überbrückungsdraht-Metallleitung ist mit der Poly-Leitung in Drahtbrückenzellen kurzgeschlossen, die periodisch jeweils zwischen 16 oder 32 Spalten von 6T-Zellen **600** eingefügt werden. Das Poly in den Drahtbrückenzellen ist hoch widerstandsfähig und ist in einem Ausführungsbeispiel der vorliegenden Erfindung von einem Metall-Überbrückungsdraht nebengeschlossen bzw. geshuntet, um den Widerstand herabzusetzen.

[0109] Im Allgemeinen existiert die 6T-Zelle **600** in einem von drei möglichen Zuständen: (1) dem STABILEN Zustand, in dem die 6T-Zelle **600** einen Signalwert hält, der einer logischen "1" oder einer logischen "0" entspricht; (2) einem LESE-Operations-Zustand; oder (3) einem SCHREIB-Operations-Zustand. In dem STABILEN Zustand ist die 6T-Zelle effektiv von dem Magnetkern (z.B. dem Kern **102** in Fig. 1) getrennt. In einem Beispiel werden die Bitleitungen, d.h., jeweils die Leitungen bit und bit_n **628**, **630**, auf HIGH (HOCH) (logisch "1") vorgeladen, bevor irgendeine LESE- oder SCHREIB-Operation stattfindet. Die Zeilenauswahltransistoren **632**, **634** werden während des Vorladens ausgeschaltet. Der lokale Leseverstärkerblock (der zwar nicht gezeigt ist, aber dem LSA **712** ähnlich ist) ist mit der bit-Leitung **628** und der bit_n-Leitung **630** verbunden, ähnlich wie der LSA **712** in den Fig. 3A, Fig. 3B und Fig. 4, um einen Vorladestrom zu liefern.

[0110] Eine LESE-Operation wird initiiert, indem ein VORLADE-Zyklus durchgeführt wird, der die Leitung bit **628** und die Leitung bit_n **630** auf logisch HIGH (HOCH) vorlädt und LwLH **626** unter Verwendung der Zeilenauswahltransistoren **632**, **634** aktiviert. Eine der Bitleitungen entlädt sich durch die 6T-Zelle **600**, und eine Differenzspannung wird zwischen der Leitung bit **628** und der Leitung bit_n **630** errichtet. Diese Spannung wird abgefühlt und auf Logikpegel verstärkt.

[0111] Eine SCHREIB-Operation in die 6T-Zelle **600** wird nach einem weiteren VORLADE-Zyklus durchgeführt, indem die Bitleitungen **628**, **630** auf den benötigten Zustand gesteuert werden, der dem Schreiben von Daten und dem Aktivieren von lwlH **626** entspricht. CMOS ist eine wünschenswerte Technologie, weil der zugeführte Strom, der von einer solchen

SRAM-Zelle abgezogen wird, typischerweise auf den Leckstrom der Transistoren **601a**–d beschränkt ist, während sie sich in dem STABILEN Zustand befindet.

[0112] Fig. 6B veranschaulicht eine alternative Darstellung der 6T-Zelle, die in Fig. 6A veranschaulicht ist. In diesem Ausführungsbeispiel sind die Transistoren **601a**, **601b**, **601c** und **601d** jeweils als antiparallelgeschaltete Inverter **636** und **638** dargestellt, wie dies veranschaulicht ist.

LOKALER DECODER

[0113] Ein Blockdiagramm eines Ausführungsbeispiels eines SRAM-Moduls **700**, das dem SRAM-Modul **300** der Fig. 3A, Fig. 3B und Fig. 4 ähnlich ist, ist in Fig. 7 veranschaulicht. Dieses Ausführungsbeispiel umfasst eine eindimensionale Array von lokalen x-Decodern oder Lx-DEC **710**, die dem LxDEC **310** ähnlich sind. Die Lx-DEC-**710**-Array ist physisch als eine vertikale Array von lokalen x-Decodern angeordnet, die sich in der Nähe der Zell-Array **708** befindet. Der LxDEC **710** ist mit einem globalen Decoder oder GxDEC **718** verbunden oder ist kommunikativ damit gekoppelt.

[0114] In einem Ausführungsbeispiel befindet sich der LxDEC **710** auf der linken Seite der Zell-Array **708**. Es sollte klar sein, dass die Begriffe "links" oder "rechts", "nach oben" oder "nach unten", "oberhalb" oder "unterhalb" willkürliche Bezeichnungen sind, die zugewiesen worden sind, um die Einheiten, die sich in einer Richtung erstrecken, von den Einheiten unterscheiden zu können, die sich in die andere Richtung erstrecken, und dass diese Begriffe unabhängig von der Ausrichtung des SRAM **700** sind. In diesem Ausführungsbeispiel befindet sich der LxDEC **710** in einer Eins-zu-Eins-Korrespondenz zu einer Zeile der Zell-Array **708**. Der LxDEC **710** aktiviert eine entsprechende lokale Wortleitung oder lwlH **726** eines Blocks, die nicht gezeigt ist. Der LxDEC **710** wird zum Beispiel von WIH-, bnkL- und BitR-**742**-Signalen in ihren jeweiligen Leitungen gesteuert.

[0115] Ein anderes Ausführungsbeispiel des Lx-DEC **710** ist in Fig. 8 veranschaulicht. In diesem Ausführungsbeispiel ist jeder LxDEC **710** in einem Block mit einer eindeutigen globalen Wortleitung **750** (die alternativ als "WIH" bezeichnet wird) verbunden, die der Speicherzeile entspricht. Die globale WIH **750** wird von anderen entsprechenden LxDECs **710** in dem gleichen Zeilenblock unter Verwendung von lwlH **750** gemeinsam genutzt. Der LxDEC **710** aktiviert die lokale Wortleitung **726** nur dann, wenn die entsprechende globale Wortleitung **750** aktiviert ist. Es sollte klar sein, dass eine Vielzahl von Zellen **754**, die den 6T-Zellen ähnlich sind, die vorher erläutert worden sind, kommunikativ mit der lwlH **726** gekoppelt sind, wie dies veranschaulicht ist.

[0116] In dem Ausführungsbeispiel, das in Fig. 8 veranschaulicht ist, nutzt jeder LxDEC **710** ganz oben oder ganz unten in einem Subblock die gleiche Bankleitung (die alternativ als "bnk Sol H" bezeichnet wird) gemeinsam. Es sollte klar sein, dass es jeweils separate Leitungen bnkL_bot **756** und bnkL_top **758** für die unteren und oberen Subblöcke gibt. Der Lx-DEC **710** wird lwlH **726** nur dann aktivieren, wenn diese Leitung aktiv ist. Die Bankleitungen werden verwendet, um selektiv unterschiedliche Blöcke innerhalb des gleichen Zeilenblocks zu aktivieren und die richtige Zugriffszeitsteuerung zu synchronisieren. Zum Beispiel wird die Bankleitung während einer LESE-Operation so früh wie möglich aktiviert werden, um die Leseoperation zu beginnen. Während einer SCHREIB-Operation zum Beispiel wird bnkL mit der Verfügbarkeit der Daten in den lokalen Bitleitungen synchronisiert.

[0117] Jeder LxDEC **710** in dem Ausführungsbeispiel, das in Fig. 8 veranschaulicht ist, nutzt die gleiche bitR-Leitung **760**. Diese Leitung wird in dem Speicherruhezustand auf VDD vorgeladen. Wenn sich bitR **760** VDD/2 nähert (d.h., einer Hälfte von VDD), signalisiert dies das Ende eines Speicherzugriffs und bewirkt, dass der LxDEC **710** lwlH **726** deaktiviert. Die bitR-Signalleitung **760** ist in der Zell-Array als eine Replik der Bitleitungen aufgebaut (d.h., in diesem Ausführungsbeispiel sind die bit-Leitung **728** und die bit n-Leitung **730** gleich der bit-Leitung **628** und der bit_n-Leitung **630**, die weiter oben diskutiert worden sind), so dass die kapazitive Belastung der bitR-**760**-Leitung die gleiche pro Einheitslänge ist wie in der Zell-Array. In einem Ausführungsbeispiel aktiviert ein lokaler Replik-Decoder, gesteuert von bnkL, das lwlRH-Signal. In diesem Ausführungsbeispiel ist lwlRH ein Synchronisiersignal, das den lokalen Controller steuert. Das lwlRH kann jedes Mal dann ausgelöst werden, wenn auf einen assoziierten Subblock (der einem wlRH entspricht) zugegriffen wird.

[0118] In einem Ausführungsbeispiel initiiert oder überträgt ein globaler Controller ein LESE- oder SCHREIB-Signal. Der assoziierte lokale Controller **714** initiiert oder überträgt ein entsprechendes Signal auf der Basis des Signals, das von dem globalen Controller übertragen worden ist (nicht gezeigt). Der lokale Controller unterzieht die bitR-Leitung **760** vom LxDEC **710** einem Pull-Down, wenn aus der richtigen Zelle GELESEN wird oder in diese GESCHRIEBEN wird, um Strom zu sparen. Wenn der Unterschied zwischen der bit-Leitung **728** und der bit_n-Leitung **730** hoch genug ist, um den Leseverstärkerabschnitt zu aktivieren, wird lwlH **726** ausgeschaltet, um Strom zu sparen. Ein Schaltbild eines Ausführungsbeispiels eines lokalen x-Decoders, der dem LxDEC **710** ähnlich ist, ist in Fig. 9 veranschaulicht.

LOKALER LESEVERSTÄRKER

[0119] Ein Ausführungsbeispiel des SRAM-Moduls umfasst eine eindimensionale Array von lokalen Leseverstärkern oder LSAs **712**, die in den Fig. 10 und Fig. 11 veranschaulicht sind, in denen die Ausgänge des LSA **712** mit dem GSA **724** über Leitungen **762** gekoppelt sind. In einem Ausführungsbeispiel sind die Ausgänge der LSAs mit dem GSA über wenigstens ein Paar von gbit- und gbit_n-Leitungen gekoppelt. Fig. 12A veranschaulicht ein Ausführungsbeispiel eines LSA **712**, der einen zentralen differentiellen kreuzgekoppelten Verstärkerkern **764** umfasst, der zwei Inverter **764A** und **764B** umfasst. Die senseH-Leitungen **766** und clusterL **798** sind mit dem Verstärkerkern durch den Transistor **771** gekoppelt.

[0120] Die LSAs **764** sind mit einem oder mehreren 4:1-Multiplexern **772** und acht Paaren von muxL-Leitungen **768A**, vier muxLs **768A**, die sich oberhalb, und vier **768B** (am besten in Fig. 7 zu sehen), die sich unterhalb des Verstärkerkerns **764** befinden, gekoppelt. In dem veranschaulichten Ausführungsbeispiel verbindet jeder der Bitleitungs-Multiplexer **772** ein entsprechendes Bitleitungspaar mit dem Verstärkerkern **764**. Die Leitungen gbit und gbit_n sind mit dem Verstärkerkern durch einen PMOS-Transistor (zum Beispiel den Transistor **770**) verbunden. Wenn ein Bitleitungspaar von dem Verstärkerkern **764** abgekoppelt wird, gleicht der Bitleitungs-Multiplexer **772** aktiv das Bitleitungspaar aus und lädt es auf VDD.

[0121] Fig. 12B veranschaulicht ein Schaltbild eines Verstärkerkerns **764**, der zwei Inverter **764A** und **764B** aufweist, wobei jeder Inverter **764A** und **764B** mit einer SenseH-Leitung **766** und einer Cluster-Leitung **798** durch einen Transistor NMOS **771** gekoppelt ist. Nur eine der SenseH- und der Cluster-Leitungen ist veranschaulicht. In dem veranschaulichten Ausführungsbeispiel wird jeder der Inverter **764A** und **764B** als gekoppelte PMOS- und NMOS-Transistoren dargestellt, wie dies im Stand der Technik allgemein bekannt ist. Fig. 12C veranschaulicht eine schematische Darstellung des Verstärkerkerns von Fig. 12B (ähnlich dem Verstärkerkern von Fig. 12A).

[0122] In einem Ausführungsbeispiel, das in Fig. 13 veranschaulicht ist, umfasst die Leseverstärker-Array eine horizontale Array von Leseverstärkern **713**, die sich in der Mitte der Zell-Array **708** befindet, wodurch sie die Zell-Array in obere **708A** und untere **708B** Subblöcke unterteilt, wie dies vorher vorgesehen worden ist. In diesem Ausführungsbeispiel ist die Breite eines einzigen LSA **712** viermal so groß wie die Breite der Zell-Array, während die Anzahl an LSA **712**-Instanzen in der Array gleich der Anzahl an Spalten/4 ist. Das heißt, jeder LSA **712** (und in einem Ausführungsbeispiel ein 4:1-Mux) befindet sich in einer Eins-zu-Eins-Korrespondenz mit vier Säulen der Zell-Array und ist mit den entsprechenden lokalen Bitleitungspaaren der Zell-Array **708** in den oberen und unteren Subblöcken **708A**, **708B** verbunden. Diese Anordnung wird als lokales 4:1-Multiplexen bezeichnet (das alternativ als "lokales 4:1-Muxen" bezeichnet wird). Es sollte klar sein, dass die Bitleitungspaare des unteren Subblocks **708B** von dem oberen Subblock **708A** getrennt sind, wodurch die kapazitive Last jeder Bitleitung **729** um einen Faktor von zwei verringert wird, wodurch die Geschwindigkeit der Bitleitung um den gleichen Faktor erhöht wird und die Energie herabgesetzt wird. Ein Ausführungsbeispiel des 4:1-Mux plus Vorladung ist in den Fig. 10 und 12 veranschaulicht und wird unten noch ausführlicher beschrieben werden.

[0123] Es ist gegenwärtig bekannt, Stromschienen **774** (als Phantomzeichnung gezeigt) zwischen die Paare von Bitleitungen zu setzen, um die Bitleitungspaare von den benachbarten Paaren abzuschirmen. Dadurch wird verhindert, dass die Signale in einem Paar von Bitleitungen die benachbarten Bitleitungspaare beeinflussen. In diesem Ausführungsbeispiel werden dann, wenn auf ein Paar von Bitleitungen **729** (bit und bit_n, **728**, **730**) zugegriffen wird, alle benachbarten Bitleitungen durch den 4:1-Mux auf VDD vorgeladen, wie dies in **Fig.** 12 veranschaulicht ist. Das Vorladen der benachbarten Bitleitungen eliminiert die Notwendigkeit von Schilden, um diese Bitleitungen zu isolieren. Dies bedeutet, dass es nicht notwendig ist, Paare von Bitleitungen voneinander zu isolieren, indem dazwischengesetzte Stromschienen **774** verwendet werden. Dies erlaubt einen größeren Bitleitungs-Abstand in der gesamten Breite und deshalb weniger Kapazität, weniger Strom und eine höhere Geschwindigkeit.

[0124] Der LSA **712** ist mit einem Paar von globalen Bitleitungen, die als gbit **776** und gbit_n **778** bezeichnet werden, über PMOS-Transistoren **770** verbunden, wie dies in Fig. 12A veranschaulicht ist. Zwei PMOS-Transistoren sind veranschaulicht, aber jede Anzahl kommt in Betracht. In einem Ausführungsbeispiel verlaufen die globalen Bitleitungen vertikal parallel zu den lokalen Bitleitungen. Die globalen Bitleitungen werden von den entsprechenden lokalen Leseverstärkern **712** in anderen Blöcken in dem gemeinsamen Spaltenblock gemeinsam genutzt. In einem Ausführungsbeispiel werden die lokalen Bitleitungen und die globalen Bitleitungen auf verschiedenen Metallbelägen geleitet. Da es viermal weniger globale Bitleitungen als lokale Bitleitungen gibt, sind die globalen Bitleitungen physisch breiter und in einem größeren Abstand (Pitch) angeordnet. Dies reduziert in signifikanter Weise den Widerstand und die Kapazität der langen globalen Bitleitungen, wodurch die Geschwindigkeit und die Zuverlässigkeit des SRAM-Moduls gesteigert werden. Die PMOS-Transistoren **770** isolieren die globalen Bitleitungen **776**, **778** gegenüber dem Leseverstärker.

[0125] Ein Ausführungsbeispiel des Bitleitungs-Multiplexers oder 4:1-Mux **772** ist in Fig. 14 veranschaulicht. In diesem Ausführungsbeispiel umfasst der 4:1-Mux **772** einen/eine Vorlade- und Ausgleichsabschnitt oder -vorrichtung **773** und zwei Übertragungsgatter pro bit/bit_n-Paar. Genauer gesagt kann das 4:1-Muxen 8 Übertragungsgatter und 4 Vorladungen und Ausgleicher umfassen, obwohl nur 4 Übertragungsgatter und 2 Vorladungen und Ausgleicher veranschaulicht sind.

[0126] In dem veranschaulichten Ausführungsbeispiel umfasst jeder Vorladungs- und Ausgleichabschnitt **773** des 4:1-Mux drei PFet-Transistoren **773A**, **773B** und **773C**. In diesem Ausführungsbeispiel umfasst der Vorladeabschnitt die PFet-Transistoren **773A** und **773B**. Der Ausgleichabschnitt umfasst den PFet-Transistor **773D**.

[0127] In dem veranschaulichten Ausführungsbeispiel umfasst jedes Übertragungsgatter einen NFet-Transistor **777A** und einen PFet-Transistor **777B**. Obwohl eine spezielle Anzahl und Anordnung von PMOS- und NMOS-Transistoren besprochen werden, kommen andere Anzahlen und Anordnungen in Betracht. Der Vorlade- und Ausgleichabschnitt **773** ist so ausgelegt, dass er die Bitleitungen **728**, **739** vorladen und ausgleichen kann, wie dies vorher vorgesehen worden ist. Das Übertragungsgatter **775** ist so ausgelegt, dass es sowohl logische "1"en als auch logische "0"en weiterleiten kann, wie dies im Stand der Technik wohl bekannt ist. Die NFet-Transistoren **777A** und **777B** zum Beispiel können Signale während einer SCHREIB-Operation durchlassen, während die PFet-Transistoren **779A** und **779B** Signale während einer LESE-Operation durchlassen können.

[0128] Die Fig. 15 und Fig. 16 veranschaulichen Ausführungsbeispiele des 2:1-Mux **772**, der mit dem Verstärkerkern **764** des LSA gekoppelt ist. Fig. 15 veranschaulicht auch eine alternative Darstellung des Übertragungsgatters. Hier sind vier Übertragungsgatter **775A**, **775B**, **775C** und **775D** veranschaulicht, die mit den Invertern **764A** und **764B** des Inverterkerns gekoppelt sind. In einem Ausführungsbeispiel der vorliegenden Erfindung sind acht Übertragungsgatter für jeden LSA, also zwei pro Bitleitungspaar in Betracht gezogen.

[0129] Fig. 16 veranschaulicht den Vorlade- und Ausgleichabschnitt **773** des 2:1-Mux, der mit den Übertragungsgatter **775A** und **775B** des Mux **772** gekoppelt ist, der wiederum mit dem Verstärkerkern gekoppelt ist. Obwohl nur ein Vorlade- und Ausgleichabschnitt **773** veranschaulicht ist, wird es in Betracht gezogen, dass ein zweiter Vorlade- und Ausgleichabschnitt **773** mit den Übertragungsgattern **775C** und **775D** gekoppelt ist.

[0130] In einem Ausführungsbeispiel, das in Fig. 7 veranschaulicht ist, wird der LSA **712** von dem folgenden Satz von Leitungen bzw. Signalen in diesen Leitungen gesteuert, die quer durch die gesamte LSA **712**-Array gemeinsam genutzt werden: (1) muxL_bot **768B**; (2) muxL_top **768A**; (3) senseH **766**; (4) genL **780**; und (5) lwlRH **782**. In einem Ausführungsbeispiel des SRAM-Moduls wählt der LSA **712** aus, welche der lokalen Bitleitungen dazu verwendet wird, die Zell-Array **708** zu initiieren oder auf diese zuzugreifen. Die lokalen Bitleitungen umfassen 8 Paare von Leitungen, 4 Paare von Mux-Leitungen **768B**, die mit dem unteren Subblock **708B** verbunden sind (die alternativ als "muxL_bot **765B**<0:3>" bezeichnet werden), und 4 Paare von Mux-Leitungen **768A**, die mit dem oberen Subblock **708A** verbunden sind (die alternativ als "muxL_top **765A**<0:3>" bezeichnet werden). Der LSA **712** wählt aus, welches der 8 Paare von lokalen Bitleitungen für den aktuellen Zugriff verwendet werden soll. Der LSA **712** hält jede lokale Bitleitung, die nicht für den Zugriff ausgewählt wird, in einem vorgeladenen und ausgeglichenen Zustand. In einem Ausführungsbeispiel hält der LSA **712** die nicht ausgewählten Bitleitungen so, dass sie auf VDD vorgeladen sind.

[0131] Der LSA **712** aktiviert auch den Verstärkerabschnitt des Leseverstärkers **713**, wobei er eine sense enable-Leitung **766** oder ein Signal in der Leitung verwendet (was alternativ als "senseH **766**" bezeichnet wird), die mit dem Transistor **773** verbunden ist. Dieses Aktivierungssignal wird in vier separate Signale verteilt, wobei jedes Signal einen von jeweils vier lokalen Leseverstärkern antippt. In einem Ausführungsbeispiel kann der lokale Controller **714** alle senseH-Leitungen **766** gleichzeitig aktivieren (was als "globales 1:1-Multiplexing" oder "globales 1:1 Muxen" bezeichnet wird), weil jeder Leseverstärker **713** durch die senseH-Leitungen **766** für jeden Zugriff aktiviert wird. Alternativ dazu kann der lokale Controller die senseH-Leitungen **766** auch paarweise aktivieren (was als "globales 2:1-Multiplexing" oder "globales 2:1-Muxen" bezeichnet wird), weil jeder zweite Leseverstärker **713** von den senseH-Leitungen **766** für jeden Zugriff aktiviert wird. Außerdem kann der LSA **712** die senseH-Leitungen **766** einzeln aktivieren (was als "globales 4:1-Multiplexen" oder "globales 4:1-Muxen" bezeichnet wird), weil jeder vierte Leseverstärker für jeden Zugriff aktiviert wird. Es sollte klar sein, dass das Verbinden der senseH-Leitungen **766** mit jedem vierten aktivierten Transistor bei dem globalen 4:1-Multiplexen mehr konfigurierbare Anordnungen für unterschiedliche Speichergrößen bereitstellt.

[0132] Der LSA **712** setzt in einem Ausführungsbeispiel die Leseverstärker **713** den globalen Bitleitungen aus. Der LSA **712** aktiviert oder initiiert die genL-Leitung **780**, wodurch er die Leseverstärker **713** den gbit- und gbit_n-Leitungen aussetzt.

[0133] In einem Ausführungsbeispiel repliziert der LSA **712** die lokalen Poly-Wortleitungen, die durch jede Zeile jedes Blocks laufen. Diese replizierte Leitung wird als Schein-Poly-Leitung **782** bezeichnet (was alternativ als "lwlRH **782**" bezeichnet wird). In diesem Ausführungsbeispiel bildet die lwlRH-Leitung **782** das Gate von Schein-Transistoren, die jede Spalte der Zell-Array **708** beenden. Jeder Schein-Transistor repliziert den Zugriffstransistor der 6T-SRAM-Zelle. Die kapazitive Belastung dieser Leitung wird dazu verwendet, die Zeitsteuerungscharakteristiken einer tatsächlichen lokalen Wortleitung zu replizieren.

[0134] Es wird in Betracht gezogen, dass sich in einem Ausführungsbeispiel die replizierte lwlRH-Leitung **782** auch auf die Metall-Überbrückungsdraht-Leitung (nicht gezeigt) ausweitet. Die replizierte Überbrückungsdraht-Leitung weist die gleiche Breite und die gleiche Nachbarmetallbeabstandung auf wie jeder lokale Wortleitungs-Überbrückungsdraht in der Zell-Array. Diese Leitung wird von dem lokalen Controller **714** lediglich als eine kapazitive Last verwendet und beeinträchtigt die Funktion des LSA **712** in keinster Weise. Genauer gesagt ist die Replik-Überbrückungsdraht-Leitung so ausgelegt, dass sie den Widerstand der lwlRH-Poly-Leitung ähnlich der Metall-Shunt-Leitung reduziert, die vorher vorgesehen worden ist. Ein Schaltbild eines Ausführungsbeispiels eines LSA **712** ist in Fig. 17 veranschaulicht.

LOKALER CONTROLLER

[0135] In einem Ausführungsbeispiel weist jeder Block einen einzigen lokalen Controller oder LxCTRL **714** auf, wie dies in den Fig. 7 und Fig. 18 veranschaulicht ist, der die Aktivitäten der lokalen x-Decoder **710** und der Leseverstärker **713** koordiniert. In diesem Ausführungsbeispiel koordiniert der LxCTRL **714** diese Aktivitäten, indem er gewisse Leitungen beeinflusst, die folgende Steuerleitungen umfassen: (1) die bitR-Steuerleitung **760**; (2) die bnkL_bot-Steuerleitung **756**; (3) die bnkL_top-Steuerleitung **758**; (4) die muxL_bot-Steuerleitung **765B**; (5) die muxL_top-Steuerleitung **765A**; (6) die senseH-Steuerleitung **766**; (7) die genL-Steuerleitung **780**; und (8) die lwlRH-Steuerleitung **782**, wie dies in Fig. 7 veranschaulicht ist. Jede dieser Leitungen wird von einem Treiber und der Steuerlogikschaltung in der Lx-CTRL-Schaltung **714** aktiviert. In einem Ausführungsbeispiel sind diese Leitungen alle normalerweise inaktiv, wenn sich das SRAM-Modul in dem Ruhezustand befindet, mit Ausnahme der genL-Leitung **780**. Die genL-Leitung **780** ist in dem Ruhezustand aktiv. Die LxCTRL-Schaltung **714** wird wiederum von externen vertikalen und horizontalen Signalen aktiviert. Vertikale Signale umfassen: (1) das lmuxL-Signal **784**; (2) das gmuxL-Signal **786**; (3) das rbankL-Signal **788**; (4) das gbitR-Signal **760**; und (5) das wbankL-Signal **792**. Die horizontalen Signale umfassen: (1) das wlRH-Signal **794**; (2) das blkSelH_bot-Signal **756**; und (3) das blkSelH_top-Signal **758**.

[0136] In einem Ausführungsbeispiel benutzen alle LxCTRL-**714**-Schaltungen in dem gleichen Spaltenblock die vertikalen Signale gemeinsam. In diesem Ausführungsbeispiel ist der LxCTRL **714** in jedem Block mit vier lokalen Mux-Leitungen **784** verbunden (die alternativ als "lmuxL<0:3>" oder "lmuxl" bezeichnet werden). Nur eine der vier lmuxL-Leitungen **768** ist zu einem Zeitpunkt aktiv. Der LxCTRL **714** initiiert oder aktiviert eine der lmuxL-Leitungen **768**, um auf eine Zell-Array **708** zuzugreifen, wobei er eine der vier Zell-Array-Spalten, die mit jedem LSA **712** verbunden sind, für den Zugriff auswählt.

[0137] In einem Ausführungsbeispiel kann ähnlich, wie dies oben diskutiert worden ist, der LSA **712** die senseH-Signale **766** einzeln aktivieren (d.h., globales 4:1-Multiplexen). In diesem Ausführungsbeispiel ist der LxCTRL **714** in jedem Block mit vier globalen Mux-Leitungen **786** verbunden (die alternativ als "gmuxL<0:3>" oder "gmuxl" bezeichnet werden). Es sollte klar sein, dass nur eine der vier gmuxL-Leitungen **768** zu einem Zeitpunkt aktiv ist, die eine von jeweils vier globalen Bitleitungen für den Zugriff auswählt oder aktiviert. In einem Ausführungsbeispiel aktiviert der LSA **712** die senseH-Leitungen **766** paarweise (d.h., globales 2:1-Multiplexen). In diesem Ausführungsbeispiel sind nur zwei der vier gmuxL-Leitungen **768** zu einem Zeitpunkt aktiv, die eine von jeweils zwei globalen Bitleitungen für den Zugriff auswählen. Bei dem globalen 1:1-Multiplexen sind alle vier gmuxL-Leitungen **786** immer aktiv und wählen alle globalen Bitleitungen für den Zugriff aus.

[0138] Alle LxCTRL-Schaltungen **714** in dem gleichen Spaltenblock nutzen die gleichen Lesebankleitungen **788** oder Signale in den Leitungen gemeinsam (die alternativ als "rbankL" bezeichnet werden). Die rbankL-Leitung **788** wird aktiviert, wenn eine LESE-Operation angefordert wird (d.h., Daten sollen aus dem Block gelesen werden). An dem Ende der LESE-Operation enthalten die globalen Bitleitungen, die von der gmuxL-Leitung **786s 786** ausgewählt wurden, differentielle Signale mit begrenztem Hub. Diese differentiellen Signale mit begrenztem Hub stellen die gespeicherten Werte in den Zellen dar, die von der lwlH-Leitung **726** und den lmuxL-Leitungen **784** ausgewählt worden sind.

[0139] In einem Ausführungsbeispiel wird eine globale Replik-Bitleitung **790** oder ein Signal in der Leitung gemeinsam von allen LxCTRL-Schaltungen **714** in dem gleichen Spaltenblock genutzt (die alternativ als "gbitR" bezeichnet wird). Die gbitR-Leitung **760** wird extern auf VDD gehalten, wenn sich der SRAM-Speicher im Ruhezustand befindet. Die gbi-

tR-Leitung **760** wird fließend gemacht, wenn ein LESE-Zugriff initiiert wird. Der LxCTRL **714** entlädt dieses Signal auf VSS, wenn eine LESE-Zugriffsanforderung synchron zu der Verfügbarkeit von LESE-Daten auf der Leitung gbit/gbit_n beendet ist.

[0140] Während einer SCHREIB-Operation aktiviert der LxCTRL **714** Schreibbankleitungen **729** oder Signale in der Leitung (alternativ als "wbnkL" bezeichnet). Differentielle Signale mit begrenztem Hub sind in den globalen Bitleitungen vorhanden, wenn die wbnkL-Leitung **792** aktiviert ist. Die differentiellen Signale mit begrenztem Hub repräsentieren die Daten, die geschrieben werden sollen.

[0141] Es sollte außerdem klar sein, dass in einem Ausführungsbeispiel alle LxCTRL-Schaltungen **714** in der gleichen Zeilenblockspalte die horizontalen Signale gemeinsam nutzen. In einem Ausführungsbeispiel nutzen alle LxCTRL-**714**-Schaltungen eine Replik der globalen Wortleitung wIH-Leitung **794** gemeinsam (alternativ als "wIRH" bezeichnet), die durch jede Zeile des Speichers läuft. Das physikalische Layout der wIRH-Leitung **794** repliziert die globale Wortleitung in jeder Zeile im Hinblick auf Metallbelag, Tiefe und Abstand. Somit sind die kapazitive Belastung der wIRH-Leitung **794** und das globale wIH-Signal gleich. Bei jedem Speicherzugriff wird die wIRH-Leitung **794** gleichzeitig mit einem einzigen globalen wIH-Signal für eine Zeile in dem Block aktiviert.

[0142] Der LxCTRL **714** zeigt dem Block an, ob auf den unteren oder den oberen Subblock **706B**, **706A** zugegriffen werden soll, und zwar unter Verwendung entweder der Leitungen blkSelH_bot **756** oder blkSelH_top **758** oder der Signale in den Leitungen. Eine dieser Leitungen ist bei jedem Speicherzugriff auf den Block aktiv, was anzeigt, ob die Übertragungsgatter des unteren Subblocks **706B** oder des oberen Subblocks **706A** in dem LSA **712** geöffnet werden sollen. Ein Schaltbild für ein Ausführungsbeispiel des lokalen Controllers ist in **Fig. 19** veranschaulicht.

SYNCHRONE STEUERUNG DES SELBSTGETAKTETEN LOKALEN BLOCKS

[0143] Ein Ausführungsbeispiel der vorliegenden Erfindung umfasst ein oder mehrere globale Elemente oder Vorrichtungen, die synchron gesteuert werden, während ein oder mehrere lokale Elemente asynchron gesteuert werden (was alternativ als "selbstgetaktet" bezeichnet wird). Es sollte klar sein, dass der Begriff synchrone Steuerung bedeutet, dass diese Vorrichtungen mit einem Taktimpuls gesteuert werden oder synchron zu einem Taktimpuls sind, der von einer Taktvorrichtung oder einer anderen externen Zeitsteuervorrichtung bereitgestellt wird. Ein Vorteil des Vorhandenseins einer synchronen Steuerung von Elementen oder Vorrichtungen auf der globalen Ebene liegt darin, dass diese Elemente, die von dem Widerstand beeinflusst werden, eingestellt werden können.

[0144] Zum Beispiel verlangsamt bzw. ändert das Verlangsamen bzw. Ändern des Taktimpulses das synchrone Signal. Das Verlangsamen bzw. Ändern des synchronen Signals verlangsamt bzw. ändert diejenigen Vorrichtungen oder Elemente, die von den synchronen Signalen gesteuert werden, was für eine solche Vorrichtung mehr Zeit bereitstellt, um zu agieren, was sie in die Lage versetzt, ihre vorgeschriebene Funktion komplett durchzuführen. In einem Ausführungsbeispiel ist der globale Controller synchron. In einem anderen Ausführungsbeispiel sind der globale Decoder und die globalen Leseverstärker synchron.

[0145] Alternativ dazu werden die lokalen Vorrichtungen oder Elemente asynchron gesteuert oder sind selbstgetaktet. Die selbstgetakteten Vorrichtungen sind diejenigen Vorrichtungen, bei denen es wenig RC-Effekte gibt. Asynchron gesteuerte Vorrichtungen sind im Allgemeinen schneller und verbrauchen weniger Strom. In einem Ausführungsbeispiel wird der lokale Block, der allgemein den lokalen Controller, den lokalen Decoder, die lokalen Leseverstärker, SenseEnable-High und die Zell-Arrays umfasst, asynchron gesteuert.

TIMING DES LESEZYKLUS

[0146] Die Zeitsteuerung des Zyklus für eine Leseoperation gemäß einem Ausführungsbeispiel der vorliegenden Erfindung umfasst das Übertragen oder Bereitstellen eines hohen (high) Signals durch den globalen Controller und bewirkt, dass die LwIH-Leitung ausgelöst wird und eine oder mehrere Speicherzellen ausgewählt werden. Beim Empfang eines Signals auf der LwIH-Leitung wird eines oder mehrere der Leitungspaare bit/bit_n ausgesetzt und klingen aus (was alternativ als die "Integrationszeit" bezeichnet wird). Zum gleichen Zeitpunkt bzw. etwa um den gleichen Zeitpunkt herum, an dem die Leitungen bit/bit_n beginnen abzuklingen, beginnt bitR abzuklingen (d.h., beim Empfang eines "high"-Signals in der lwIRH-Leitung). Aber die Leitung bitR klingt etwa 5 bis 6 mal schneller ab als die Leitungen bit/bit_n, wodurch die Integration gestoppt wird, bevor die Leitungen bit/bit_n vollständig abklingen (d.h., eine Hubleitungsspannung abfühlen), und initiiert die Verstärkung der Spannung.

[0147] BitR aktiviert eine oder mehrere der SenseH-Leitungen. In Abhängigkeit von dem Multiplexen werden alle vier SenseH-Leitungen (1:1-Muxen), zwei SenseH-Leitungen (2:1-Muxen) oder eine SenseH-Leitung (4:1-Muxen) aktiviert.

[0148] Nachdem das SenseH-Leitungssignal aktiviert ist, löst der Leseverstärker die Daten auf, die globale enable-low- oder genL-Leitung wird aktiviert (d.h., ein niedriges Signal (low signal) wird auf genL übertragen). Das Aktivieren der genL-Leitung setzt den lokalen Leseverstärker den globalen Leitungen bit und bit_n aus. Das genL-Signal startet auch das Abklingen des Signals in der gbitR-Leitung. Wiederum klingt das gbitR-Signal etwa 5- bis 6-mal schneller ab als das gbit-Signal, was den Pull-Down des gbit abschaltet. In einem Ausführungsbeispiel klingt das gbitR-Signal etwa 5 bis 6 mal schneller ab als das gbit-Signal, so dass das Signal in der gbit-Leitung nur um etwa 10% von VDD abklingt, bevor es abgeschaltet wird.

[0149] Das Signal in der Leitung gbitR schaltet das Signal in der SenseH-Leitung aus und aktiviert den globalen Leseverstärker. Mit anderen Worten, das Signal in der gbitR-Leitung schaltet den lokalen Leseverstärker aus, was den Pull-Down in den Leitungen gbit und gbit_n stoppt. In einem Ausführungsbeispiel ist das SenseH-Signal total asynchron.

[0150] Die Zyklus-Zeitsteuerung für eine LESE-Operation, die ein Ausführungsbeispiel der vorliegenden Erfindung verwendet (das dem von Fig. 7 ähnlich ist), ist in Fig. 20 veranschaulicht. Während der LESE-Operation wird eine der vier lmuxL<0:3>-Leitungen **784** aktiviert, wodurch eine der vier Zell-Array-Spalten ausgewählt wird, die von jedem LSA **712** unterstützt werden. Eine, zwei oder vier gmuxL<0:3>-Leitungen **786** werden aktiviert, um jede vierte, jede zweite oder jede globale Bitleitung für den Zugriff auszuwählen, was von der globalen Multiplex-Option abhängt (d.h., 4:1-, 2:1- oder 1:1-Muxen).

[0151] Entweder blkSelH_bot **756** oder blkSelH_top **758** wird aktiviert, um dem Block anzuzeigen, dass jeweils auf den unteren oder den oberen Subblock **706B**, **706A** zugegriffen wird. Die rbankL-Leitung **788** wird aktiviert, um eine Leseoperation aus dem Block anzufordern. Die wIH-Leitung wird für die Speicherzeile aktiviert, auf die zugegriffen wird, während die wIRH-Leitung **793** gleichzeitig für alle Blöcke in dem Zeilenblock aktiviert wird, der die Speicherzeile enthält.

[0152] Der LxCTRL **714** deaktiviert die genL-Leitung **780**, um die lokalen Leseverstärker von den globalen Bitleitungen zu isolieren. Der LxCTRL **714** aktiviert die bnkL-Leitung, um dem LxDEC **710** zu signalisieren, dass er eine lokale Wortleitung aktivieren soll. Der LxCTRL **714** aktiviert eine der vier muxL<0:3>-Leitungen, die dem aktivierten muxL-Signal entspricht. Dies bewirkt, dass der LSA **712** eine der vier Zellenspalten mit dem Leseverstärker-Verstärkerkern **762** verbindet. Der LxDEC **710**, der der aktivierten globalen Wortleitung entspricht, aktiviert die lokale Wortleitung. Gleichzeitig aktiviert der LxCTRL **714** die lwIRH-Leitung **794** **782**. Alle Zellen in der Zeile, die der aktivierten lokalen Wortleitung entspricht, beginnen damit, eine Bitleitung in jedem Bitleitungspaar zu entladen, was dem gespeicherten Wert der 6T-Zelle entspricht.

[0153] Nach einem vorbestimmten Zeitraum hat sich eine ausreichende Spannungsdifferenz quer durch jedes Bitleitungspaar entwickelt. In einem Beispiel ist eine Spannungsdifferenz von etwa 100mV ausreichend. Es sollte klar sein, dass dieser vorbestimmte Zeitraum abhängig ist von den "Process-Corner"-Bedingungen, der Übergangsstellentemperatur, der Stromzufuhr und der Höhe der Zell-Array.

[0154] Gleichzeitig bewirkt das Signal lwIRH **782**, dass der LxCTRL **714** die bitR-Leitung **760** mit einem NMOS-Transistor entlädt, der einen bestimmten Strom mit einem festen Vielfachen des Zellstroms abzieht. Die Leitung bitR **760** entlädt damit mit einer Geschwindigkeit, die proportional zu der Bitleitungs-Entladegeschwindigkeit ist. Es sollte klar sein, dass die Konstante der Proportionalität invariant ist (bis zu einer ersten Ordnung) im Hinblick auf die "Process-Corner"-Bedingungen, die Übergangsstellentemperatur, die Stromzufuhr und die Höhe der Zell-Array **708**.

[0155] Wenn das bitR-Signal **760** einen vorbestimmten Schwellwert überschreitet, deaktiviert der LxDEC **710** die lokale Wortleitung und die 6T-Zellen stoppen das Entladen durch die Bitleitungen. Auf diese Weise wird eine Differenzspannung mit begrenztem Hub quer durch die Bitleitungen unabhängig von den "Process-Corner"-Bedingungen, der Übergangsstellentemperatur, der Stromzufuhr und der Höhe der Zell-Array erzeugt (bis zu einer ersten Ordnung). In einem Beispiel ist eine Differenzspannung von etwa 100mV ausreichend. Gleichzeitig deaktiviert der LxCTRL **714** die muxL-Leitung **768**, so dass die entsprechenden Bitleitungen von dem Verstärkerkern **762** getrennt werden und ausgeglichen und vorgeladen werden.

[0156] Zu dem gleichen Zeitpunkt, an dem der LxCTRL **714** die muxL-Leitung **768** deaktiviert, aktiviert der LxCTRL **714** die senseH-Leitungen **766**, und in Abhängigkeit von dem globalen Multiplexen verstärkt der Verstärkerkern **762** schnell das differentielle Signal quer durch die Abfühlknoten. Sobald der Verstärkerkern **762** damit begonnen hat, das differentielle Signal abzufühlen, aktiviert der LxCTRL **714** die genL-Leitung **780**, so dass die lokalen Leseverstärker mit den globalen Bitleitungen verbunden werden. Der Verstärkerkern **762** setzt das Verstärken der differentiellen Signale auf die globalen Bitleitungen im Abhängigkeit von dem globalen Multiplexen fort. Der LxCTRL **714** entlädt das Signal gbitR **760**, um das

Ende der LESE-Operation zu signalisieren. Wenn das Signal gbitR **760** einen vorbestimmten Schwellwert überschreitet, deaktiviert der LxCTRL **714** die Signale senseH **766** und der Verstärkerkern **762** der LSA-Array stoppt das Verstärken. Dies führt zu einem differentiellen Signal mit begrenztem Hub in den globalen Bitleitungen, das repräsentativ für die Daten ist, die aus den Zellen ausgelesen worden sind.

[0157] Wenn die wlRH-Leitung **794** deaktiviert wird, lädt der LxCTRL **714** die bitR-Leitung **760** vor, um sie für den nächsten Zugriff vorzubereiten. Wenn die rbankL-Leitung **788** deaktiviert wird, deaktiviert der LxCTRL **714** die bnkL-Leitung, um sie für den nächsten Zugriff vorzubereiten.

TIMING DES SCHREIBZYKLUS

[0158] Die Zyklus-Zeitsteuerung für eine Schreiboperation gemäß einem Ausführungsbeispiel der vorliegenden Erfindung umfasst, dass der globale Controller und der globale Leseverstärker Daten oder ein Signal empfangen, die/das auf wbnkL übertragen werden/wird, dass ein hohes (high) Signal auf einer LwIH-Leitung übertragen oder bereitgestellt wird, und dass eine oder mehrere Speicherzellen ausgewählt werden. Die Schreiboperation ist vollendet, wenn die lokale Wortleitung hoch (high) ist.

[0159] Daten, die in eine Speicherzelle geschrieben werden sollen, werden synchron zu wbnkL in die gbit-Leitung gegeben. In diesem Ausführungsbeispiel agiert die Leitung wbnkL als die gbitR-Leitung in der Schreiboperation. In diesem Ausführungsbeispiel führt die Leitung wbnkL zur gleichen Zeit einen Pull-Down durch wie die Leitung gbit, aber um etwa 5- bis 6-mal schneller.

[0160] Das niedrige (low) Signal in der wbnkL-Leitung löst ein Signal in SenseH und einen lokalen Leseverstärker aus. Mit anderen Worten, genL geht hoch, wodurch der lokale Leseverstärker isoliert wird. Ein Signal in der Leitung wbnkL löst auch die Leitung bnkL aus, so dass lwIH hoch geht, wenn wlH ankommt. Nachdem das Signal in der Leitung SenseH übertragen ist, öffnet sich der Imux-Schalter, so dass Daten aus dem lokalen Leseverstärker auf die lokalen Bitleitungen wandern. BitR wird einem Pull-Down unterzogen. Mit anderen Worten, bitR und bit werden mit der gleichen Geschwindigkeit einem Pull-Down unterzogen, mit der ein voller BDT gespeichert wird. LwIL geht hoch und überlappt die Daten in den Bitleitungen. BitR schaltet LwIH ab und schließt den Imux-Schalter und SenseH.

[0161] Die Zyklus-Zeitsteuerung für eine SCHREIB-Operation, die ein Ausführungsbeispiel der vorliegenden Erfindung verwendet, ist in Fig. 21 veranschaulicht. Eine von vier ImuxL<0:3>-Leitungen **784** wird aktiviert, um eine der vier Zell-Array-Spalten auszuwählen, die von jedem LSA **712** unterstützt werden. Eine, zwei oder vier gmuxL<0:3>-Leitungen **786** werden aktiviert, um jede vierte, jede zweite oder jede globale Bitleitung für den Zugriff (d.h., 4:1-, 2:1- oder 1:1-Muxen) in Abhängigkeit von der globalen Multiplex-Option auszuwählen. Die Leitung blkSelH_bot **756** oder blkSelH_top **758** wird aktiviert, um dem Block anzuzeigen, ob auf den unteren Subblock **706B** oder auf den oberen Subblock **706A** zugegriffen werden soll. Die globale Wortleitung wird für eine bestimmte Speicherzeile, auf die zugegriffen werden soll, aktiviert.

[0162] Die wlRH-Leitung **794** wird gleichzeitig für alle Blöcke in dem Zeilenblock aktiviert, der die Speicherzeile enthält. Der GSA **724** präsentiert differentielle Daten mit einem begrenzten Hub oder mit vollem Hub in den globalen Bitleitungen. Die wbnkL-Leitung **792** wird aktiviert, um eine SCHREIB-Operation in den Block anzufordern. Der LxCTRL **714** aktiviert sofort die senseH-Leitungen **766** in Abhängigkeit von dem globalen Multiplexen, und der Verstärkerkern **762** verstärkt schnell das differentielle Signal quer durch die Abfühlknoten. Nur die Daten von den globalen Bitleitungen, die von dem globalen Multiplexen ausgewählt worden sind, werden verstärkt.

[0163] Der LxCTRL **714** aktiviert die bnkL-Leitung, um dem LxDEC **710** zu signalisieren, dass er eine lokale Wortleitung aktivieren soll. Der LxCTRL **714** aktiviert eine der vier muxL<0:3>-Leitungen **768**, die der aktivierten ImuxL-Leitung **784** entspricht. Dies bewirkt, dass der LSA **712** eine der vier Zellspalten mit dem Leseverstärker-Verstärkerkern **762** verbindet. Der Verstärkerkern **762** entlädt eine Bitleitung in jedem ausgewählten Paar auf VSS in Abhängigkeit von den originalen Daten in den globalen Wortleitungen. Der LxDEC **710**, der der aktivierten globalen Wortleitung entspricht, aktiviert die lokale Wortleitung. Die Daten aus den lokalen Wortleitungen werden in die Zellen geschrieben.

[0164] Gleichzeitig zu dem Schreiben der Daten aus den lokalen Bitleitungen in die Zellen aktiviert der LxCTRL **714** die lwlRH-Leitung **794**. Dieses Signal bewirkt, dass der LxCTRL **714** schnell die bitR-Leitung **760** entlädt. Wenn das Signal in der bitR-Leitung **760** einen vorbestimmten Schwellwert überschreitet, deaktiviert der LxDEC **710** die lokale Wortleitung. Die Daten sind nun vollständig in die Zellen geschrieben. Gleichzeitig deaktiviert der LxCTRL **714** die Leitungen senseH **766** und muxL **768** und reaktiviert die genL-Leitung **780**. Wenn die wlRH-Leitung **794** deaktiviert ist, lädt der LxCTRL **714** die bitR-Leitung **760** vor, um sie für den nächsten Zugriff vorzubereiten. Wenn die rbankL-Leitung **788** deaktiviert wird, deaktiviert der LxCTRL **714** die bnkL-Leitung, um diese für den nächsten Zugriff vorzubereiten. In einem Ausführungsbeispiel stellt bnkL lokale Banksignale für den lokalen Decoder bereit. Es wird in Betracht gezogen,

dass bnkL auch bnkL-top und bnkL-bot umfassen kann, wie dies vorher vorgesehen worden ist.

BURN-IN-MODUS

[0165] Nun kehren wir zu Fig. 7 zurück. Ein Ausführungsbeispiel der vorliegenden Erfindung umfasst einen Burn-in-Prozessormodus für die lokalen Blöcke, der von einer Burn-in-Leitung **796** (die alternativ als "BIL" bezeichnet wird) aktiviert wird. Dieser Prozess oder Modus belastet das SRAM-Modul oder den Block, um Defekte zu erfassen. Dies wird ermöglicht, indem gleichzeitig alle Leitungen lmuxL<0:3> **784**, blkSelH_bot **756**, blkSelH_top **758** und rbankL **788** aktiviert werden, aber nicht die Leitung wIRH **794** (d.h., die wIRH-Leitung **794** bleibt inaktiv). In diesem Fall wird BIL **796** aktiviert, was es den lokalen Wortleitungen erlaubt, in die LxDEC-**710**-Array durchzuschalten. Auch werden alle LSA-Muxe offen sein, was erlaubt, dass alle Bitleitungen gleichzeitig ausschwingen können. Schließlich wird bitR **760** nicht abklingen, weil wIRH **794** nicht aktiviert ist, und der Zyklus wird unendlich fortgesetzt, bis die Periode des hohen Taktes beendet ist.

LOKALER CLUSTER

[0166] In einem Ausführungsbeispiel kann ein Block in mehrere Cluster unterteilt werden. Das Unterteilen des Blocks in Cluster erhöht die Multiplexing-Tiefe des SRAM-Moduls und somit den Speicher. Obwohl die gemeinsamen lokalen Wortleitungen durch alle Cluster in einem einzigen Block laufen, werden nur die Leseverstärker in einem Cluster aktiviert. In einem Ausführungsbeispiel ist der lokale Clusterblock ein dünner Block mit einem niedrigen Overhead und einem Ausgang, der den Schwanzstrom aller lokalen Leseverstärker **712** in dem gleichen Cluster senkt. In diesem Ausführungsbeispiel umfasst der Block globale Schnittstellen oder Leitungen clusterL **799** und lokale Schnittstellen oder Leitungen clusterL **798** (was am besten in Fig. 7 zu sehen ist).

[0167] Vor einer LESE- oder SCHREIB-Operation wird eine globale clusterL-Leitung **799** (die alternativ als "gclusterL" bezeichnet wird) durch die externe Schnittstelle für alle Cluster, die in der LESE-/SCHREIB-Operation involviert sind, aktiviert. Der lokale Cluster umfasst eine gclusterL-Leitung **799** oder ein Signal in der Leitung, das gepuffert wird und zu clusterL **798** gesteuert wird. Die clusterL-Leitung **798** verbindet sich direkt mit dem Schwanzstrom aller lokalen Leseverstärker **712** in dem Cluster. Wenn der Cluster aktiv ist, werden die Leseverstärker durchschalten, aber wenn der Cluster inaktiv ist, werden die Leseverstärker nicht durchschalten. Da der Clustertreiber den Schwanzstrom der Leseverstärker tatsächlich senkt, muss der NMOS-Pull-Down sehr groß sein. Die Anzahl an Schwanzströmen, die der Cluster unterstützen kann, ist durch die Größe des NMOS-Pull-Down und die Breite der gemeinsamen Leitung, die an dem lokalen Leseverstärker-Schwanzstrom angebracht ist, beschränkt.

[0168] Es sollte klar sein, dass die Multiplexing-Architektur, die oben beschrieben worden ist, ohne den Verstärkerabschnitt des LSA **712** selbständig benutzt werden kann, wie dies in Fig. 2 veranschaulicht ist. In diesem Ausführungsbeispiel werden die lokalen Bitleitungs-Übertragungsgatter dazu verwendet, die lokalen Bitleitungen direkt mit den globalen Bitleitungen zu verbinden. Die GSAs **724** führen alle Funktionen des lokalen Leseverstärkers durch. Die Fläche des LSA **712** und des LxCTRL **714** wird kleiner, da weniger Funktionalität von diesen Blöcken gefordert wird. Für kleine Speicher und Speicher mittlerer Größe kann auch die Zugriffszeit abnehmen, da eine Kommunikationsstufe eliminiert worden ist. Das heißt, die Bitleitungen kommunizieren nun direkt mit dem GSA **724** anstatt mit dem LSA **712**. Die reduzierte Schnittstelle und die reduzierte Zeitsteuerung umfassen den LxDEC **710**, wie er vorher bereitgestellt worden ist, aber andere LSA **712** und LxCTRL **714**.

[0169] In diesem Ausführungsbeispiel sind die lokalen Bitleitungen hierarchisch ohne den LSA portioniert. Da gbit eine niedrigere Kapazität aufweist als lbit (aufgrund der Tatsache, dass sie zum Beispiel auseinander gespreizt ist und keine Diffusionslast ist), sind solche hierarchischen Speicher im Allgemeinen schneller und weisen eine niedrigere Energieperformanz auf im Vergleich zu einfachen flachen Speichern.

[0170] In einem Ausführungsbeispiel umfasst der Cluster eine eindimensionale Array von LSAs **712**, die sich aus vier Paaren von Bitleitungs-Multiplexern zusammensetzen. Jeder Bitleitungs-Multiplexer kann ein entsprechendes Paar von Bitleitungen mit der globalen Bitleitung durch eine ganzes Übertragungsgatter verbinden. Wenn ein Bitleitungspaar von der globalen Bitleitung getrennt wird, gleicht der Bitleitungs-Multiplexer das Bitleitungspaar aktiv aus und lädt es auf VDD vor. Weil es viermal weniger globale Bitleitungen als lokale Bitleitungen gibt, sind die globalen Bitleitungen physisch breiter und werden in einem größeren Abstand platziert. Wiederum reduziert dies den Widerstand und die Kapazität der langen globalen Bitleitungen beträchtlich, wodurch die Geschwindigkeit und die Zuverlässigkeit des Speichers erhöht werden.

[0171] Der LSA **712** wird von den Signalen muxL und lwIH gesteuert, die durch die gesamte LSA-**712**-Array gemeinsam genutzt werden. Die Leitung muxL<0:3> **768** wählt aus, welches der vier Paare von lokalen Bitleitungen für den aktuellen Zugriff verwendet werden soll. Jede lokale Bitleitung, die nicht für den Zugriff ausgewählt wird, wird von dem LSA **712** immer in einem vorgeladenen und aus-

geglichenen Zustand gehalten. In einem Beispiel sind die lokalen Bitleitungen auf VDD vorgeladen.

[0172] Die lwlRH-Leitung **794** stellt eine Schein-Poly-Leitung dar, die die lokale Poly-Wortleitung repliziert, die durch jede Zeile des Blocks läuft. Die lwlRH-Leitung **794** bildet das Gate der Schein-Transistoren, die jede Spalte der Zell-Array beenden. Jeder Schein-Transistor repliziert den Zugriffstransistor der 6T-SRAM-Zelle.

[0173] In einem globalen Clustermodus weist jeder Block einen einzelnen lokalen Controller auf, der die Aktivitäten der lokalen x-Decoder und Multiplexer durch das Ausüben der Steuersignale bitR **760**, bnkL, muxL **768** und lwlRH **782** koordiniert. Jedes dieser Signale wird von einem Treiber und einer Steuerlogikschaltung in der LxCTRL-Schaltung **714** aktiviert. Alle diese Signale sind normalerweise inaktiv, wenn sich der Speicher in dem Ruhezustand befindet. Die LxCTRL-Schaltung **714** wird wiederum von vertikalen und horizontalen Signalen aktiviert.

[0174] Die vertikalen Signale sind diejenigen Signale, die von allen LxCTRL-**714**-Schaltungen in dem gleichen Spaltenblock gemeinsam genutzt werden, und die die Leitungen lmuxL **784**, rbnkL **788**, rgbitR **760**, gbitR **760** und wbnkL **792** oder Signale in der Leitung umfassen. Nur eines/eine der vier Signale bzw. Leitungen lmuxL <0:3> **784** ist zu einem Zeitpunkt aktiv. Die aktive Leitung wählt eine der vier Zell-Array-Spalten, die mit jedem LSA **712** verbunden sind, für den Zugriff aus. Die rbnkL-Leitung **788** wird aktiviert, wenn eine LESE-Operation von dem Block angefordert wird. An dem Ende der LESE-Operation werden alle globalen Bitleitungen, die nicht aktiv von dem GSA **724** vorgeladen werden und die differentielle Signale mit begrenztem Hub enthalten, die die gespeicherten Werte in den Zellen darstellen, von der wIH-Leitung und den lmuxL-Signalen ausgewählt.

[0175] Die rgbitR-Leitung **769** wird extern auf VDD gehalten, wenn der Speicher im Ruhezustand ist, und wird zum Fließen gebracht, wenn ein Lesezugriff initiiert wird. Der LxCTRL-**714**-Block verbindet diese Leitung mit bitR **760** und entlädt diese Signalleitung auf VSS, wenn ein LESE-Zugriff vollendet ist.

[0176] Die wgbitR-Leitung **760** wird extern auf VDD gehalten, wenn der Speicher im Ruhezustand ist und wird während eines Schreibzugriffs entladen. Der LxCTRL-**714**-Block verbindet diese Leitung mit bitR **760** und baut auf das Signal, das bei VSS ankommt, um eine SCHREIB-Operation zu verarbeiten.

[0177] Die wbnkL-Leitung **792** wird aktiviert, wenn eine SCHREIB-Operation von dem Block angefordert wird. Differentielle Signale mit vollem Hub, die die Daten darstellen, die geschrieben werden sollen, sind in den globalen Bitleitungen vorhanden, wenn diese Leitung aktiviert ist.

[0178] Alle LxCTRL-**714**-Schaltungen in dem gleichen Zeilenblock nutzen die horizontalen Signale gemeinsam. Die wlRH-Leitung **794** ist eine Replik der globalen Wortleitung wlH, die durch jede Zeile des Speichers läuft. Das physikalische Layout der Leitung im Hinblick auf Metallbelag, Breite und Beabstandung repliziert die globale Wortleitung in jeder Zeile, um so die kapazitive Last gleich zu machen. Diese Leitung wird gleichzeitig mit einer einzigen globalen Wortleitung für eine Zeile in dem Block bei jedem Speicherzugriff aktiviert. Die blkSelH-Leitung ist bei jedem Speicherzugriff auf den Block aktiv und zeigt an, dass das Übertragungsgatter geöffnet werden soll.

[0179] Die Fig. 22A, Fig. 22B und Fig. 22C veranschaulichen unterschiedliche globale und Multiplexing-Anordnungen. Fig. 22A veranschaulicht ein Ausführungsbeispiel eines lokalen Leseverstärkers, der ein 4:1-Muxen und Vorladen und Ausgleichen umfasst. Der LSA wird hier als eine einzige Vorrichtung dargestellt, die vier bit/bit_n-Paare; eine SenseH-Leitung, eine GenL-Leitung, eine clusterL-Leitung und ein damit gekoppeltes gbit/bit_n-Paar aufweist. **Fig.** 22 veranschaulicht ein Beispiel des 4:1-Multiplexing (das alternativ als lokales 4:1-Muxen bezeichnet wird), das in den LSA eingebaut ist. In einem Ausführungsbeispiel ist jeder LSA mit 4 bit/bit_n-Paaren gekoppelt. Während einer LESE-/SCHREIB-Operation wird ein Bitleitungspaar der vier möglichen Bitleitungspaare, die mit jedem LSA gekoppelt sind, ausgewählt. Aber es werden auch Ausführungsbeispiele in Betracht gezogen, in denen die Cluster ohne das Fallenlassen der LSAs verwendet werden (d.h., die Clusters werden mit den LSAs verwendet).

[0180] Fig. 22B veranschaulicht ein Ausführungsbeispiel der vorliegenden Erfindung, das das 16:1-Muxen umfasst. Wiederum ist jeder LSA mit 4 Bitleitungspaaren gekoppelt (das lokale 4:1-Muxen, das oben vorgesehen worden ist). Hier sind vier SenseH-Leitungen <0:3> veranschaulicht, die mit den LSAs gekoppelt sind, wobei jeweils eine SenseH-Leitung mit jeweils einem LSA gekoppelt ist. Dies wird als 16:1-Muxen bezeichnet, das das globale 4:1-Muxen aufgrund der SenseH-Leitungen und das lokale 4:1-Muxen umfasst. Wenn eine der SenseH-Leitungen durchschaltet, wird einer der vier LSAs aktiviert, wodurch ermöglicht wird, dass eines der vier Bitleitungspaare, die mit dem aktivierten LSA gekoppelt sind, ausgewählt werden kann. Mit anderen Worten, diese Kombination ermöglicht, dass wenigstens ein Bitleitungspaar aus den insgesamt 16 zur Verfügung stehenden Bitleitungspaaren ausgewählt werden kann.

[0181] Fig. 22C veranschaulicht ein Ausführungsbeispiel der vorliegenden Erfindung, das das 32:1-Muxen umfasst. Wiederum ist jeder LSA mit 4 Bitleitungspaaren gekoppelt (das lokale 4:1-Muxen, das vorher vorgesehen worden ist). Hier sind vier SenseH-Leitungen <0:3> veranschaulicht, die mit den LSAs gekoppelt sind, wobei jeweils eine SenseH-Leitung mit zwei LSAs gekoppelt ist. Zum Beispiel ist eine SenseH-Leitung mit dem LSA 0 und dem LSA 4 gekoppelt, eine SenseH-Leitung ist mit dem LSA 1 und 4 gekoppelt, usw. Dieses Ausführungsbeispiel umfasst zwei lokale Clustervorrichtungen, wobei die erste lokale Clustervorrichtung mit den LSAs 1–3 über eine erste ClusterL-Leitung gekoppelt ist, während die zweite lokale Clustervorrichtung mit den LSAs 4–7 über eine zweite ClusterL-Leitung gekoppelt ist. Wenn ClusterL niedrig (low) ist, schalten die zugehörigen LSAs durch.

[0182] Die Clustervorrichtungen sind auch so veranschaulicht, dass sie mit den SenseH-Leitungen <0:3> und der GCTRL gekoppelt sind. Die GCTRL aktiviert eine oder mehrere lokale Clustervorrichtungen, die wiederum die assoziierte ClusterL-Leitung aktivieren. Wenn die assoziierte SenseH-Leitung durchschaltet, dann ist der LSA aktiv und ein Bitleitungspaar wird ausgewählt. Wenn zum Beispiel die GCTRL die erste Clustervorrichtung aktiviert, dann schaltet die erste ClusterL-Leitung durch (d.h., ClusterL ist niedrig). Wenn SenseH <0> ebenfalls durchschaltet, dann ist LSA 0 aktiv und eine der vier Bitleitungspaare, die mit dem LSA 0 gekoppelt sind, wird ausgewählt. Mit anderen Worten, diese Kombination ermöglicht es, dass wenigstens ein Bitleitungspaar aus den insgesamt 32 zur Verfügung stehenden Bitleitungspaaren ausgewählt werden kann.

[0183] Obwohl nur das 4:1-, 16:1- und 32:1-Muxen veranschaulicht ist, wird jede Multiplexing-Anordnung in Betracht gezogen (d.h., 8:1, 64:1, 128:1, etc.). Außerdem wird, obwohl nur zwei Clustervorrichtungen und zwei ClusterL-Leitungen veranschaulicht sind, jede Anzahl oder Anordnung davon in Betracht gezogen. Zum Beispiel kann die Anzahl an Clustervorrichtungen und Clusterleitungen in Abhängigkeit von der Anzahl der lokalen Blöcke in der Speicherarchitektur oder von den Multiplexing-Anforderungen variieren. Dies führt zu mehr Flexibilität und mehr Auswahlmöglichkeiten für eine gegebene Speicheranforderung.

[0184] Ein Ausführungsbeispiel der vorliegenden Erfindung betrifft ein System und ein Verfahren zum Anlegen einer Belastung gleichzeitig an alle Teile in der Speicherstruktur in einer parallelen Art und Weise. Dieses Ausführungsbeispiel umfasst die Hardware und eine Sequenz von Operationen, die notwendig sind, um schwache Defekte vollständig zu belasten, wodurch diese in der Testzeit zu einem vollständig durchgebrannten elektrischen Versagen gezwungen werden. Ein anderes Ausführungsbeispiel der vorliegenden Erfindung umfasst architektonische Verbesserungen und Schaltungsverbesserungen, die in einer hierarchischen Speicherarchitektur implementiert werden, die es ermöglicht, dass alle Speicherbits und die peripheren Schaltungen parallel belastet werden können.

[0185] Fig. 24 veranschaulicht ein Flussdiagramm, das allgemein mit **2400** bezeichnet ist und das eine Sequenz von Operationen zum Anlegen einer vollen Spannungsbelastung an alle benachbarten Leitungen veranschaulicht. Genauer gesagt sorgt diese Operation für das Anlegen einer vollen Spannungsbelastung an alle benachbarten Wortleitungen und Bitleitungen einer Speicherstruktur in einer parallelen Art und Weise. In einem Ausführungsbeispiel wird eine Burn-in-Leitung, die mit dem lokalen Vordecodierer gekoppelt ist (die der BiL **769** und dem Lx-PRED **716** ähnlich sind, die in Fig. 7 veranschaulicht sind), um den Burn-in-Modus zu initiieren.

[0186] Ein Ausführungsbeispiel der vorliegenden Erfindung betrifft ein System und ein Verfahren zum Anlegen einer Belastung an eine hierarchische Speicherstruktur in einer parallelen Art und Weise, wodurch die Speicherstruktur auf schwache Defekte getestet wird. Die vorliegende Erfindung umfasst das Schreiben einer logischen 0 in alle Speicherzellen in einer Speicherstruktur. Alle vordecodierten Leitungen mit hoher Adresse und alternierende vordecodierte Leitungen für die niedrigste Adresse werden aktiviert. Ein Spannungsabfall zwischen benachbarten Wortleitungen und Bitleitungen wird beeinflusst. Eine logische 1 wird in alle Speicherzellen in der Speicherstruktur geschrieben. Eine entgegengesetzte Spannungspolarität wird in den Bitleitungen aufgrund der logischen 1 in den Speicherzellen erzeugt. Eine umgekehrte Spannungspolaritätsbelastung wird in den Wortleitungen erreicht, indem der Zustand der niedrigsten vordecodierten Leitung gewechselt wird (d.h., durch das Ändern der Eingabeadresse, die dieser Leitung entspricht).

[0187] In einem Ausführungsbeispiel umfasst die Speicherarchitektur wenigstens zwei Gruppen von Leitungen, nämlich eine erste Gruppe, die wenigstens einen Satz von horizontalen Leitungen umfasst, und eine zweite Gruppe, die wenigstens einen Satz von vertikalen Leitungen umfasst. Wie in den Fig. 25 und Fig. 26 veranschaulicht ist, umfasst der erste Satz eine Vielzahl von Sätzen horizontaler Leitungen oder Wortleitungen. Der zweite Satz umfasst eine Vielzahl von Sätzen vertikaler Leitungen oder Bitleitungen. In diesem Ausführungsbeispiel verläuft die Vielzahl der Sätze horizontaler Leitungen im Allgemeinen parallel zueinander, während die Sätze vertikaler Leitungen im Allgemeinen parallel zueinander verlaufen. Des Weiteren verläuft, wie in den Fig. 25 und Fig. 26 veranschaulicht ist, wenigstens eine der

Wortleitungen im Allgemeinen senkrecht zu wenigstens einer der Bitleitungen. Obwohl die vorliegende Erfindung im Hinblick auf Wortleitungen und Bitleitungen erläutert wird, kommen alle möglichen parallelen Leitungspaare in Betracht. Außerdem kommen, obwohl in diesem Ausführungsbeispiel die ersten und zweiten Sätze jeweils horizontale und vertikale Leitungen umfassen, auch andere Anordnungen ebenfalls in Betracht. Zum Beispiel könnte der erste Satz vertikale Leitungen umfassen, während der zweite Satz horizontale Leitungen umfasst, obwohl auch diagonale Leitungen in Betracht kommen. Des Weiteren wird auch nur ein Satz von Leitungen in Betracht gezogen.

[0188] Wie in Fig. 24 veranschaulicht wird, werden die Speicherzellen in der Speicherarchitektur initialisiert. In einem Ausführungsbeispiel umfasst das Initialisieren der Speicherzellen das Schreiben von logischen 0en in alle Bitzellen in wenigstens einer Zell-Array, wie dies von Block **2402** angegeben wird. Ein Satz oder ein Paar der ersten Gruppe von Leitungen (zum Beispiel die Wortleitungen) wird aktiviert, wie dies jeweils durch den Block **2406** und **2406** angegeben wird. In einem Ausführungsbeispiel umfasst das Aktivieren der auslösenden Leitungen das Aktivieren der vordecodierten Leitungen der hohen Adresse und der alternierenden vordecodierten Leitungen für die niedrigste Adresse. Das Ergebnis einer solchen alternierenden Aktivierung ist, dass die ungeradzahligen (oder die geradzahligen Wortleitungen in Abhängigkeit davon, welche vordecodierten Leitungen die niedrigste Adresse aufweisen) aktiv werden, indem sie auf volle Netzspannung geladen werden. Dieser Vorgang wird wiederholt. Die Zellen werden initialisiert, wobei das Initialisieren der Zellen das Schreiben einer logischen "1" in alle Bitzellen umfasst, wie dies von Block **2408** veranschaulicht ist. Der erste und der zweite Satz von vordecodierten Leitungen werden aktiviert, wie dies von den Blöcken **2410** und **2412** veranschaulicht ist.

[0189] Die Fig. 25 und Fig. 26 veranschaulichen Ausführungsbeispiele des alternierenden Aktivierungsmusters. Fig. 25 veranschaulicht eine Vielzahl von lokalen x-Decodern **2510** (die dem LxDec **310** und dem LxDec **710** ähnlich sind, die jeweils in den Fig. 3B, Fig. 7 und Fig. 8 veranschaulicht worden sind). Jeder LxDec **2510** ist mit wenigstens einer vordecodierten Leitung gekoppelt. In diesem veranschaulichten Ausführungsbeispiel ist der LxDEC **2510** mit wenigstens zwei Adressvordecodiererleitungen **2502** und der Leitung **2504** oder der Leitung **2506** gekoppelt, so dass die Adressvordecodiererleitungen **2504** und **2506** mit dem LxDEC **2500** in einer alternierenden Art und Weise gekoppelt sind.

[0190] Die horizontalen benachbarten Metallwortleitungsleitungen **2526A** und **2526B** entwickeln eine potentielle Belastung aufgrund des alternierenden Aktivierungsmusters, wie dies in den Fig. 25 und Fig. 26 veranschaulicht ist. Das Durchschalten der vordecodierten Leitungen unter Verwendung einer minimalen Fläche und einer Zeitsteuerungs-Penalty wird in einem Ausführungsbeispiel erreicht, indem eine synchrone Hochgeschwindigkeits-Logikschaltung verwendet wird, wie dies unten bereitgestellt wird. Da alle Bänke aktiviert sind, sind die alternierenden Zeilen in allen Subblöcken innerhalb der Speicherzellen-Array aktiv. Da alle Speicherzellen in diesem Ausführungsbeispiel auf 0 initialisiert worden sind, enthalten die Zellen dieselben Daten, wobei das Abklingen der lokalen Bitleitungen eine Potentialdifferenz zwischen den Bitleitungen bildet. Diese Aktivierung wird wiederholt, wenn die Zellen auf logische 1en initialisiert werden, was die Belastungsspannungspolarität in den Bitleitungen umdreht. In einem Ausführungsbeispiel werden beide Belastungsoperationen (d.h., das Initialisieren der Zellen auf logische 0en und 1en) einmal mehr mit den geradzahligen (anstatt den ungeradzahligen) Wortleitungen wiederholt, die durchschalten, um die Belastungsspannungspolarität in den Bitleitungen umzukehren. In diesem Ausführungsbeispiel werden insgesamt 4 Zyklen benötigt, um den Speicher vollständig zu belasten.

[0191] Es wird in Betracht gezogen, dass die Daten, die in den Zellen gespeichert sind, die Bitleitungsbelastungspolarität bestimmen. Die niedrigste vordecodierte Leitung bestimmt, ob ungeradzahlige oder geradzahlige Wortleitungen aktiviert werden. Fig. 26 veranschaulicht eine Vielzahl von Speicherzellen **2632** in der Speicherzellen-Array **2608**, die mit einer Vielzahl von horizontalen Wortleitungen **2526A** und **2526B** gekoppelt sind. Die Zellen werden auf eine logische 0 initialisiert. Die lokalen Bitleitungen **2628**, **2630** werden belastet, so dass sie sich durch die aktivierten Zellen **2632** entladen. Die gleiche Operation wird wiederholt, wenn die Zellen **2632** auf logische "1"en initialisiert sind, was die Belastungsspannungspolarität in den Bitleitungen **2628** und **2630** umdreht. Wiederum werden in einem Ausführungsbeispiel beide Belastungsoperationen unter Verwendung der Zellen **2632**, die auf logische 0en und 1en initialisiert sind, mit den geradzahligen (anstatt den ungeradzahligen) Wortleitungen wiederholt, die durchschalten, um die Belastungsspannungspolarität in den Bitleitungen umzukehren. Folglich werden insgesamt 4 Zyklen verwendet, um den Speicher zu belasten.

[0192] In einem Ausführungsbeispiel wird der lokale Vordecodierer (der dem logischen Vordecodierer **716** ähnlich ist, der in Fig. 7 veranschaulicht worden ist) so modifiziert, dass, obwohl die Wortleitungen **2526A** und **2526B** durchschalten, die replizierten Wortleitungen, die den lokalen Controllerblock aktivieren, nicht durchschalten. Auf diese Weise werden die lokalen Bitleitungen **2628** und **2630** mit den globalen Bitleitungen **2710** und **2712** durch Passiergatter in dem

LSA verbunden, ohne dass der LSA durchschaltet, wie in Fig. 27 veranschaulicht ist. Die bit-gbit- und bit_n-gbit_n-Verbindungen oder -Durchgänge sind jeweils mit **2714** und **2716** bezeichnet. Die lokalen und globalen Controller blockieren das Vorladen der Bitleitungen **2628** und **2630** in dem Burn-in-Modus, so dass die Bitleitungen auf Null ausschwingen werden. Ein Rückblick auf Fig. 27 veranschaulicht auch, dass die gbit-Leitungen ausschwingen, wodurch eine Spannungsbelastung zwischen gbit und gbit_n erzeugt wird.

[0193] In einem Ausführungsbeispiel der vorliegenden Erfindung ist die Taktzykluszeit sehr lang (in der Größenordnung von Millisekunden). Der lange Taktzyklus steigert die Effizienz der Belastungszyklen. Eine quasistatische Spannung ist viel effizienter bei der Verschlimmerung der schwachen Defekte als eine dynamische Spannungsbelastung.

[0194] In einem Ausführungsbeispiel der vorliegenden Erfindung wird die Speicherarchitektur so modifiziert, dass sie einen Burn-in-Pin oder eine Burn-in-Leitung umfasst (was alternativ als "BiL" bezeichnet wird und ähnlich zu BiL **796** ist). Wenn die BiL-Leitung aktiviert wird, nachdem die Zellen korrekt auf logische 0en oder 1en initialisiert worden sind, werden alle Wortleitungen und Bitleitungen parallel belastet.

[0195] Ein Ausführungsbeispiel der vorliegenden Erfindung betrifft Logikschaltungen mit schneller Einstellung und mit einer großen Anzahl von Eingängen zum Decodieren von Speicheradressen. Solche Schaltungen können unter Verwendung eines Taktes mit einer relativ niedrigeren Geschwindigkeit zurückgesetzt werden. Eine solche bekannte Schaltung ist in Fig. 28 veranschaulicht.

[0196] Fig. 28 veranschaulicht eine taktvordecodierte Schaltung **2800** des NOR-Typs (NICHT-ODER-Typs), die entweder in einem Auswahlmodus (select mode) oder in einem Deaktivierungsmodus (deselect mode) arbeitet. Die veranschaulichte vordecodierte Schaltung **2800** umfasst 4 PFet-Transistoren **2810**, **2812**, **2814** und **2816**, wobei die PFets **2810** und **2816** Takteingänge an ihren jeweiligen Gates aufweisen. Außerdem umfasst die vordecodierte Schaltung jeweils 5 NFet-Transistoren **2818**, **2820**, **2822**, **2824** und **2826**. Wie veranschaulicht ist, weisen die NFet-Transistoren **2818**, **2820** und **2822** Adresseingänge A0, A1 und A2 auf, die mit ihren jeweiligen Gates gekoppelt sind, während der Takteingang mit dem Gate des NFet-Transistors **2826** gekoppelt ist. Außerdem ist der PFet-Transistor **2810** mit dem Knoten int0 **2830** gekoppelt, während der PFet-Transistor **2814** mit dem Knoten out0 **2832** gekoppelt ist. Der Knoten out0 **2832** ist mit dem Inverter **2828** gekoppelt, der einen Ausgang **2834** aufweist. Wenn alle Adresseingänge A0, A1 und A2 zu den NFet-Transistoren **2818**, **2820** und **2822** Null sind, werden die Transistoren ausgeschaltet und der Knoten int0 bleibt hoch (high), nachdem eine positive Taktflanke angekommen ist. Der Knoten out0 entlädt sich, was dazu führt, dass der Ausgang **2834** durchschaltet. Dies wird als der Auswahlmodus bezeichnet.

[0197] Aber wenn einer der Adresseingänge A0, A1 oder A2 hoch (high) ist, dann wird sein entsprechender Transistor eingeschaltet. Der Knoten int0 **2830** entlädt sich, wodurch der Knoten out0 **2830** auf hoch (high) gehalten wird. Dies bedeutet, dass der Knoten out0 unverändert bleibt und der Ausgang **2834** nicht durchschaltet. Dies wird als der Deaktivierungsmodus bezeichnet.

[0198] Die veranschaulichte vordecodierte Schaltung des NOR-Typs ist bekannt und kann dazu verwendet werden, viele Adresseingänge in einem einstufigen Gate schnell zu decodieren. Aber das Ausführungsbeispiel, das in Fig. 28 veranschaulicht ist, stellt keinen Burn-in-Modus bereit, wobei eine Standardsetz-/Rücksetzaktivität erfordert wird, die unabhängig von dem Adresseingang ist.

[0199] Fig. 29 veranschaulicht eine vordecodierte Schaltung **2900** mit Burn-in-Fähigkeit, die eine Standardsetz-/Rücksetzaktivität unabhängig von dem Adresseingang ermöglicht. Fig. 29 veranschaulicht vier PFet-Transistoren **2910**, **2912**, **2914** und **2916**, die wie gezeigt gekoppelt sind. Die PFet-Transistoren **2910** und **2916** weisen Takteingänge auf, die mit ihren jeweiligen Gates gekoppelt sind. Das veranschaulichte Ausführungsbeispiel umfasst sechs NFet-Transistoren **2918**, **2920**, **2922**, **2924**, **2926** und **2938**, wie dies veranschaulicht ist. Die Transistoren **2918**, **2920** und **2922** weisen Adresseingänge A0, A1 und A2 auf, die mit ihren jeweiligen Gates gekoppelt sind. Der Transistor **2926** weist einen Takt auf, der mit seinem Gate verbunden ist. Außerdem umfasst die vordecodierte Schaltung out0 **2932**, der mit einem Inverter **2928** gekoppelt ist, wobei der Ausgang **2934** damit gekoppelt ist.

[0200] Aber die Schaltung **2900** umfasst auch einen Burn-in-Transistor **2938**, der zusammen mit dem Transistor **2926** mit dem Knoten X **2936** gekoppelt ist. Der Transistor **2938** weist eine Burn-in-Leitung auf, die mit seinem Gate gekoppelt ist. In einem Ausführungsbeispiel ist diese Burn-in-Leitung der Leitung BiL **796** ähnlich. Wenn der Burn-in-Eingang niedrig ist, schaltet die vordecodierte Schaltung **2900** unabhängig von den Adresseingangswerten für A0, A1 und A2 durch. Während des Burn-in-Modus werden dann, wenn die alternierenden vordecodierten Leitungen aktiviert werden, die Leitungen mit dem niedrigstwertigen Adresseingang mit dem Knoten **2936** verbunden. Die vordecodierte Schaltung aktiviert nur diejenigen Leitungen, in denen die niedrigst-

wertige Adresse Null ist.

[0201] Noch ein anderes Ausführungsbeispiel einer vordecodierten Schaltung **3000** mit einer Burn-in-Fähigkeit ist in Fig. 30 veranschaulicht. Die veranschaulichte vordecodierte Schaltung ermöglicht nicht nur eine NOR/NAND-Funktionalität (NICHT-ODER/NICHT-UND-Funktionalität), sondern ermöglicht auch jeden komplexen booleschen Ausdruck in einem synchronen einstufigen Gate. Diese Schaltung **3000** umfasst zwei PFet-Transistoren **3010** und **3016**, von denen beide Takteingänge aufweisen, die mit ihren Gates gekoppelt sind. Außerdem umfasst diese Schaltung 10 NFet-Transistoren **3018**, **3020**, **3024**, **3026**, **3040**, **3042**, **3044**, **3046**, **3048** und **3050**, die wie gezeigt gekoppelt sind. Die Transistoren **3018**, **3020**, **3040**, **3042**, **3044**, **3046**, **3048** und **3050** sind logische Eingänge **3052**. Außerdem sind die NFet-Transistoren **3024** und der PFet-Transistor **3016** mit dem Knoten 0 **3032** gekoppelt, der mit dem Inverter **3028** gekoppelt ist. Der Inverter **3028** weist einen Ausgang **3034** auf, der dem ähnlich ist, der oben beschrieben worden ist. Die veranschaulichte vordecodierte logische Schaltung **3000** kann bei allen komplexen Vordecodierfunktionen in einem Speicher verwendet werden, wie etwa zum Beispiel bei einer Burn-in-Funktionalität, Redundanz, Multiplexing- und Auffrischdecodiervorgängen.

[0202] Ein anderes Ausführungsbeispiel einer vordecodierten Schaltung ist in Fig. 31 veranschaulicht. Diese Schaltung **3100** umfasst wiederum vier PFet-Transistoren **3110**, **3112**, **3114** und **3116**. 4 NFet-Transistoren **3118**, **3120**, **3122** und **3126**, ein Inverter **3128**, out0 **3132** und ein Ausgang **3136** sind ebenfalls veranschaulicht. Das Taktsignal wird dem PFet-Transistor **3110**, dem PFet-Transistor **3116** und dem NFet-Transistor **3126** eingegeben. Die NFet-Transistoren **3118** und **3120** sind Dateneingabetransistoren **3121**. Aber dieses Ausführungsbeispiel umfasst einen PFet-Transistor **3154** (als Phantomzeichnung dargestellt), der mit dem PFet-Transistor **3116** gekoppelt ist und der einen hohen Burn-in-Eingang aufweist, der mit seinem Gate gekoppelt ist, und einen NFet-Transistor **3152** (als Phantomzeichnung dargestellt), der mit dem Knoten 0 **3132** gekoppelt ist und der wiederum einen hohen Burn-in-Eingang aufweist, der mit seinem Gate gekoppelt ist. In diesem Ausführungsbeispiel kann der Transistor **3152** an Masse gelegt sein oder kann mit dem gemeinsamen Taktknoten gekoppelt sein.

[0203] Ein Vorteil der Schaltung **3100** liegt darin, dass die Dateneingabetransistoren für eine spannungsspitzenfreie Operation verwendet werden. Wenn während des Burn-in-Belastungsmodus ein hohes Signal in der Burn-in-Leitung übertragen wird, wird der PFet-Transistor **3154** abgeschaltet und der NFet-Transistor **3152** wird eingeschaltet, so dass der Ausgang **3136** unabhängig von den Eingangswerten der Dateneingabetransistoren durchschaltet.

[0204] Ein alternatives Ausführungsbeispiel der vordecodierten Schaltung wird in Betracht gezogen, wenn ein alternierendes Aktivieren erfordert wird. Dieses Ausführungsbeispiel umfasst einen Stapel (stack) **3256**, der den Transistor **3152** von Fig. 31 ersetzt. Dieser Stapel umfasst zwei NFet-Transistoren **3258** und **3260**, die miteinander gekoppelt sind, wobei BiLH mit dem Gate des Transistors **3258** und addr 0 mit dem Gate des Transistors **3260** gekoppelt ist.

[0205] Ein alternatives Ausführungsbeispiel einer Vordecodiererschaltung **3300**, die den Stapel **3256** umfasst, ist in Fig. 33 veranschaulicht. Die Schaltung umfasst drei PFet-Transistoren **3310**, **3312** und **3314**, wobei die PFet-Transistoren **3312** und **3314** in Reihe mit dem Knoten 0 **3332** geschaltet sind. 5 NFet-Transistoren **3318**, **3320**, **3322**, **3324** und **3326** sind veranschaulicht, wobei addr 0_n mit dem Gate des Transistors **3318** gekoppelt ist und ein Takteingang mit dem Gate des Transistors **3326** gekoppelt ist. Der Stapel **3256** ist so veranschaulicht, dass er mit dem Knoten 0 **3332** gekoppelt ist, der mit dem Inverter **3328** gekoppelt ist, der mit dem Ausgang **3336** gekoppelt ist. Der Stapel **3256** umfasst den Transistor **3258**, bei dem BiLH mit seinem Gate gekoppelt ist, während bei dem Transistor **3260** addr 0 mit seinem Gate gekoppelt ist. Wenn der Stapel **3256** an Masse gelegt ist, wie veranschaulicht ist, kann das Durchbrennen (Burn-in) erzielt werden, ohne dass der Speicher getaktet werden muss.

[0206] Viele Modifikationen und Variationen der vorliegenden Erfindung sind angesichts der oben genannten Lehren möglich. Somit sollte es klar sein, dass die Erfindung innerhalb des Rahmens der anhängenden Ansprüche auf andere Weise als oben beschrieben praktiziert werden kann.

Patentansprüche

1. Verfahren zum Belasten einer hierarchischen Speicherstruktur, die mit einer Gruppe paralleler Leitungen (**2526A**, **2526B**) gekoppelt ist, die erste und zweite Sätze alternierender vordecodierter Leitungen einer niedrigsten Adresse aufweist, wobei sich der erste Satz von dem zweiten Satz unterscheidet, wobei das Verfahren die folgenden Schritte umfasst:
– Schreiben einer logischen 0 in alle Zellen der Speicherstruktur;
– Aktivieren aller vordecodierten Leitungen einer höheren Adresse (**2502**) aus der Gruppe paralleler Leitungen (**2526A**, **2526B**);
– Aktivieren des ersten Satzes alternierender vordecodierter Leitungen einer niedrigsten Adresse (**2506**), wodurch ein Spannungsabfall zwischen benachbarten Leitungen innerhalb der Gruppe paralleler Leitungen (**2526A**, **2526B**) erzeugt wird;
dann:

– Schreiben einer logischen 1 in alle Zellen der Speicherstruktur;
– Aktivieren aller vordecodierten Leitungen einer höheren Adresse (**2502**) aus der Gruppe paralleler Leitungen (**2526A**, **2526B**); und
– Aktivieren des zweiten Satzes alternierender vordecodierter Leitungen einer niedrigsten Adresse (**2506**), wodurch ein umgekehrter Spannungsabfall zwischen benachbarten Leitungen innerhalb der Gruppe paralleler Leitungen (**2526A**, **2526B**) erzeugt wird, wobei die Zellen an komplementäre Bitleitungspaare gekoppelt sind.

2. Verfahren nach Anspruch 1, wobei die vordecodierten Leitungen (**2502**, **2506**) gleichzeitig aktiviert werden.

3. Verfahren nach Anspruch 1, wobei das Aktivieren der vordecodierten Leitungen das Aktivieren von Bitleitungen (**2502**, **2506**), die mit der Speicherstruktur gekoppelt sind, umfasst.

4. Verfahren nach Anspruch 3, wobei das Aktivieren der Bitleitungen das Aktivieren aller vordecodierten Bitleitungen höherer Adresse (**2502**) und alternierender vordecodierter Bitleitungen für die niedrigste Adresse (**2506**) umfasst.

5. Verfahren nach Anspruch 1, wobei das Aktivieren der vordecodierten Bitleitungen das Aktivieren von Wortleitungen, die mit der Speicherstruktur gekoppelt sind, umfasst.

6. Verfahren nach Anspruch 5, wobei das Aktivieren der Wortleitungen das Aktivieren aller vordecodierten Wortleitungen der hohen Adresse und der alternierenden vordecodierten Wortleitungen für die niedrigste Adresse umfasst.

Es folgen 32 Blatt Zeichnungen

Anhängende Zeichnungen

FIG. 1

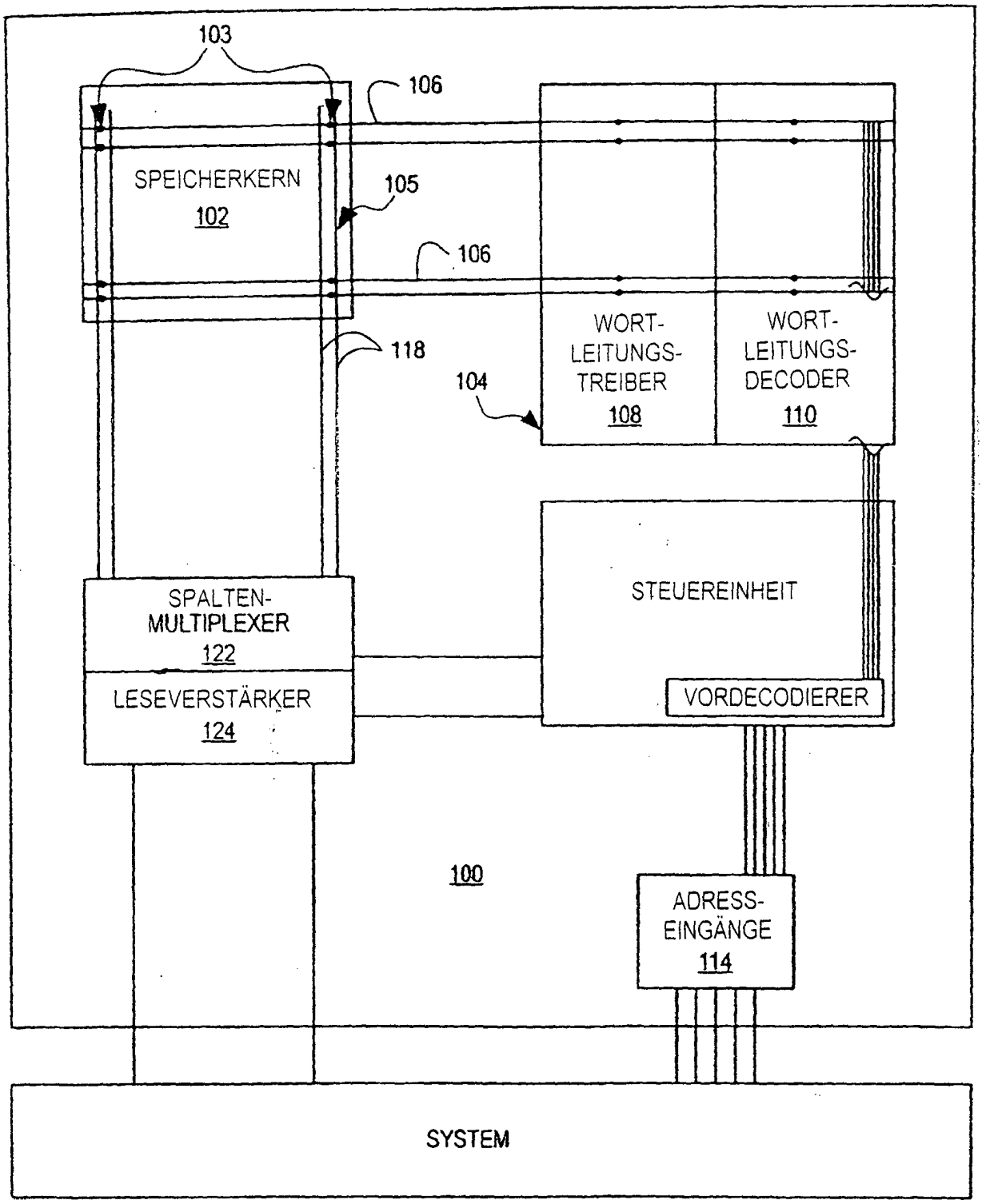


FIG. 2

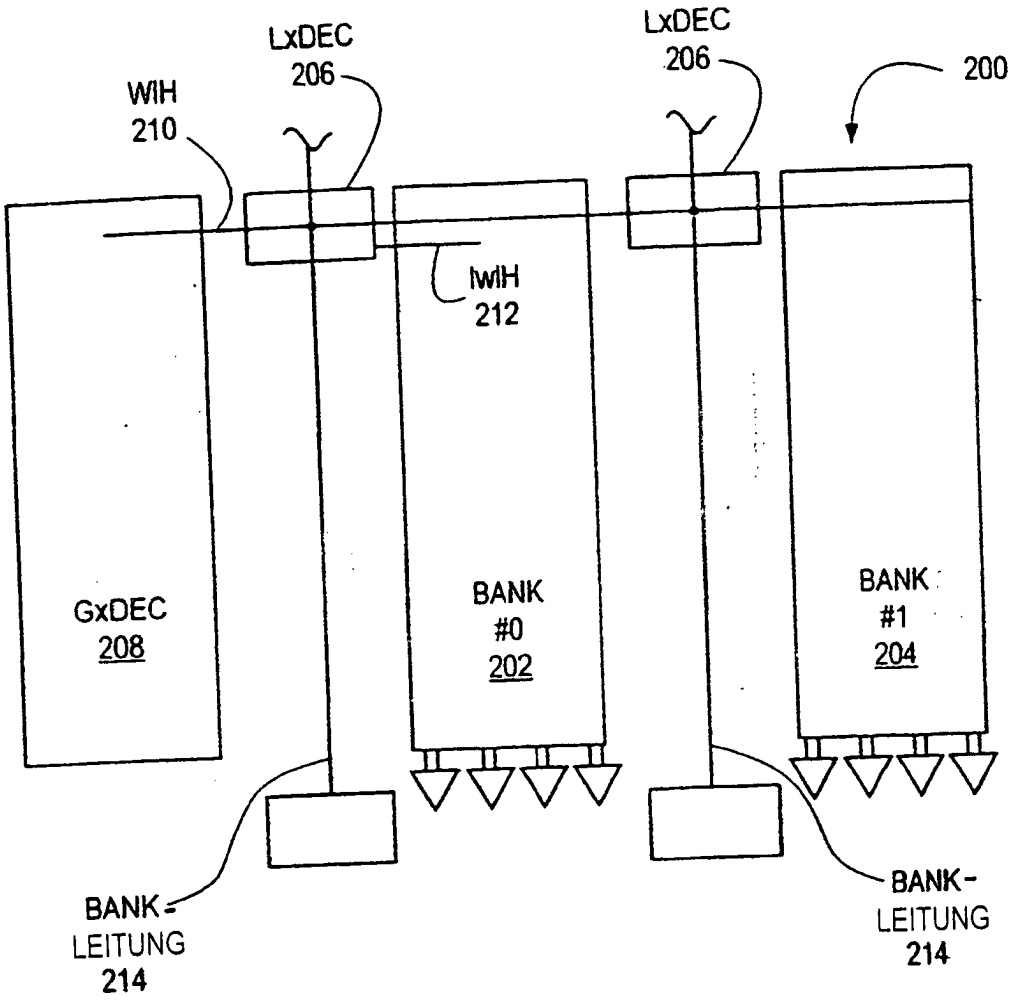

LxDEC
206
LxDEC
206
200
WlH
210
lwlH
212
GxDEC
208
BANK
#0
202
BANK
#1
204
BANK-
LEITUNG
214
BANK-
LEITUNG
214

FIG. 3A

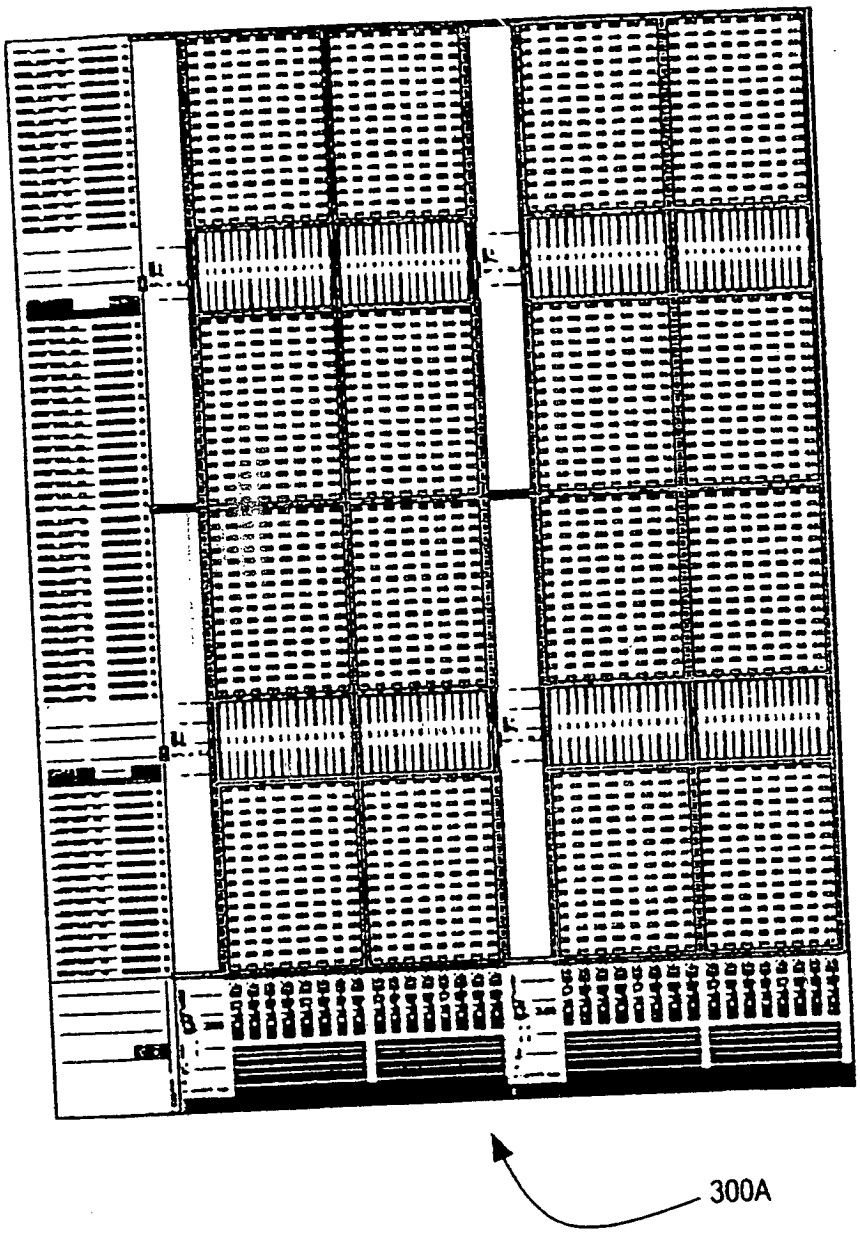


FIG. 3B

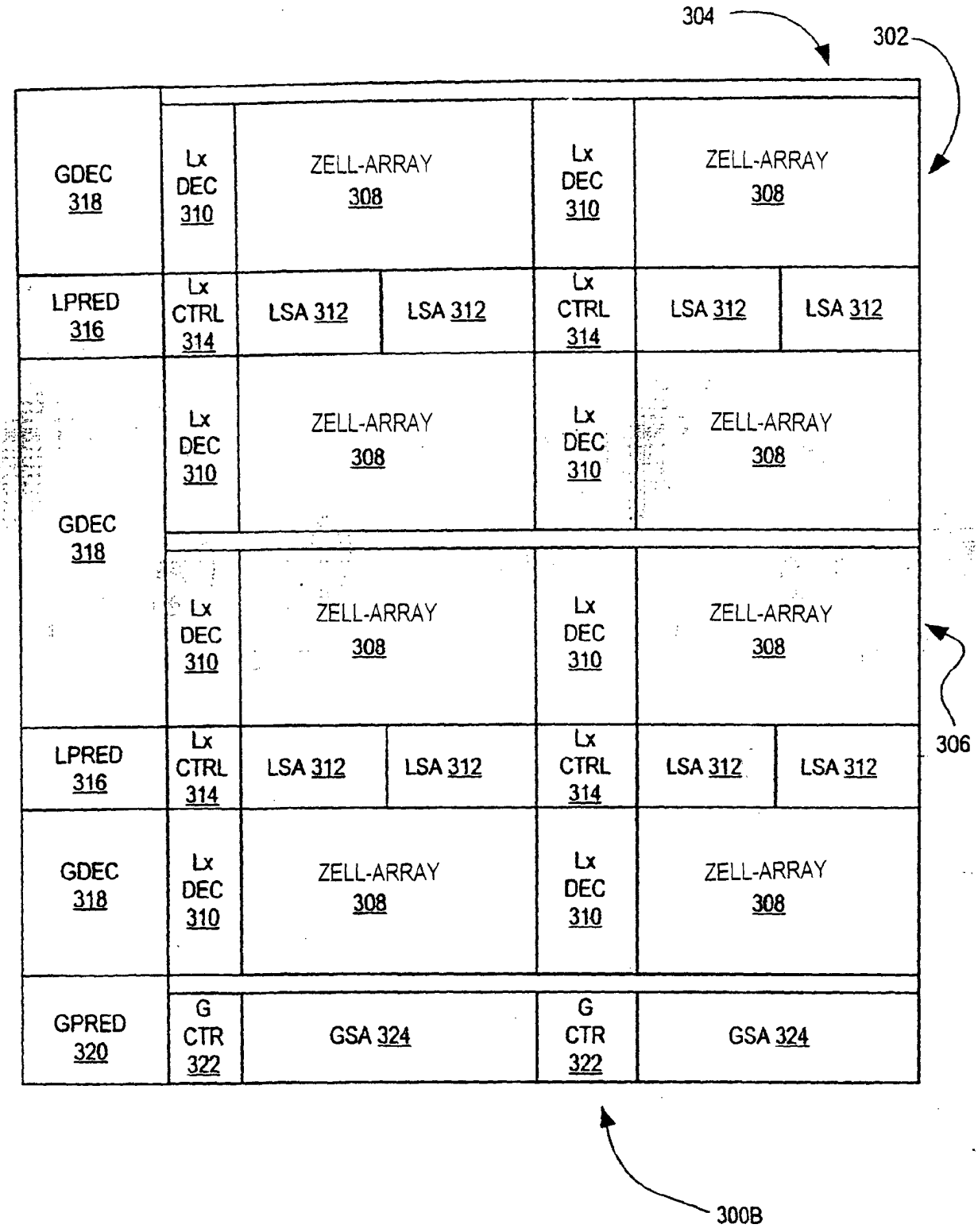

304
302
GDEC 318
Lx DEC 310
ZELL-ARRAY 308
Lx DEC 310
ZELL-ARRAY 308
LPRED 316
Lx CTRL 314
LSA 312
LSA 312
Lx CTRL 314
LSA 312
LSA 312
GDEC 318
Lx DEC 310
ZELL-ARRAY 308
Lx DEC 310
ZELL-ARRAY 308
Lx DEC 310
ZELL-ARRAY 308
Lx DEC 310
ZELL-ARRAY 308
306
LPRED 316
Lx CTRL 314
LSA 312
LSA 312
Lx CTRL 314
LSA 312
LSA 312
GDEC 318
Lx DEC 310
ZELL-ARRAY 308
Lx DEC 310
ZELL-ARRAY 308
GPRED 320
G CTR 322
GSA 324
G CTR 322
GSA 324
300B

FIG. 4

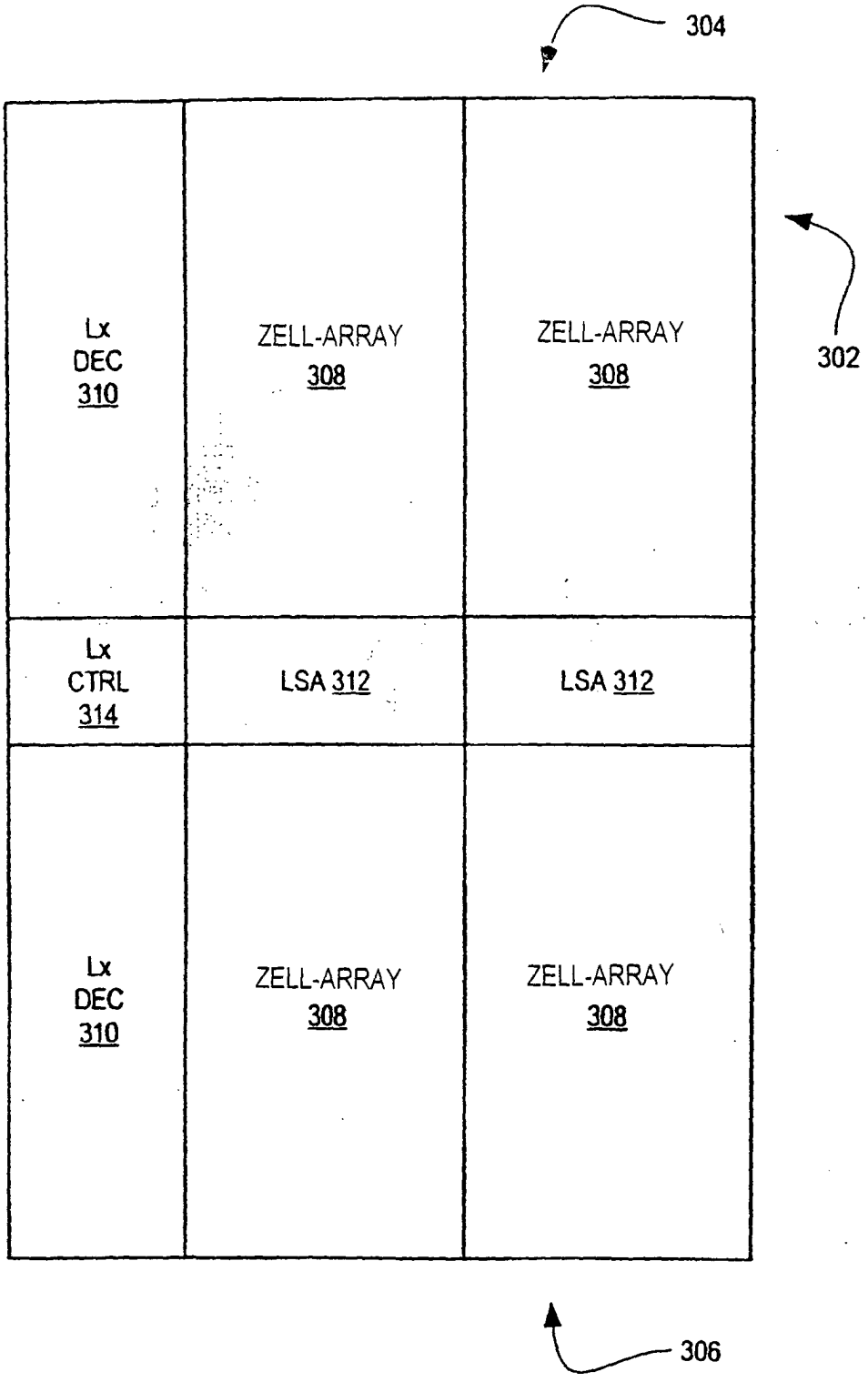

304
Lx DEC 310
ZELL-ARRAY 308
ZELL-ARRAY 308
302
Lx CTRL 314
LSA 312
LSA 312
Lx DEC 310
ZELL-ARRAY 308
ZELL-ARRAY 308
306

FIG. 5

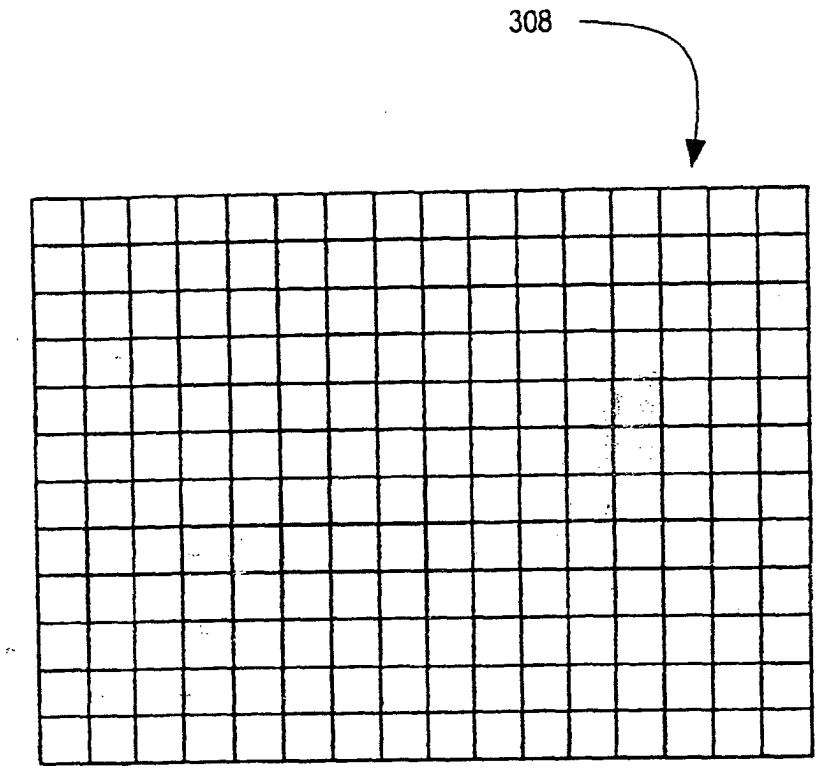


FIG.6A

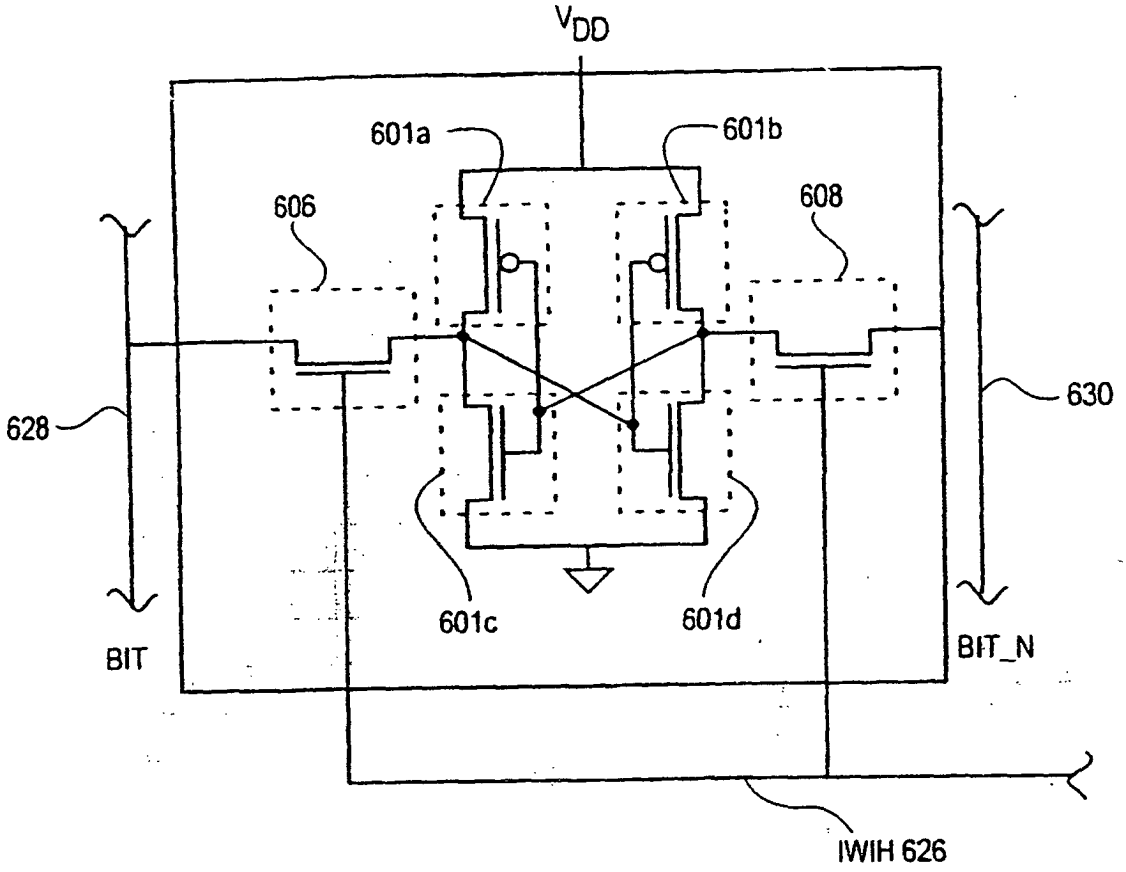

V_{DD}
601a
601b
606
608
628
630
601c
601d
BIT
BIT_N
IWIH 626

FIG.6B

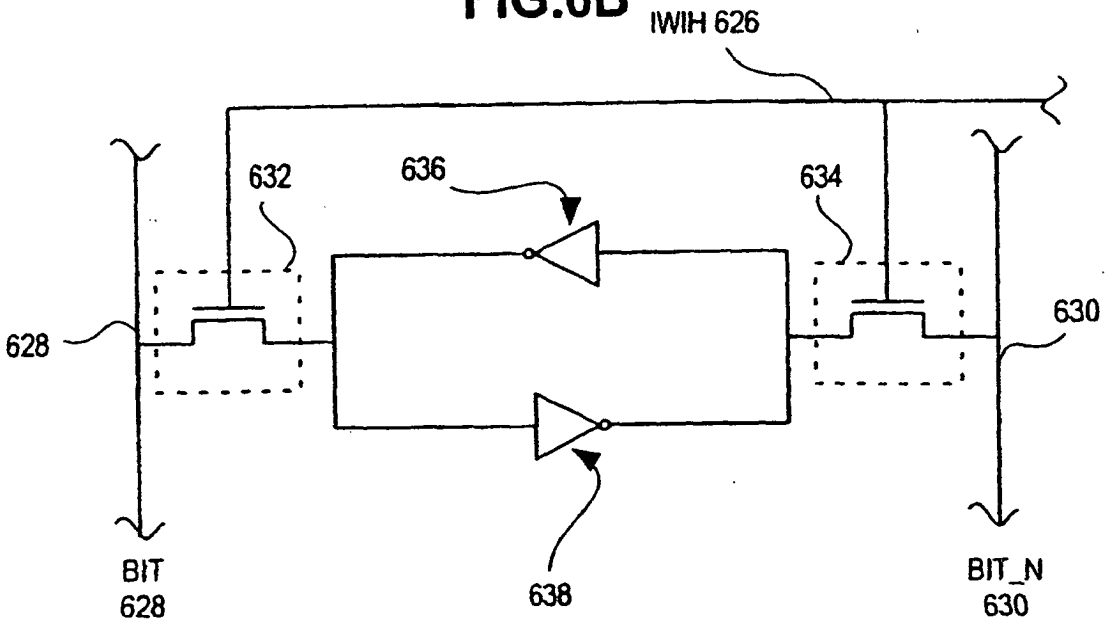

IWIH 626
632
636
634
628
630
BIT
628
638
BIT_N
630

FIG. 7

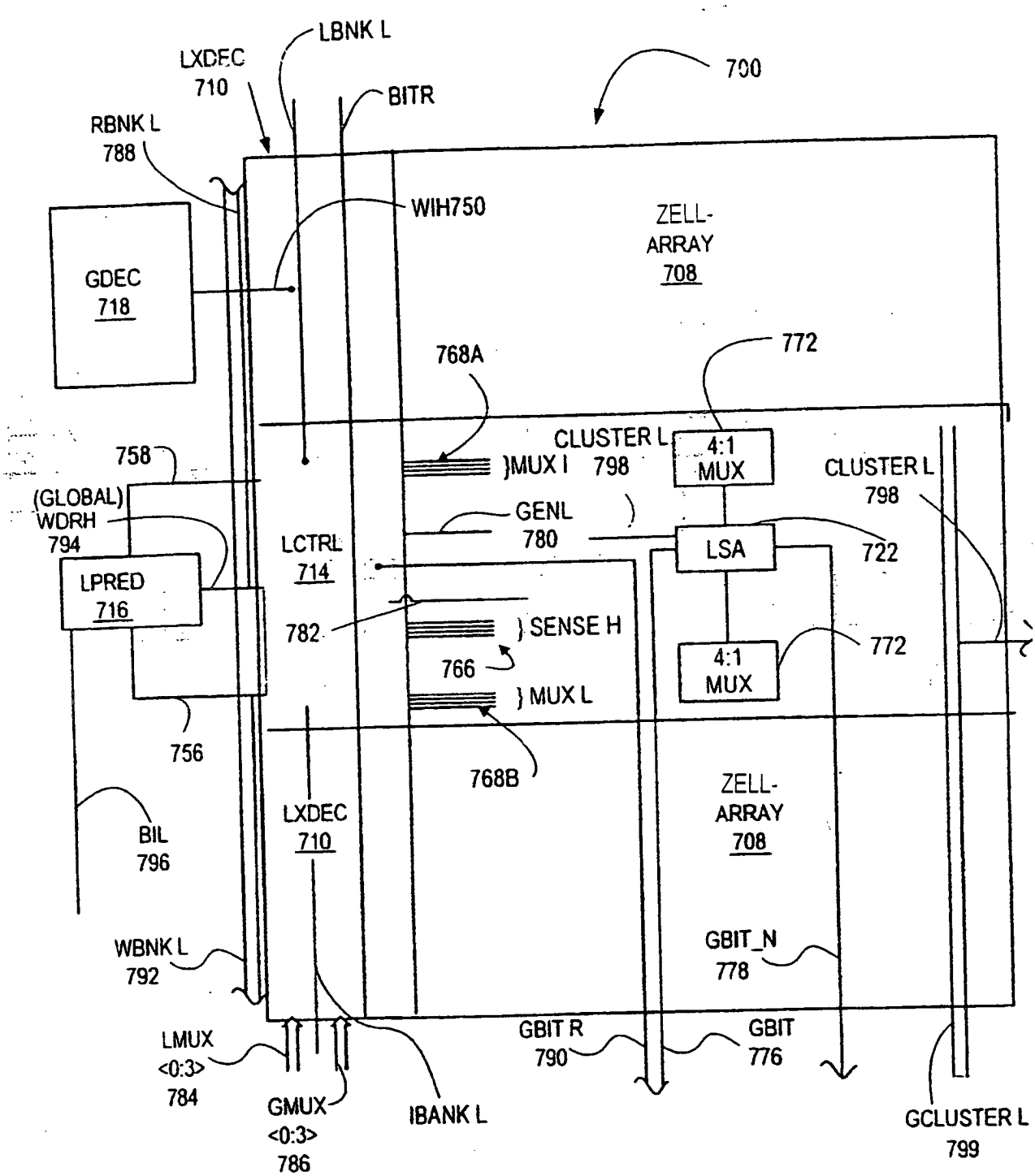

LBNK L
LXDEC
710
BITR
700
RBNK L
788
WIH750
ZELL-
ARRAY
708
GDEC
718
768A
772
CLUSTER L
758
}MUX I
798
4:1
MUX
CLUSTER L
798
(GLOBAL)
WDRH
794
GENL
780
LCTRL
714
LSA
722
LPRED
716
782
} SENSE H
766
4:1
MUX
772
} MUX L
756
768B
ZELL-
ARRAY
708
LXDEC
710
BIL
796
WBNK L
792
GBIT_N
778
LMUX
<0:3>
784
GBIT R
790
GBIT
776
GMUX
<0:3>
786
IBANK L
GCLUSTER L
799

FIG. 8

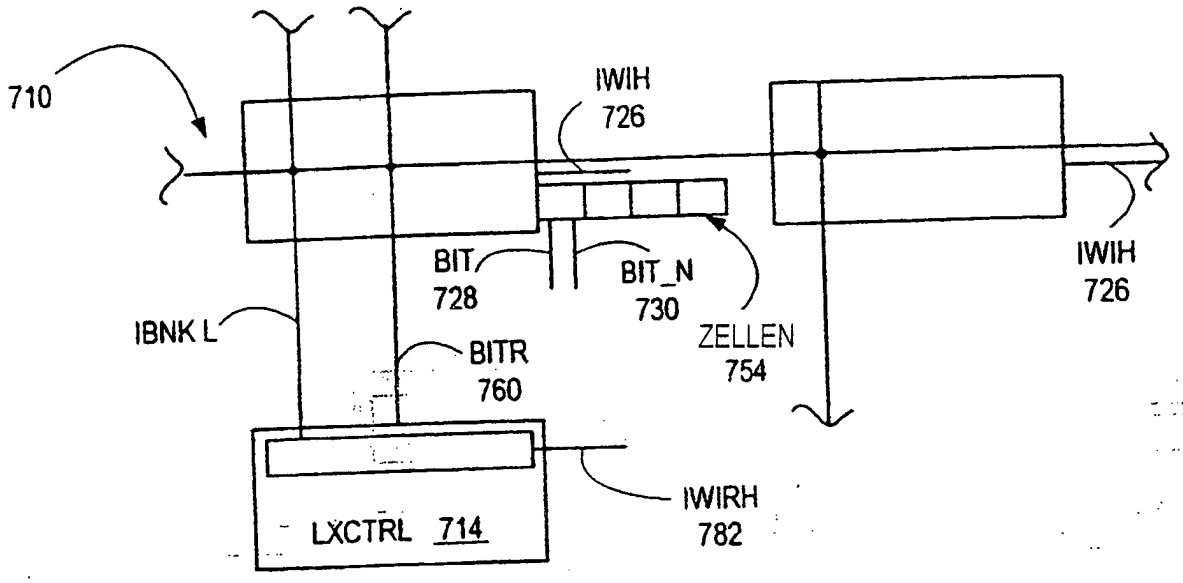

710
IWIH
726
BIT
728
BIT_N
730
ZELLEN
754
IWIH
726
IBNK L
BITR
760
IWIRH
782
LXCTRL 714

FIG. 9

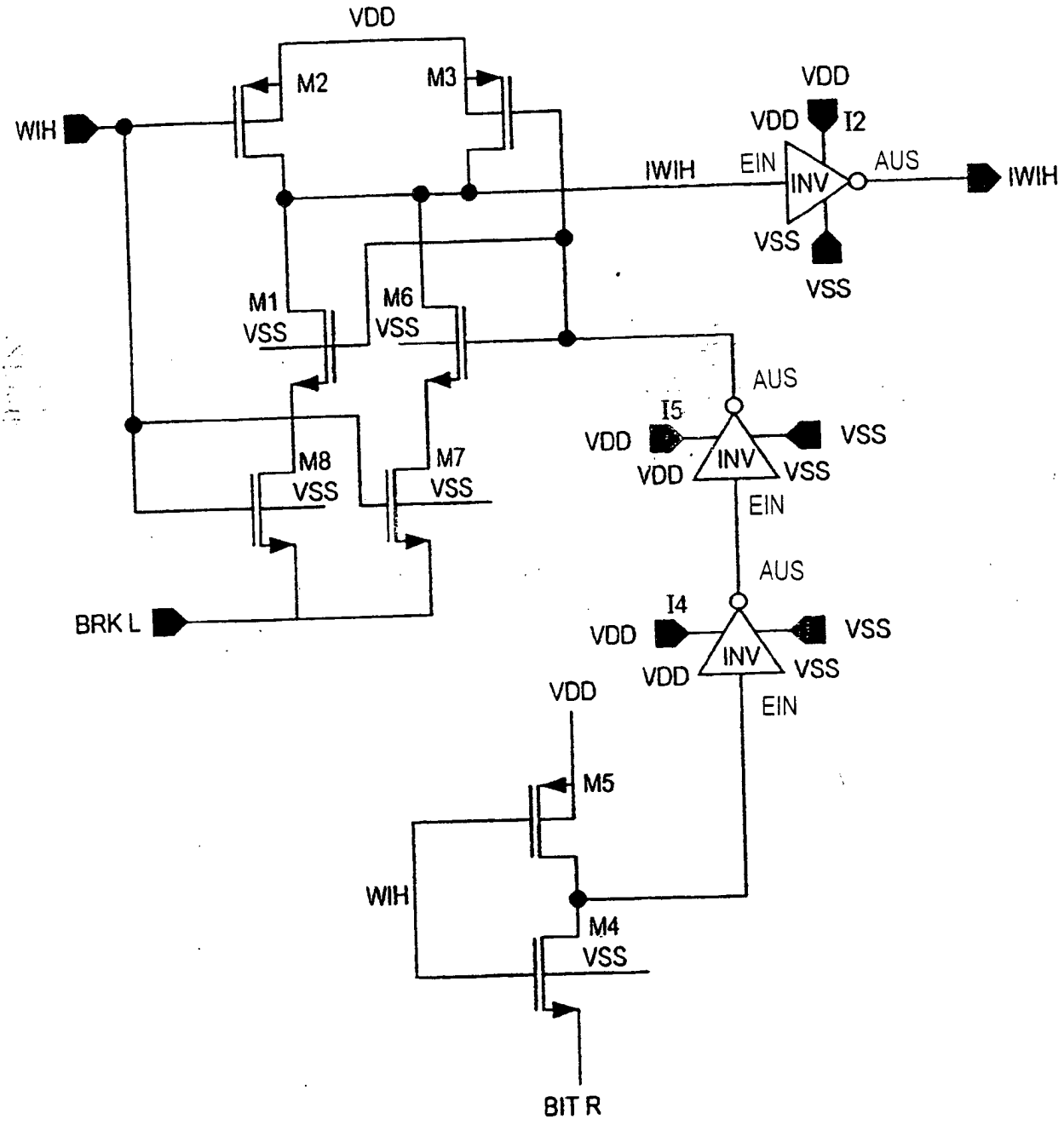

VDD
M2
M3
WIH
VDD
VDD
I2
IWIH
EIN
INV
AUS
IWIH
VSS
VSS
M1
VSS
M6
VSS
AUS
I5
VDD
INV
VSS
VDD
VSS
M8
VSS
M7
VSS
EIN
AUS
BRK L
I4
VDD
INV
VSS
VDD
VSS
EIN
VDD
M5
WIH
M4
VSS
BIT R

FIG. 10

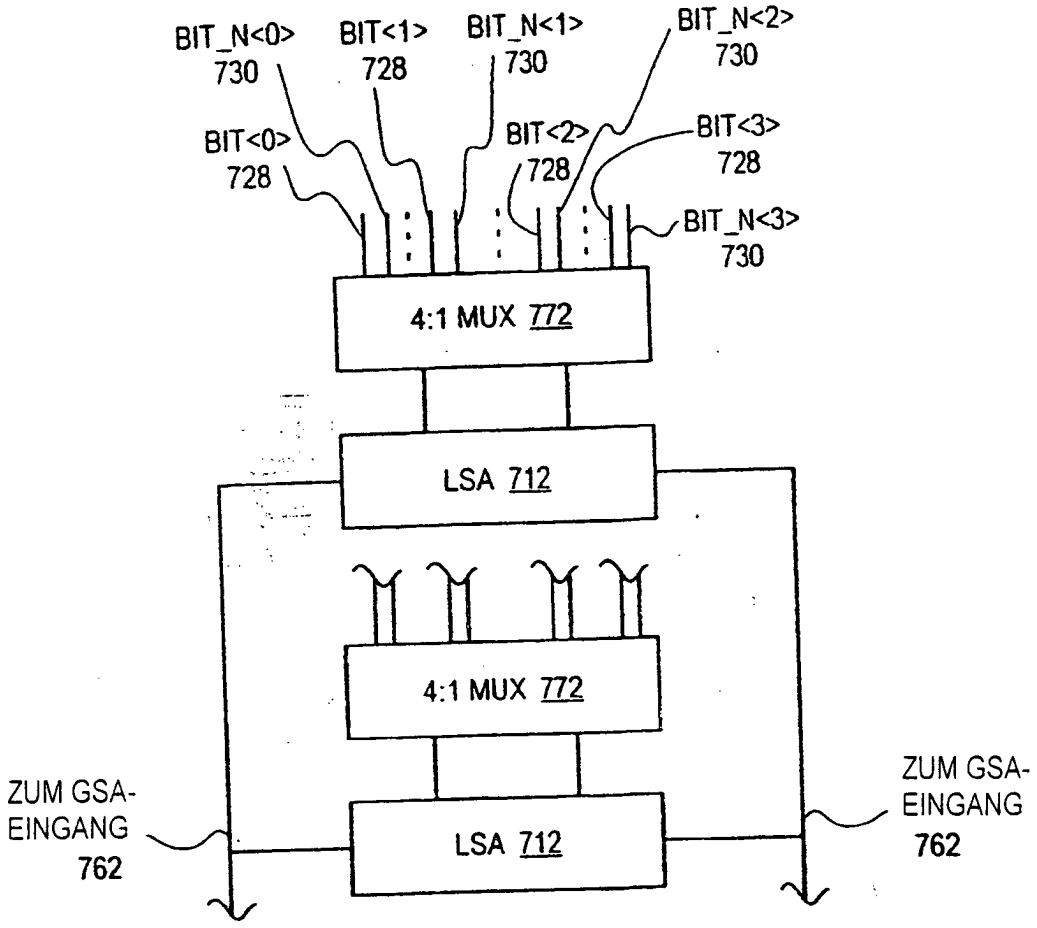

BIT_N<0>
730
BIT<1>
728
BIT_N<1>
730
BIT_N<2>
730
BIT<0>
728
BIT<2>
728
BIT<3>
728
BIT_N<3>
730
4:1 MUX 772
LSA 712
4:1 MUX 772
LSA 712
ZUM GSA-
EINGANG
762
ZUM GSA-
EINGANG
762

FIG.11

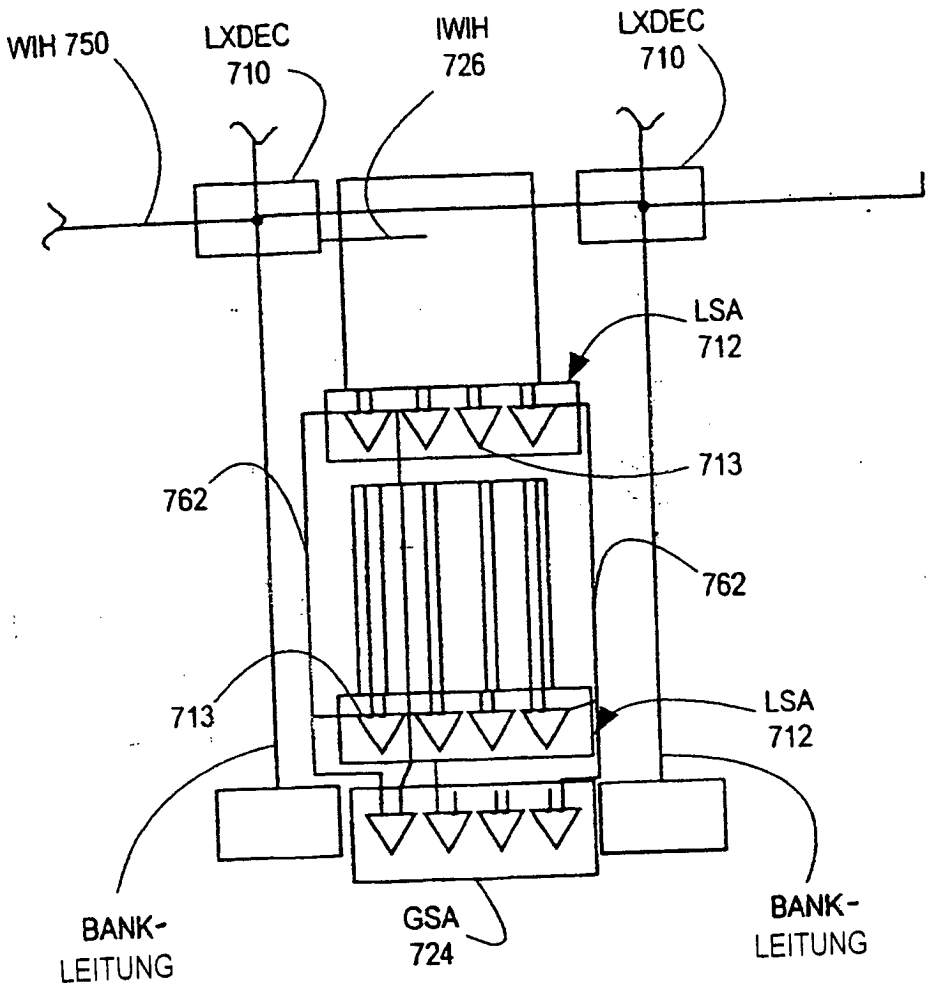

WIH 750
LXDEC 710
IWIH 726
LXDEC 710
LSA 712
713
762
762
713
LSA 712
BANK-LEITUNG
GSA 724
BANK-LEITUNG

FIG. 12A

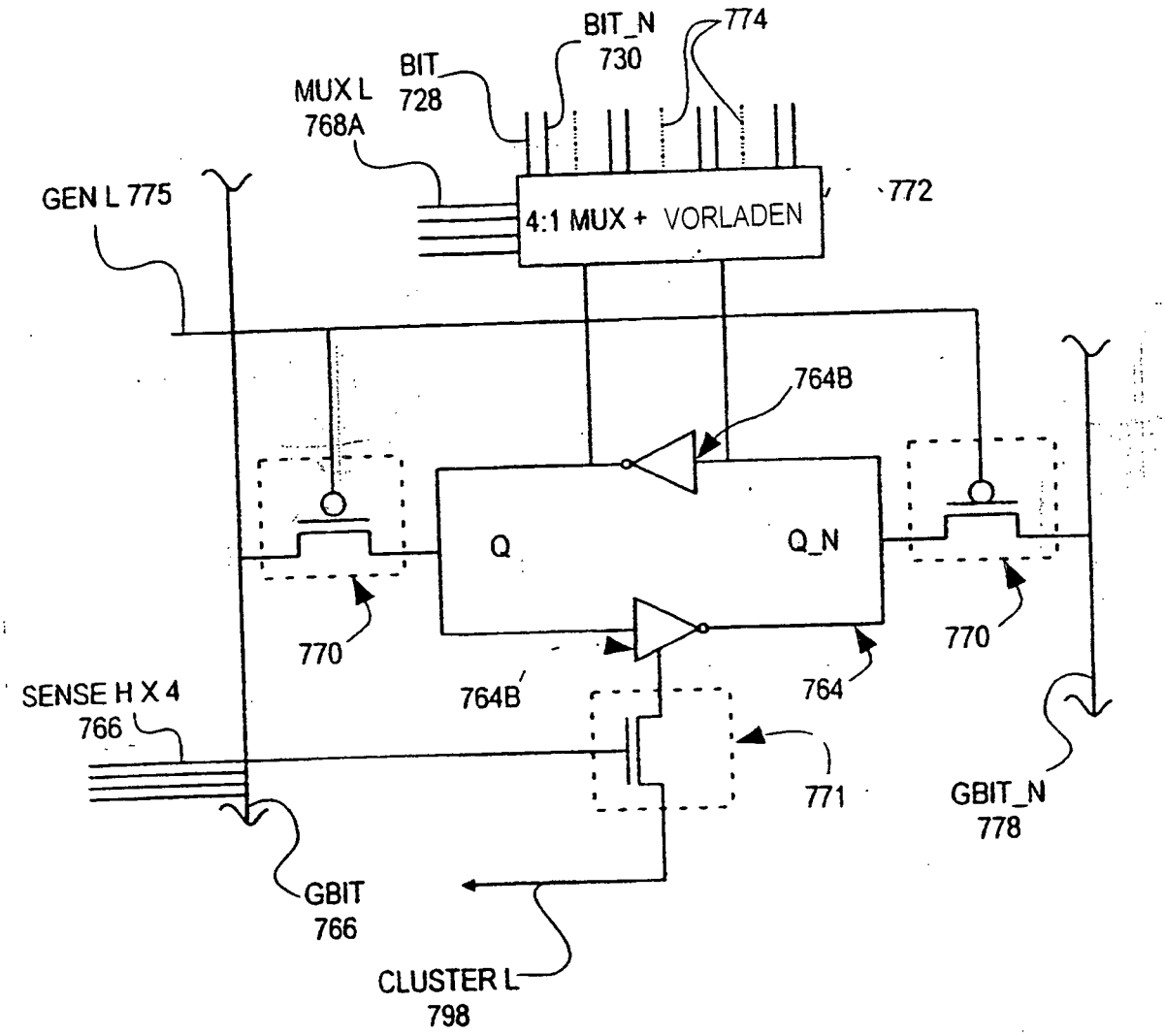

BIT_N
730
774
BIT
728
MUX L
768A
GEN L 775
4:1 MUX + VORLADEN
772
764B
Q
Q_N
770
770
764B'
764
SENSE H X 4
766
771
GBIT_N
778
GBIT
766
CLUSTER L
798

FIG. 12B

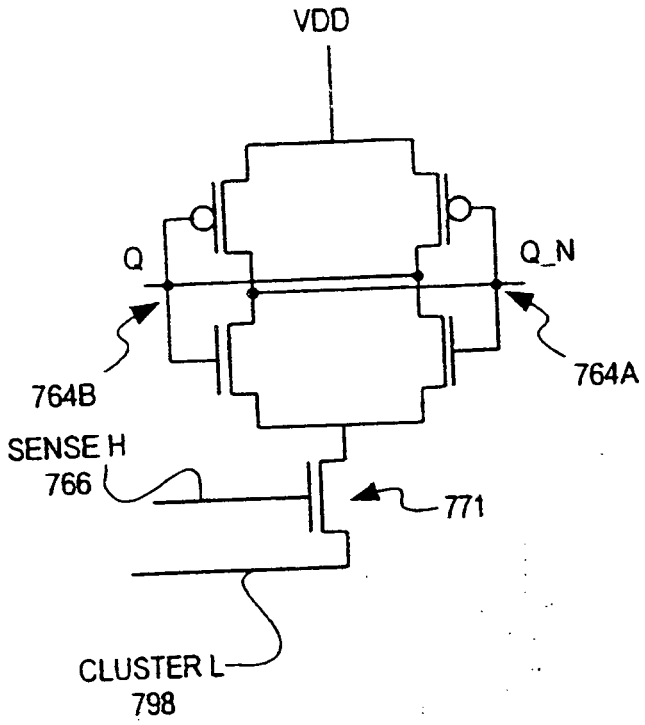


FIG. 12C

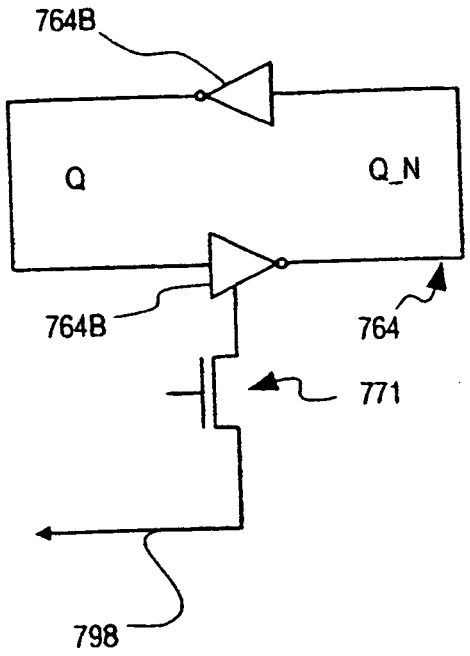


FIG. 13

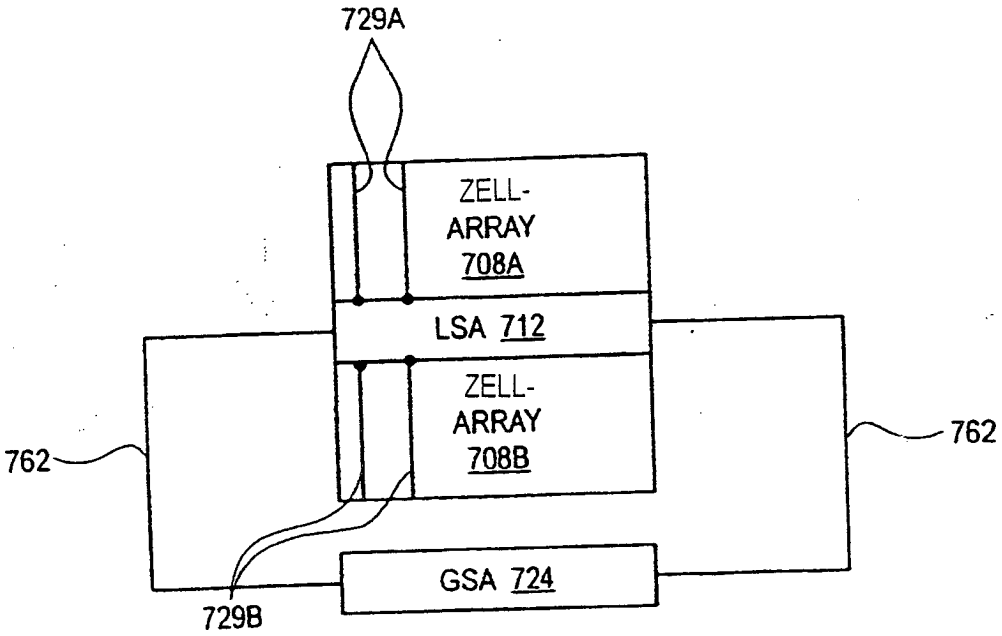

729A
ZELL-
ARRAY
708A
LSA 712
ZELL-
ARRAY
708B
762
762
GSA 724
729B

FIG. 14

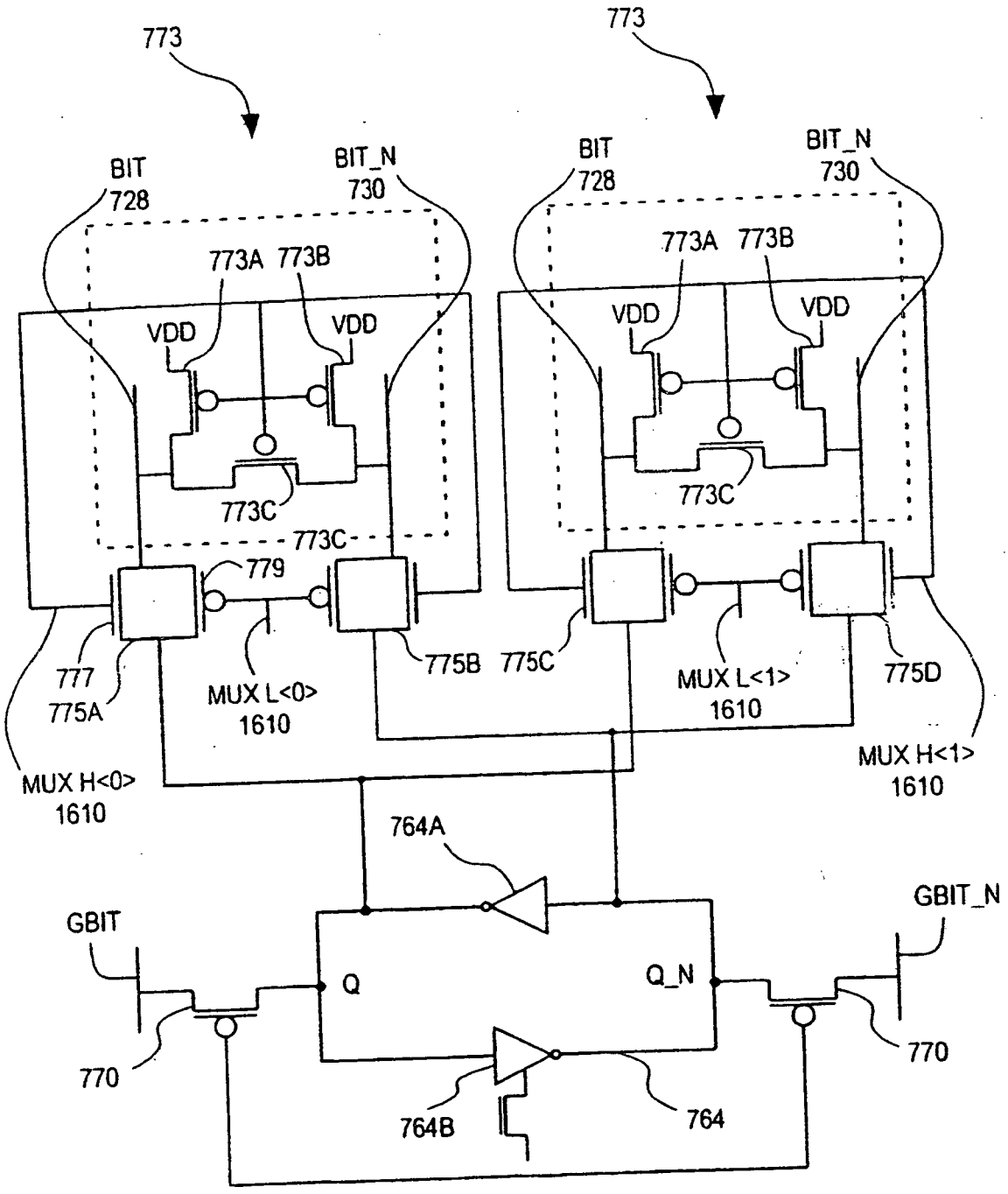

773
773
BIT
728
BIT_N
730
BIT
728
BIT_N
730
773A 773B
773A 773B
VDD
VDD
VDD
VDD
773C
773C
773C
779
777
775A
MUX L<0>
1610
775B
775C
MUX L<1>
1610
775D
MUX H<0>
1610
MUX H<1>
1610
764A
GBIT
GBIT_N
Q
Q_N
770
770
764B
764

FIG. 15

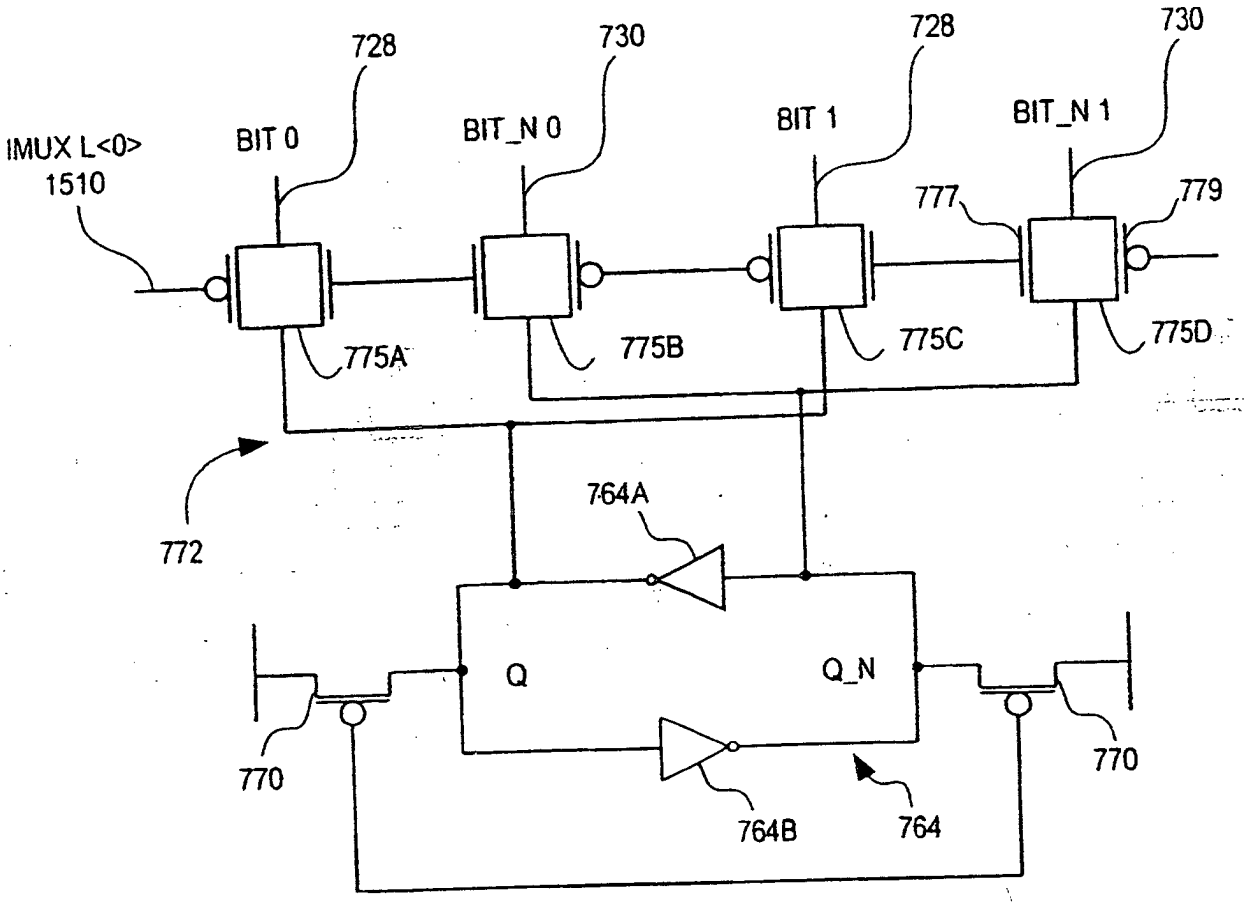
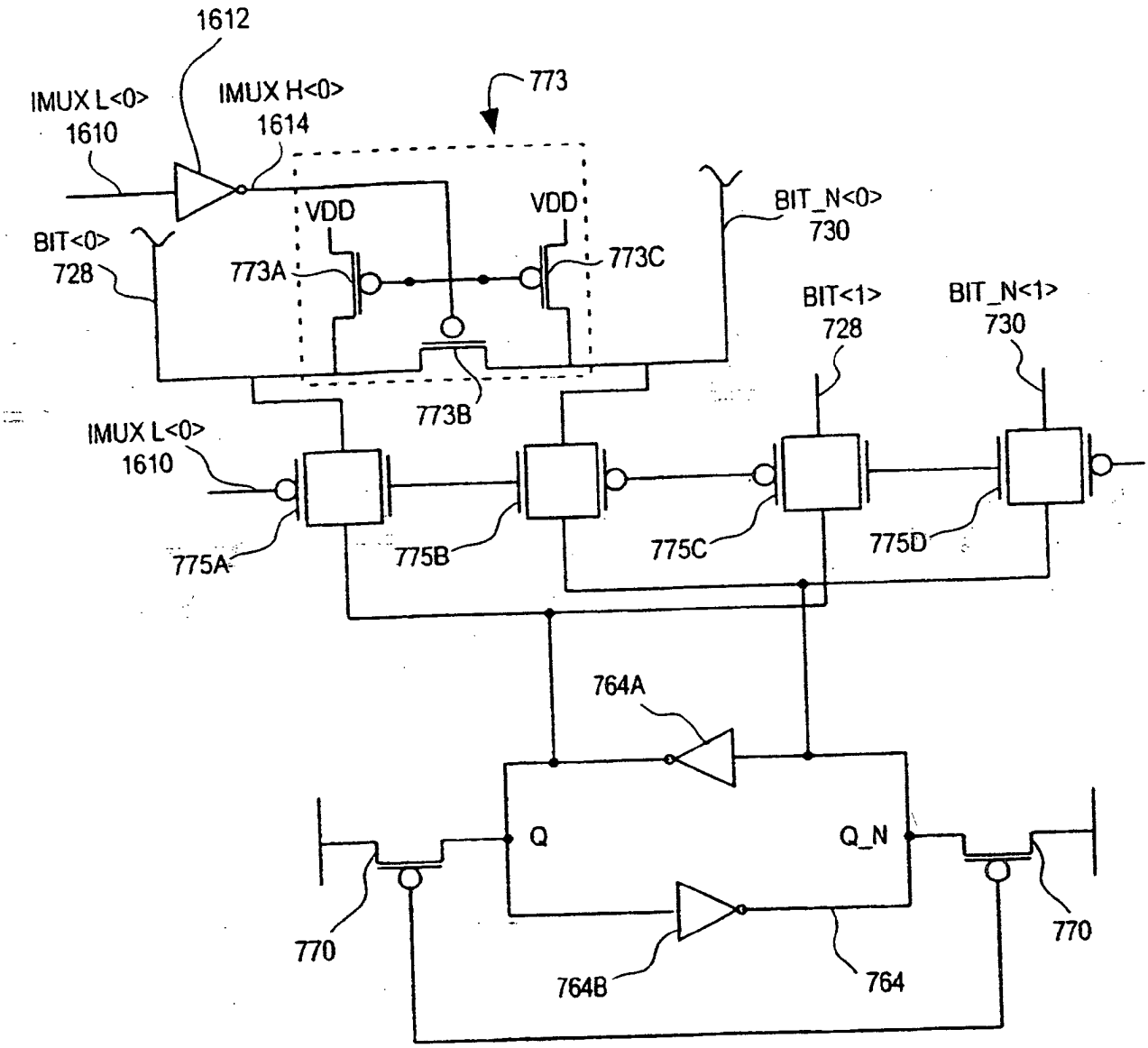

IMUX L<0>
1510
BIT 0
728
BIT_N 0
730
BIT 1
728
BIT_N 1
730
777
779
775A
775B
775C
775D
772
764A
Q
Q_N
770
770
764B
764

FIG. 16


1612
IMUX L<0>
1610
IMUX H<0>
1614
773
VDD
VDD
BIT_N<0>
730
BIT<0>
728
773A
773C
BIT<1>
728
BIT_N<1>
730
773B
IMUX L<0>
1610
775A
775B
775C
775D
764A
Q
Q_N
770
770
764B
764

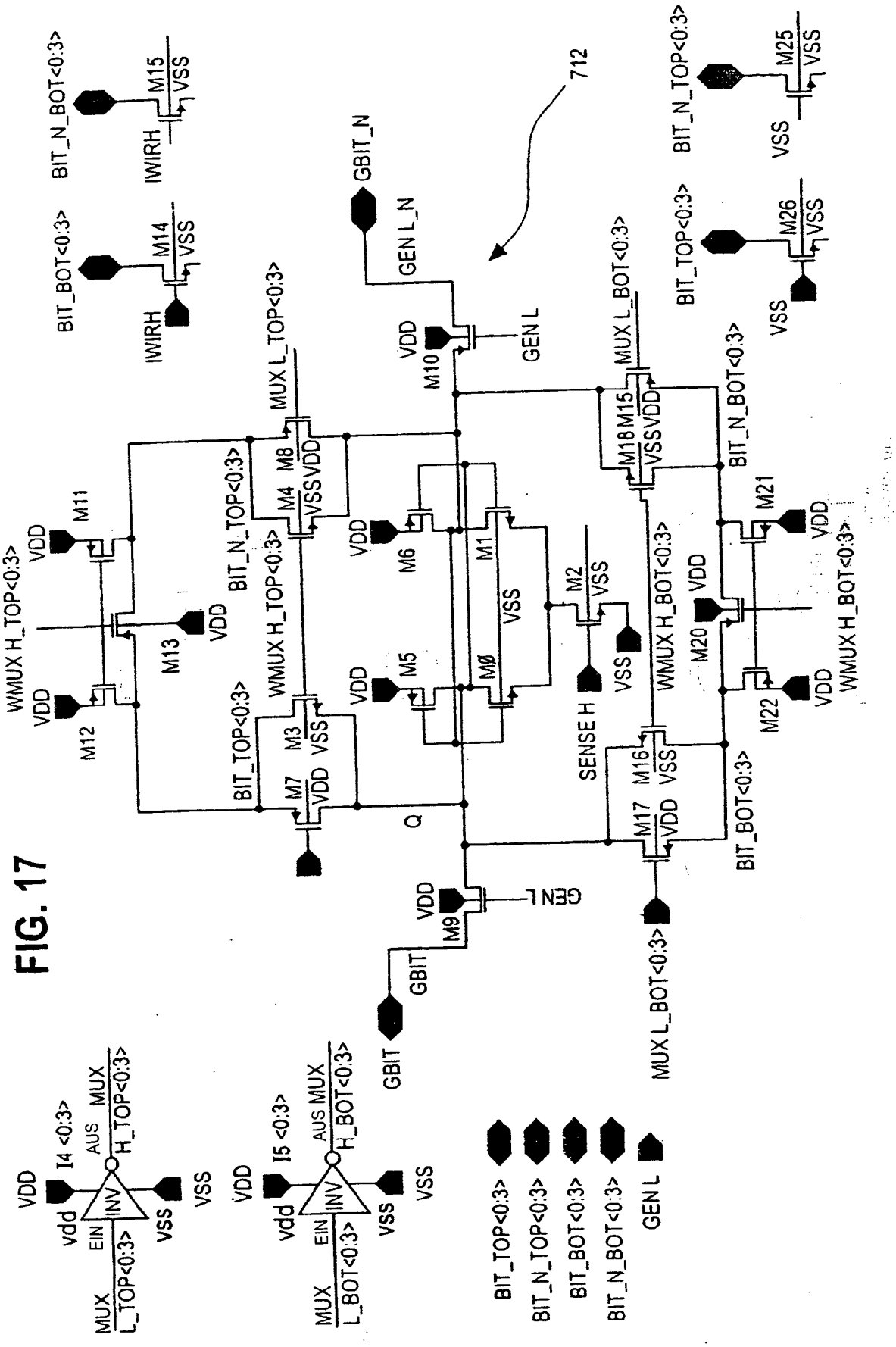
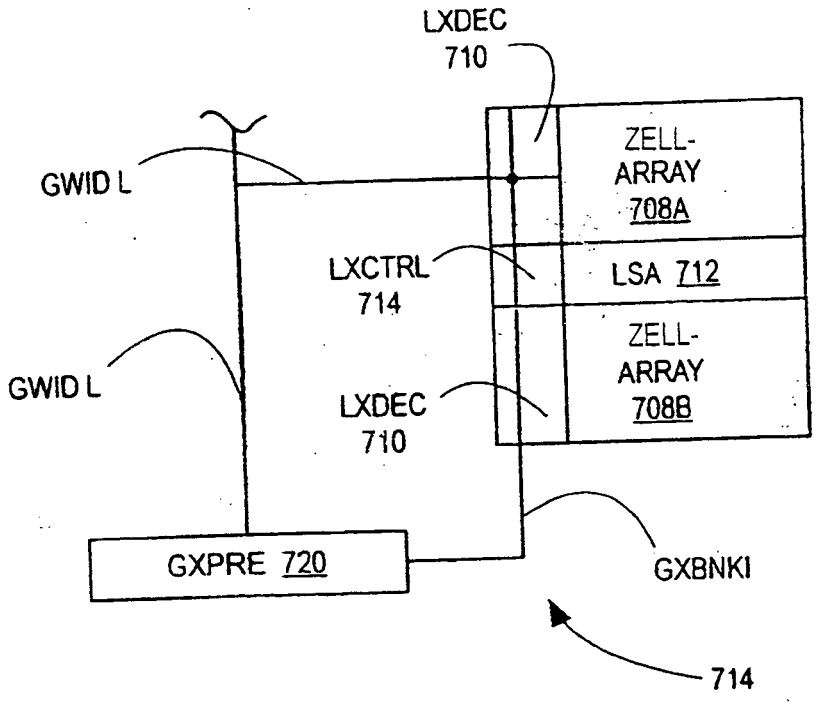

712
BIT_N_BOT<0:3>
M15
IWIRH
VSS
BIT_BOT<0:3>
M14
IWIRH
VSS
GBIT_N
GEN L_N
BIT_N_TOP<0:3>
M25
VSS
BIT_TOP<0:3>
M26
VSS
MUX L_TOP<0:3>
MUX L_BOT<0:3>
VDD
M10
GEN L
M18 M15
VSS VDD
WMUX H_TOP<0:3>
VDD
M11
M13
VDD
VDD
M12
BIT_N_TOP<0:3>
M4 M8
VSS VDD
WMUX H_TOP<0:3>
BIT_TOP<0:3>
M3
M7
VSS
VDD
VDD
M6
VDD
M5
M1
M0
VSS
M2
VSS
SENSE H
VSS
BIT_N_BOT<0:3>
M21
VDD
WMUX H_BOT<0:3>
VDD
M20
VDD
M22
VDD
WMUX H_BOT<0:3>
M16
VSS
M17
VDD
VSS
BIT_BOT<0:3>
Q
VDD
M9
GEN L
GBIT
GBIT
MUX L_BOT<0:3>
VDD
I4 <0:3>
AUS
MUX
H_TOP<0:3>
vdd
EIN
INV
VSS
VSS
MUX
L_TOP<0:3>
VDD
I5 <0:3>
AUS
MUX
H_BOT<0:3>
vdd
EIN
INV
VSS
VSS
MUX
L_BOT<0:3>
BIT_TOP<0:3>
BIT_N_TOP<0:3>
BIT_BOT<0:3>
BIT_N_BOT<0:3>
GEN L

FIG. 17

FIG. 18


LXDEC
710
ZELL-
ARRAY
708A
GWID L
LXCTRL
714
LSA 712
ZELL-
ARRAY
708B
GWID L
LXDEC
710
GXPRE 720
GXBNKI
714

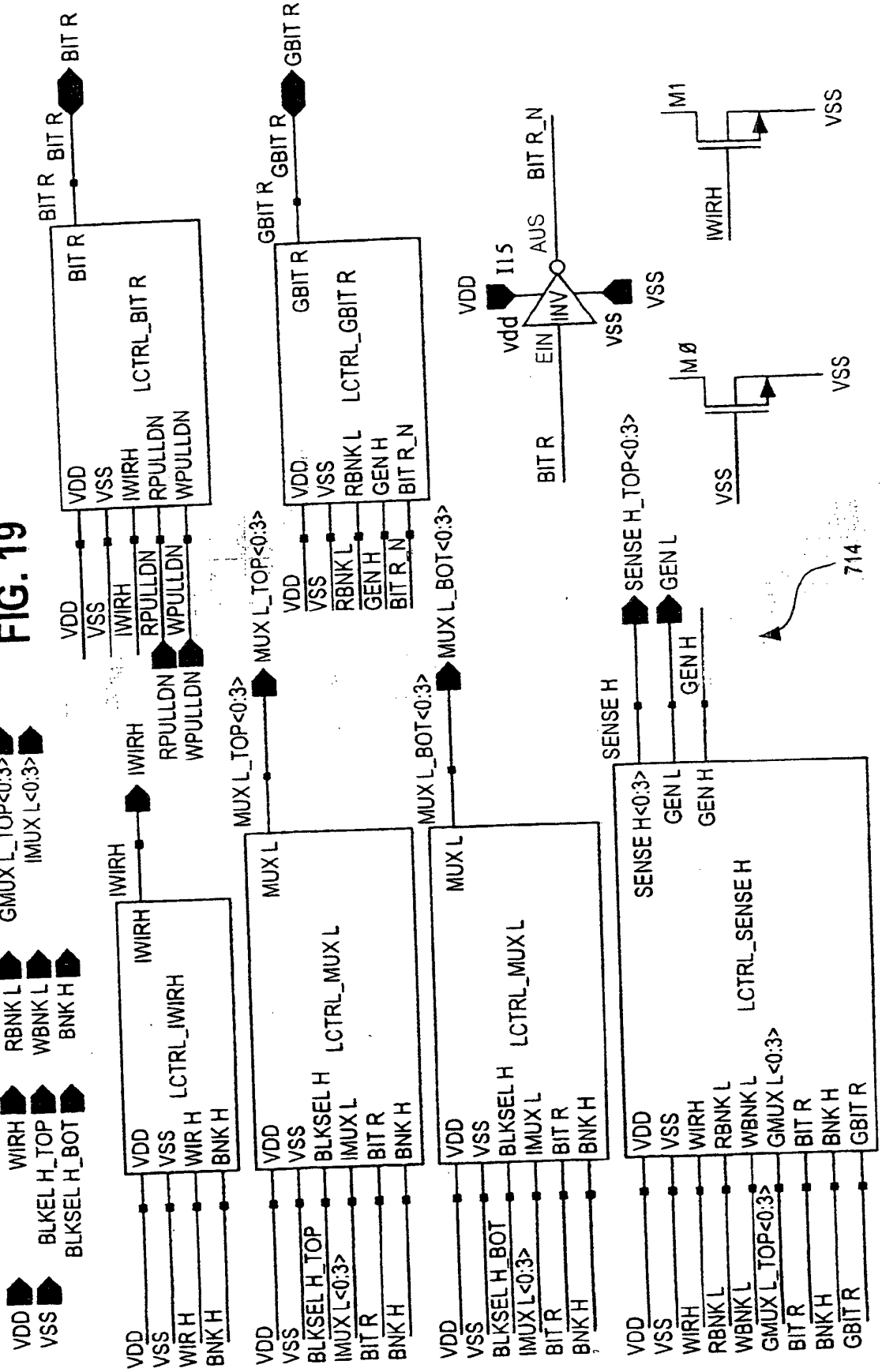

VDD
VSS
WIRH
BLKSEL H_TOP
BLKSEL H_BOT
RBNK L
WBNK L
BNK H
GMUX L_TOP<0:3>
IMUX L<0:3>
LCTRL_IWIRH
LCTRL_MUX L
LCTRL_SENSE H
LCTRL_BIT R
LCTRL_GBIT R
IWIRH
RPULLDN
WPULLDN
MUX L_TOP<0:3>
MUX L_BOT<0:3>
SENSE H_TOP<0:3>
GEN L
GEN H
BIT R
GBIT R
BIT R_N
I15
AUS
INV
EIN
vdd
M1
M0
714

FIG. 19

FIG. 20

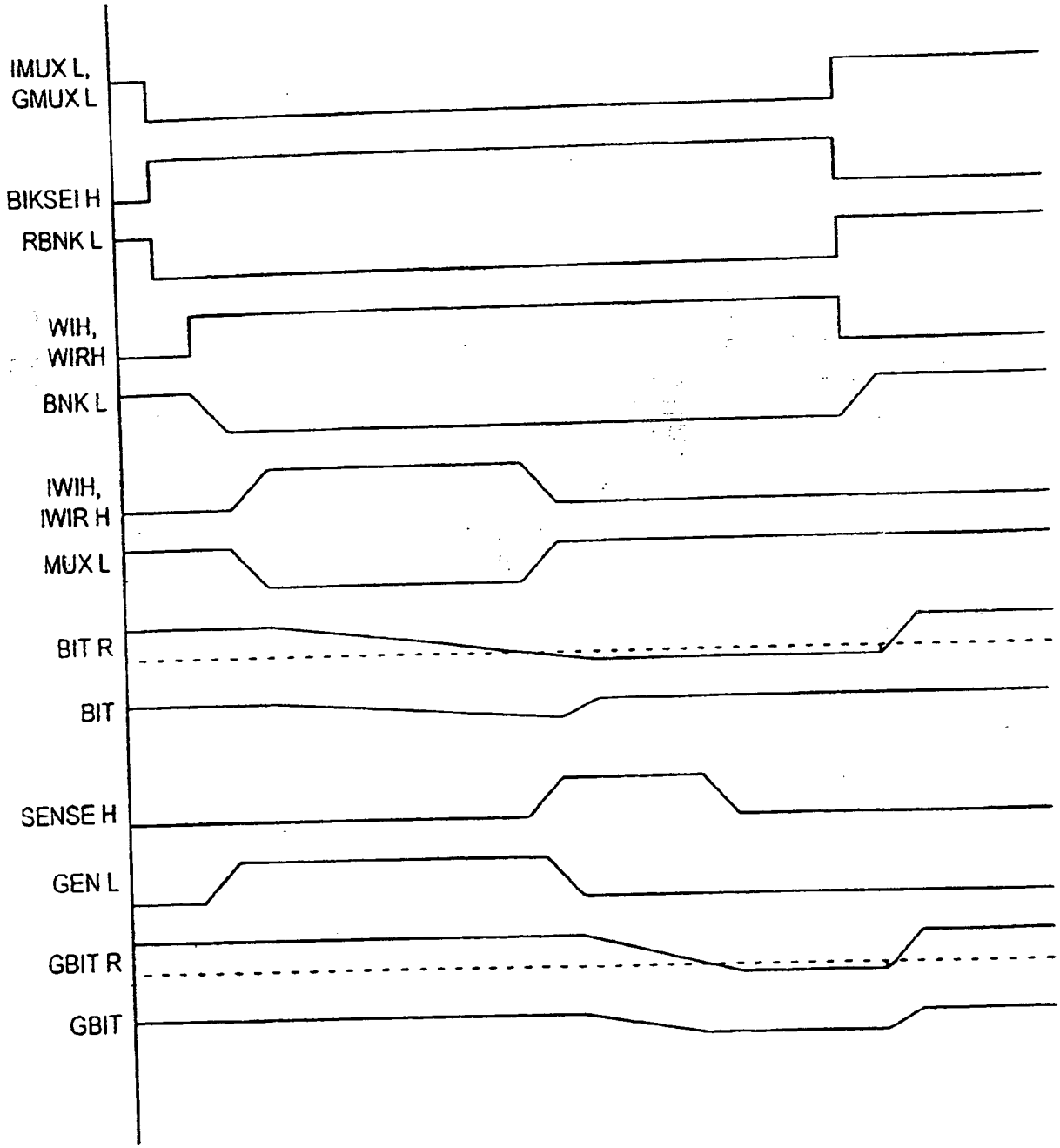

IMUX L,
GMUX L
BIKSEI H
RBNK L
WIH,
WIRH
BNK L
IWIH,
IWIR H
MUX L
BIT R
BIT
SENSE H
GEN L
GBIT R
GBIT

FIG. 21

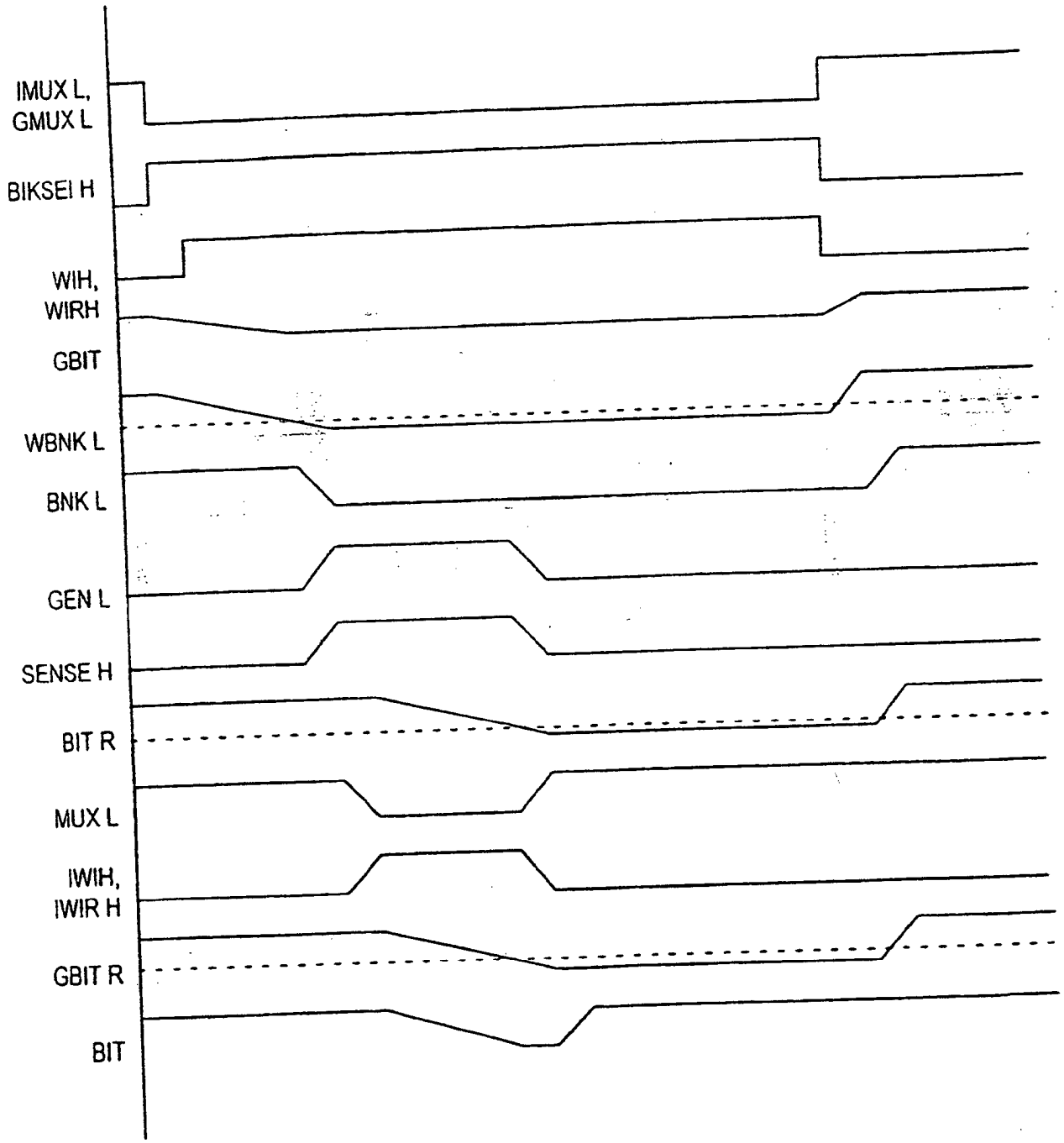

IMUX L,
GMUX L
BIKSEl H
WIH,
WIRH
GBIT
WBNK L
BNK L
GEN L
SENSE H
BIT R
MUX L
IWIH,
IWIR H
GBIT R
BIT

FIG. 22A

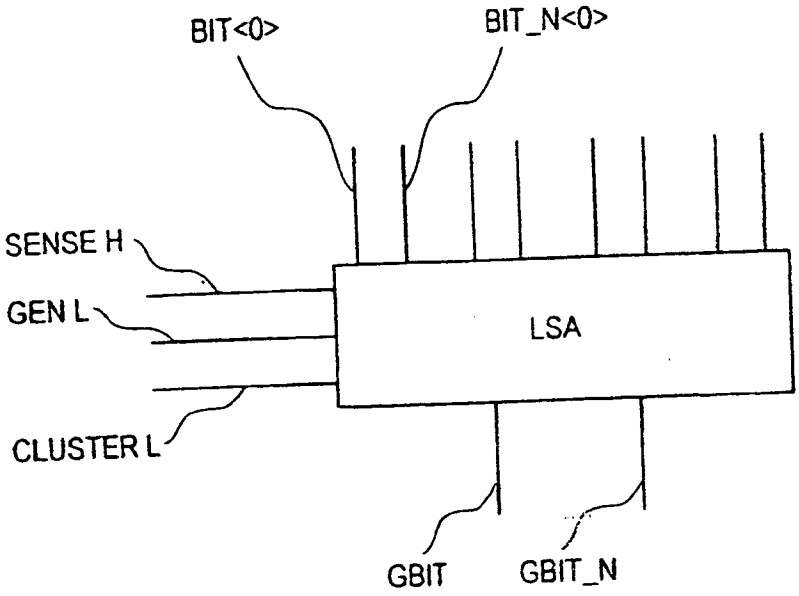


FIG. 22B

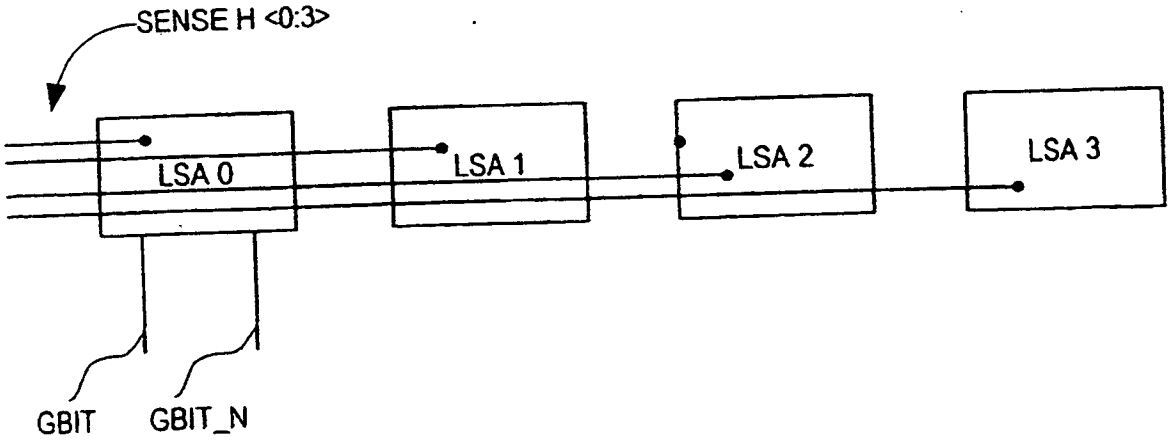


FIG. 22C

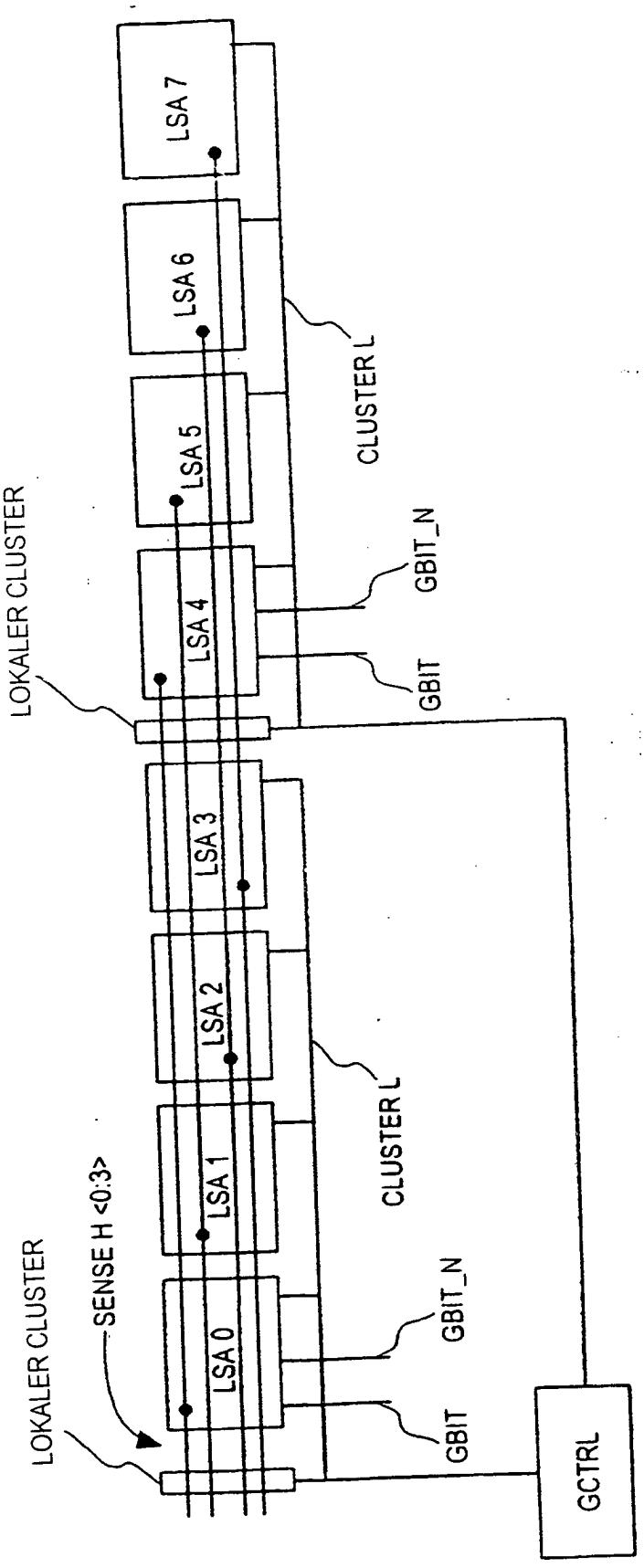

LSA 7
LSA 6
LSA 5
LSA 4
LSA 3
LSA 2
LSA 1
LSA 0
CLUSTER L
GBIT_N
GBIT
LOKALER CLUSTER
SENSE H <0:3>
LOKALER CLUSTER
CLUSTER L
GBIT_N
GBIT
GCTRL

FIG. 23

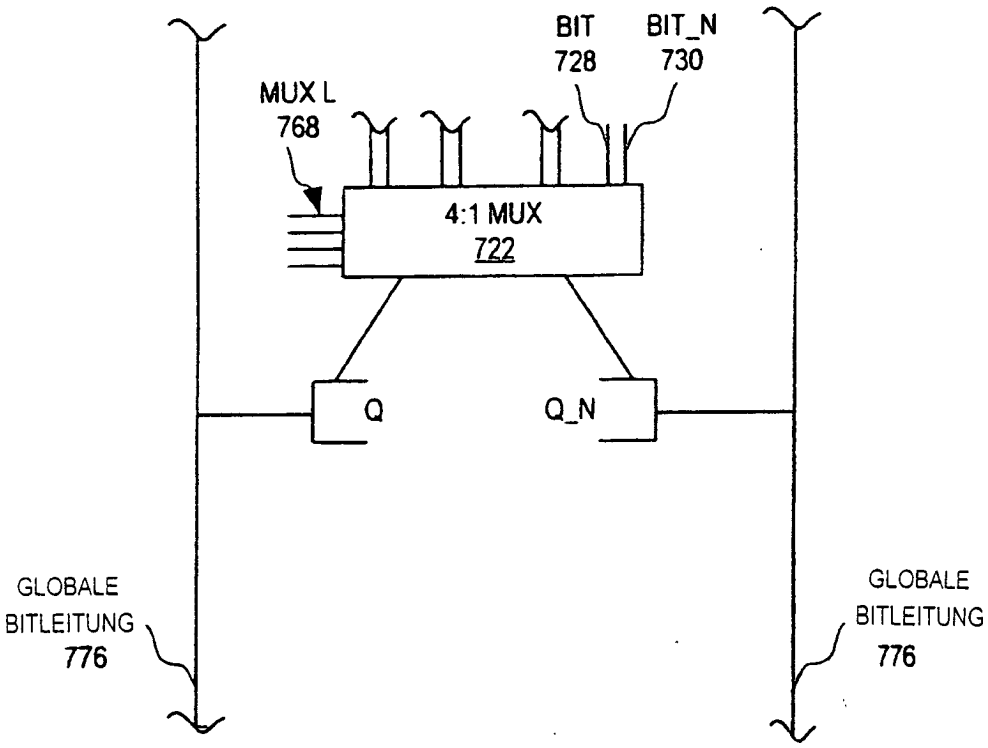

BIT
728
BIT_N
730
MUX L
768
4:1 MUX
722
Q
Q_N
GLOBALE
BITLEITUNG
776
GLOBALE
BITLEITUNG
776

FIG. 24

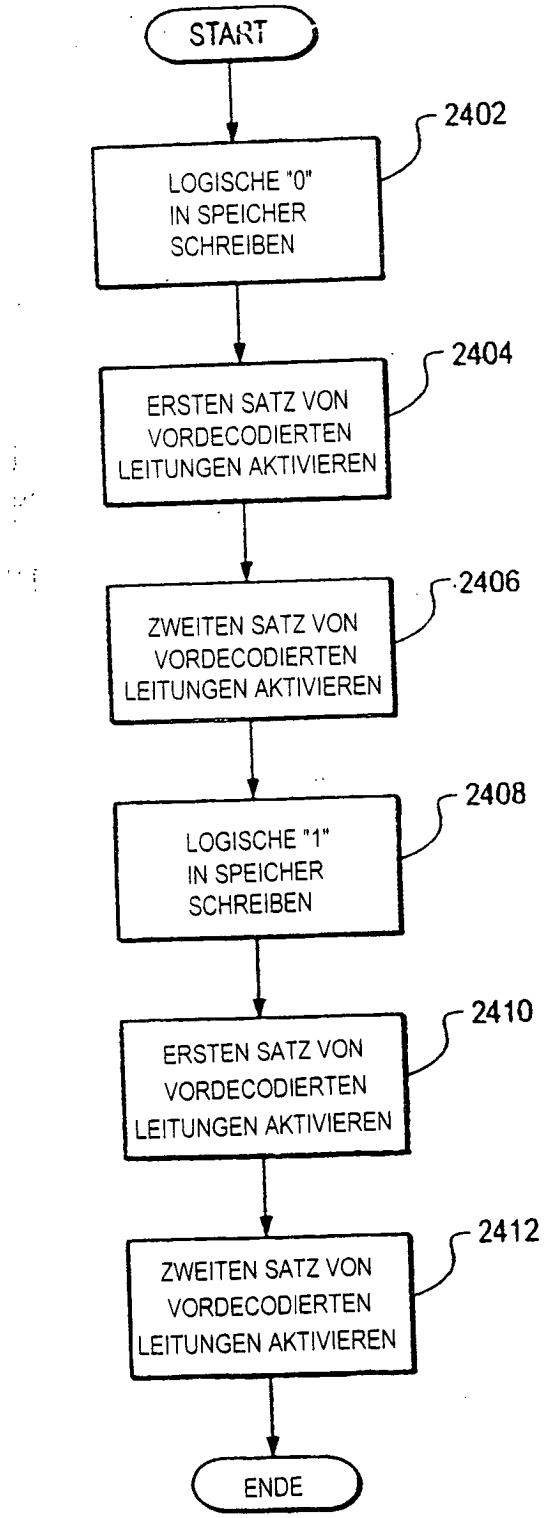

START
2402
LOGISCHE "0" IN SPEICHER SCHREIBEN
2404
ERSTEN SATZ VON VORDECODIERTEN LEITUNGEN AKTIVIEREN
2406
ZWEITEN SATZ VON VORDECODIERTEN LEITUNGEN AKTIVIEREN
2408
LOGISCHE "1" IN SPEICHER SCHREIBEN
2410
ERSTEN SATZ VON VORDECODIERTEN LEITUNGEN AKTIVIEREN
2412
ZWEITEN SATZ VON VORDECODIERTEN LEITUNGEN AKTIVIEREN
ENDE

FIG. 25

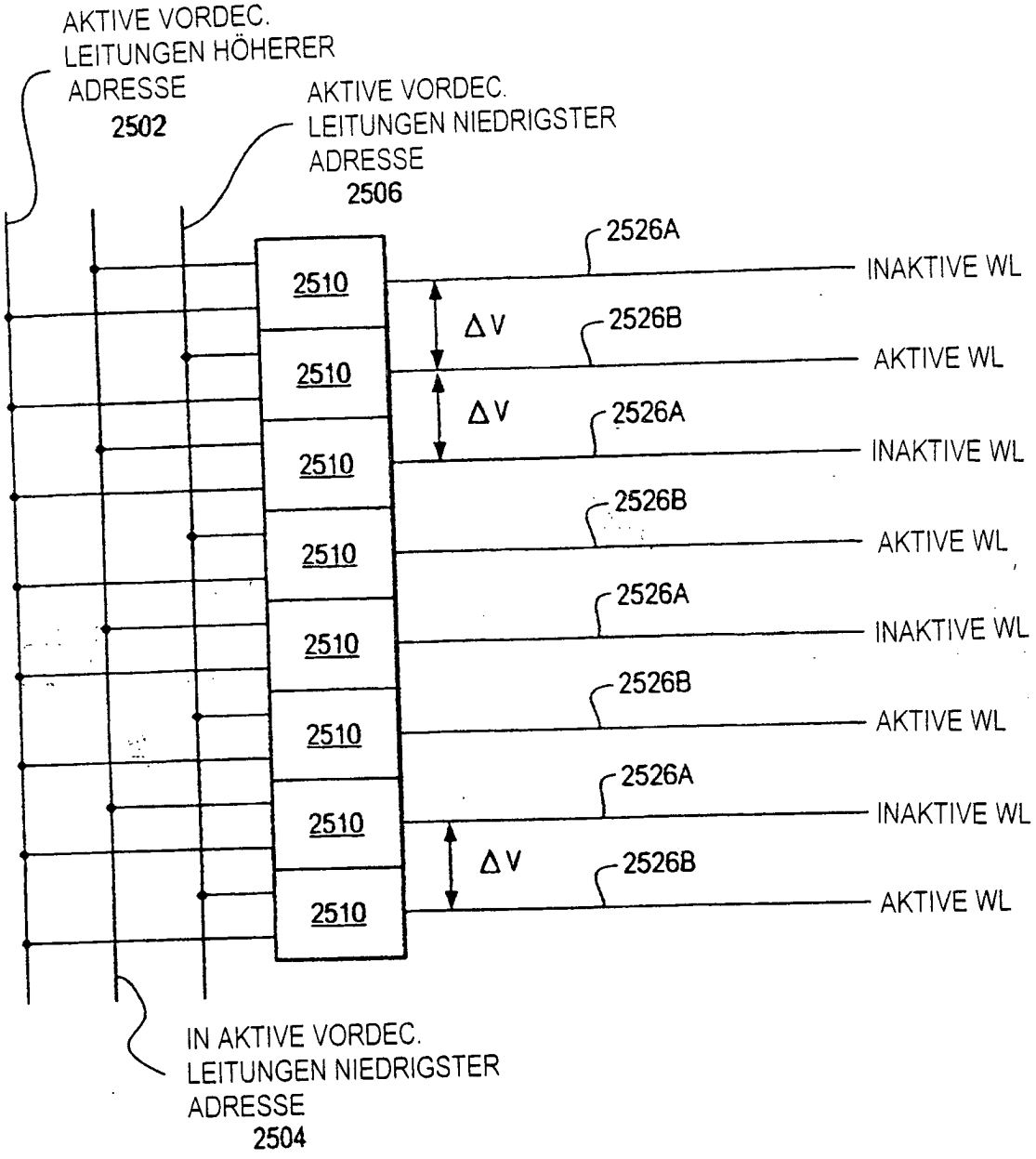
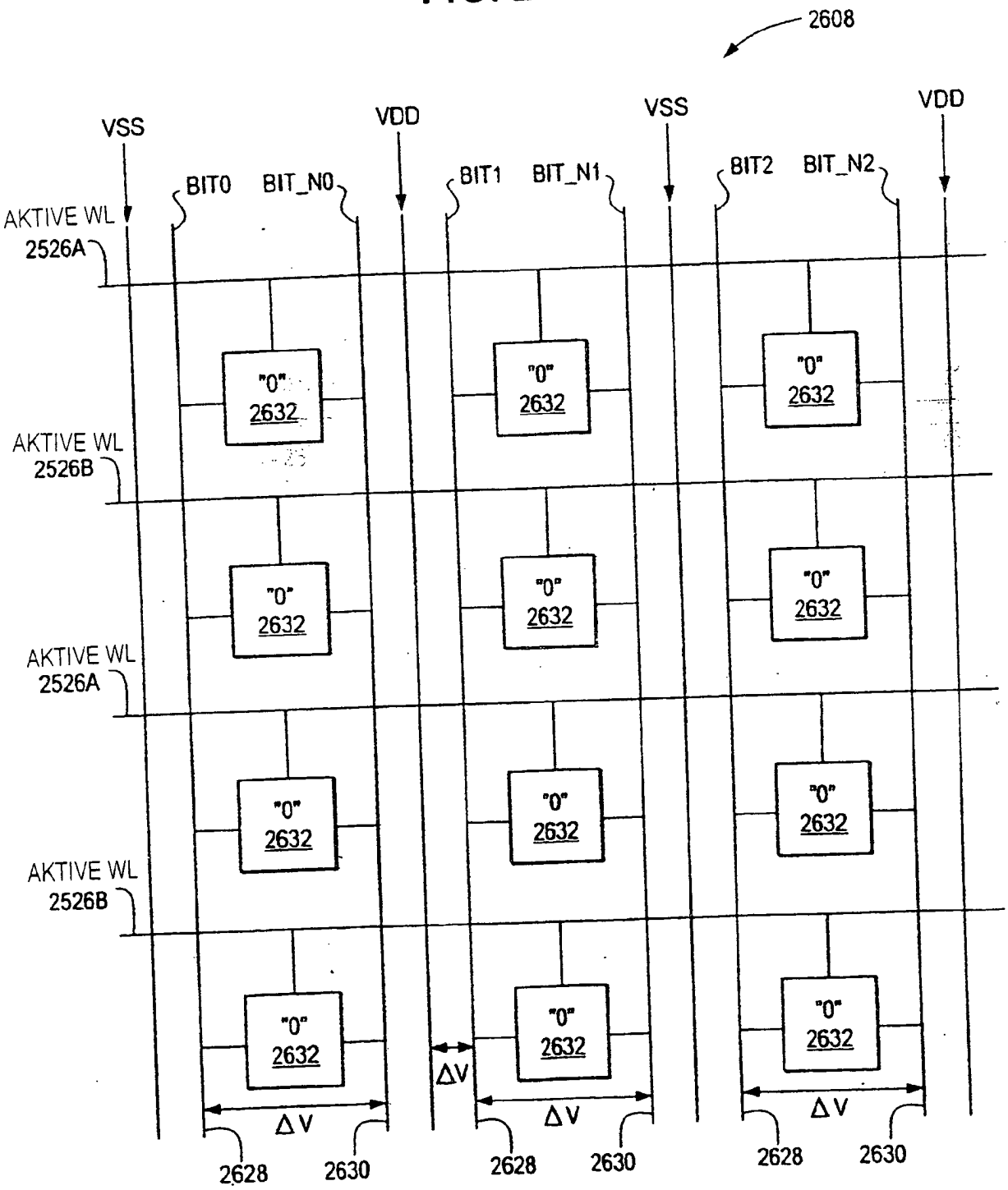

AKTIVE VORDEC. LEITUNGEN HÖHERER ADRESSE 2502
AKTIVE VORDEC. LEITUNGEN NIEDRIGSTER ADRESSE 2506
2510
2510
2510
2510
2510
2510
2510
2510
2526A
INAKTIVE WL
ΔV
2526B
AKTIVE WL
ΔV
2526A
INAKTIVE WL
2526B
AKTIVE WL
2526A
INAKTIVE WL
2526B
AKTIVE WL
2526A
INAKTIVE WL
ΔV
2526B
AKTIVE WL
IN AKTIVE VORDEC. LEITUNGEN NIEDRIGSTER ADRESSE 2504

FIG. 26


2608
VSS
VDD
VSS
VDD
BIT0
BIT_N0
BIT1
BIT_N1
BIT2
BIT_N2
AKTIVE WL
2526A
AKTIVE WL
2526B
AKTIVE WL
2526A
AKTIVE WL
2526B
"0"
2632
ΔV
2628
2630

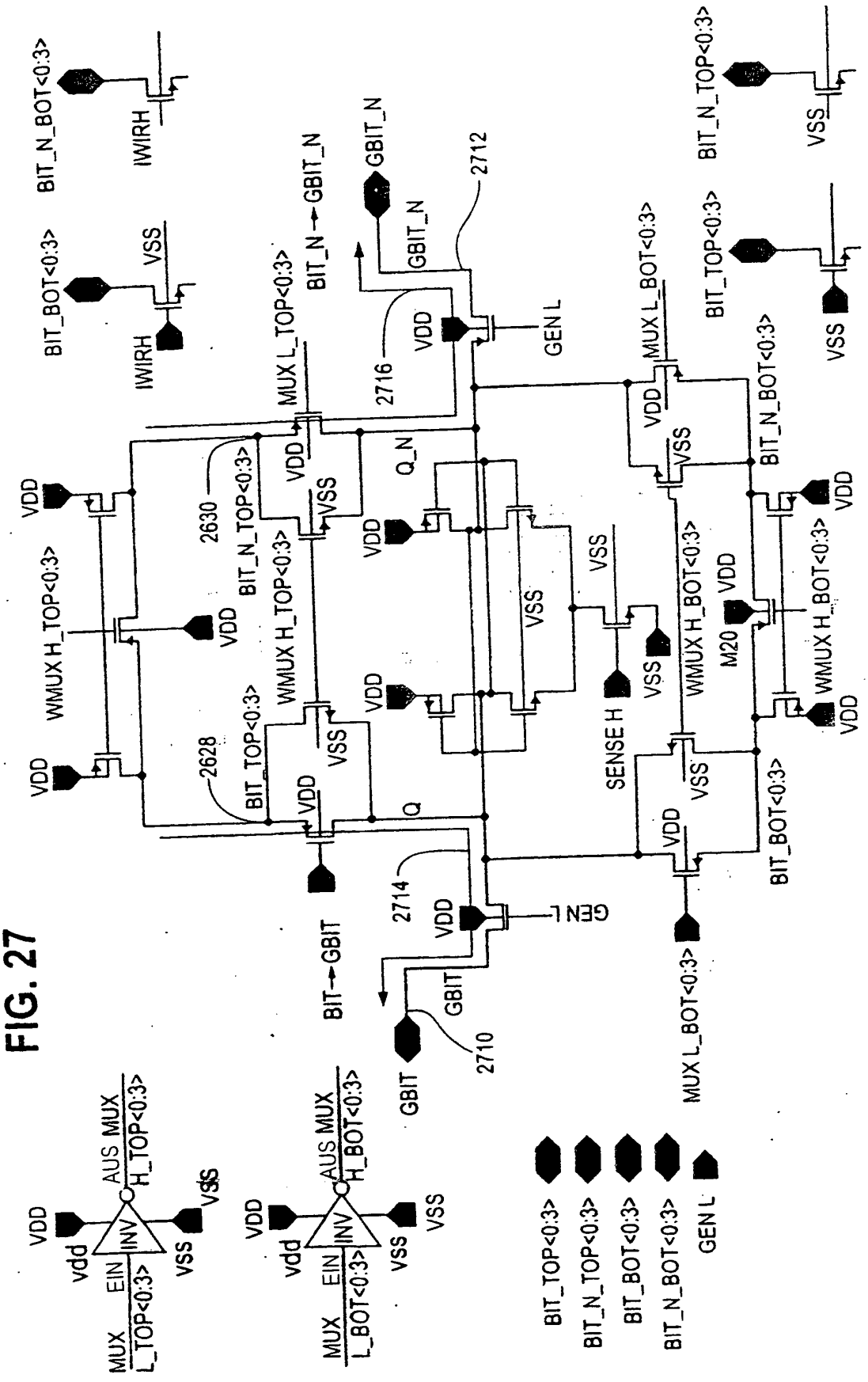

BIT_N_BOT<0:3>
IWIRH
BIT_BOT<0:3>
VSS
IWIRH
MUX L_TOP<0:3>
BIT_N→GBIT_N
GBIT_N
2712
2716
VDD
GEN L
BIT_N_TOP<0:3>
VSS
BIT_TOP<0:3>
VSS
MUX L_BOT<0:3>
VDD
VSS
BIT_N_BOT<0:3>
VDD
VDD
WMUX H_TOP<0:3>
VDD
2630
BIT_N_TOP<0:3>
VDD
VSS
Q_N
VDD
VSS
VSS
VSS
WMUX H_BOT<0:3>
M20
VDD
WMUX H_BOT<0:3>
VDD
VDD
2628
BIT_TOP<0:3>
VDD
VSS
VDD
Q
SENSE H
VSS
VSS
VDD
BIT_BOT<0:3>
2714
VDD
GEN L
BIT→GBIT
GBIT
GBIT
2710
MUX L_BOT<0:3>
AUS MUX
H_TOP<0:3>
VDD
vdd
EIN
INV
VSS
VSS
MUX
L_TOP<0:3>
AUS MUX
H_BOT<0:3>
VDD
vdd
EIN
INV
VSS
VSS
MUX
L_BOT<0:3>
BIT_TOP<0:3>
BIT_N_TOP<0:3>
BIT_BOT<0:3>
BIT_N_BOT<0:3>
GEN L

FIG. 27

FIG. 28
STAND DER TECHNIK

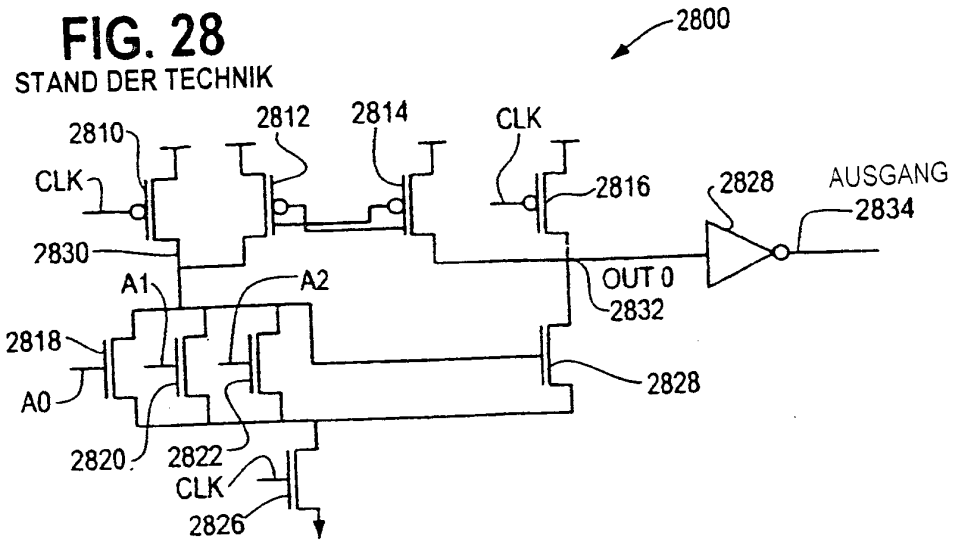


FIG. 29

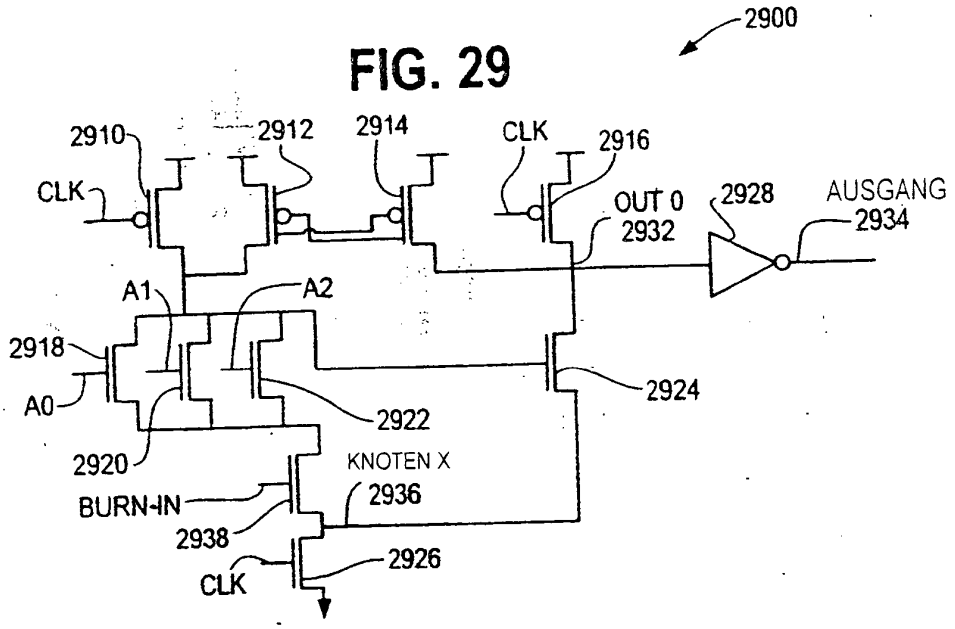


FIG. 30

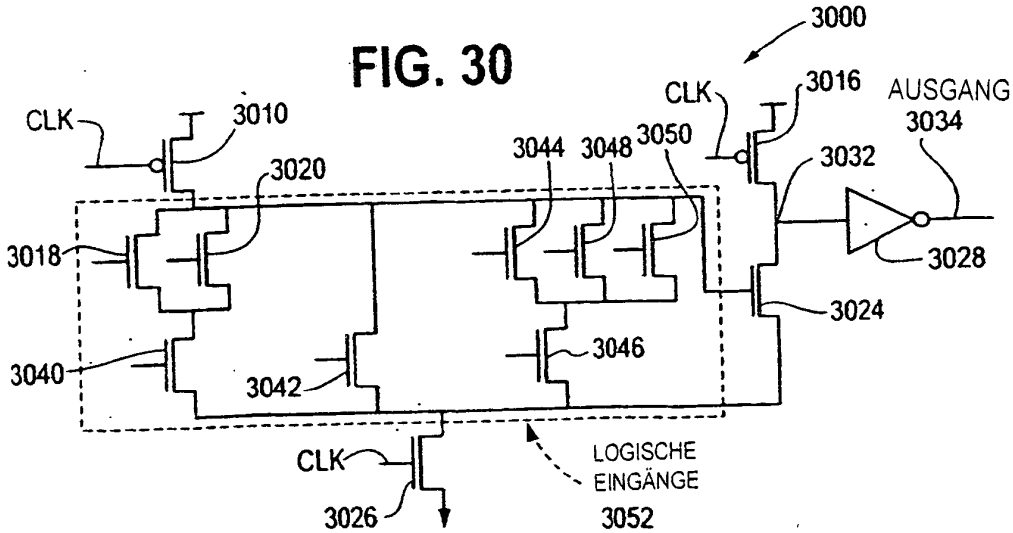


FIG. 31

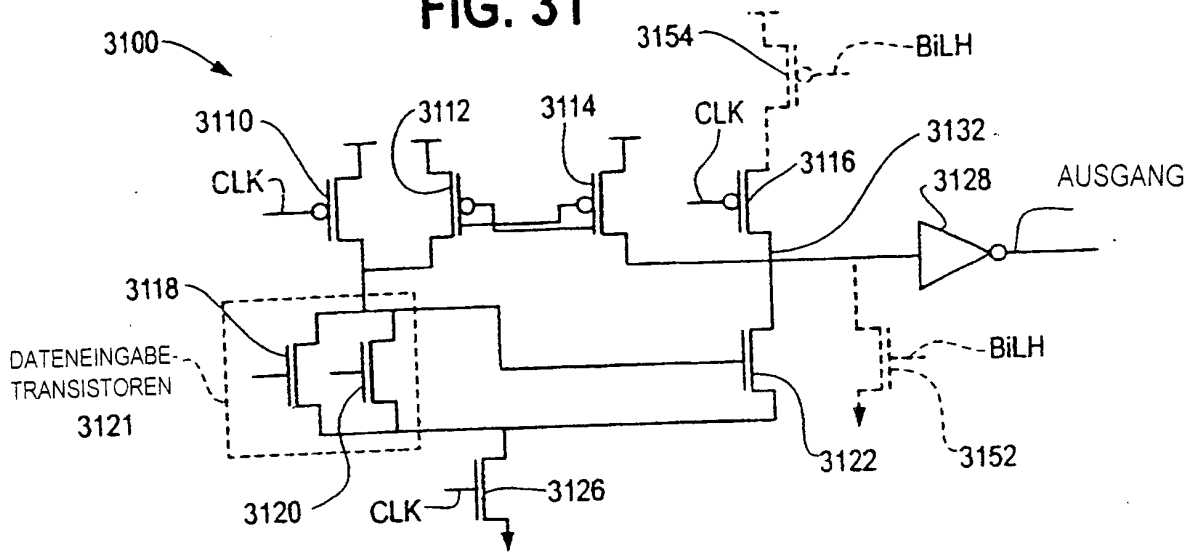


FIG. 32

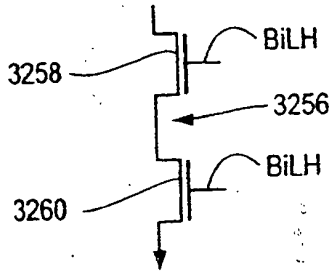


FIG. 33