



(10) **DE 10 2007 008 180 B4 2018.12.20**

(12) **Patentschrift**

(21) Aktenzeichen: **10 2007 008 180.6**  
 (22) Anmeldetag: **12.02.2007**  
 (43) Offenlegungstag: **13.09.2007**  
 (45) Veröffentlichungstag  
 der Patenterteilung: **20.12.2018**

(51) Int Cl.: **G11C 29/52 (2006.01)**  
**G11C 29/42 (2006.01)**  
**G11C 7/10 (2006.01)**  
**G06F 11/08 (2006.01)**  
**G06F 11/10 (2006.01)**  
**H03M 13/29 (2006.01)**

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(30) Unionspriorität:  
**10-2006-0013898 13.02.2006 KR**

(72) Erfinder:  
**Chung, Hoe-Ju, Yongin, Kyonggi, KR; Kim, Kyu-Hyoun, Suwon, Kyonggi, KR**

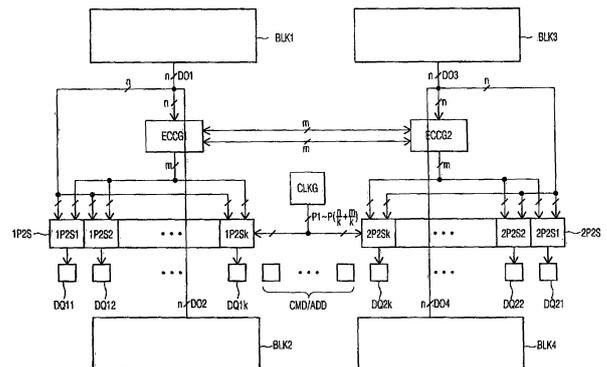
(73) Patentinhaber:  
**Samsung Electronics Co., Ltd., Suwon-si, Gyeonggi-do, KR**

(56) Ermittelter Stand der Technik:  
**US 6 076 182 A**

(74) Vertreter:  
**Patentanwälte Ruff, Wilhelm, Beier, Dauster & Partner mbB, 70174 Stuttgart, DE**

(54) Bezeichnung: **Halbleiterspeicherelement, Speichersystem und Daten-Sende-/Empfangssystem**

(57) Hauptanspruch: Halbleiterspeicherelement, aufweisend:  
 einen ersten Speicherzellenfeldblock (BLK1), der erste Daten (DO1) erzeugt;  
 einen zweiten Speicherzellenfeldblock (BLK3), der zweite Daten (DO3) erzeugt;  
 einen ersten Fehlererkennungscodegenerator (ECCG1), der einen ersten Fehlererkennungscode (1ec1) für die ersten Daten (DO1) erzeugt und der einen Anteil von Bits des ersten Fehlererkennungscode (1ec1) mit einem Anteil von Bits eines zweiten Fehlererkennungscode (2ec1) kombiniert, um ein erstes endgültiges Fehlererkennungscode (fec1) zu erzeugen; und  
 einen zweiten Fehlererkennungscodegenerator (ECCG2), der den zweiten Fehlererkennungscode (2ec1) für die zweiten Daten (DO3) erzeugt und der verbleibende Bits des zweiten Fehlererkennungscode (2ec1), die sich von dem Anteil von Bits des zweiten Fehlererkennungscode (2ec1) unterscheiden, mit verbleibenden Bits des ersten Fehlererkennungscode (1ec1), die sich von dem Anteil von Bits des ersten Fehlererkennungscode (1ec1) unterscheiden, kombiniert, um ein zweites endgültiges Fehlererkennungscode zu erzeugen.



## Beschreibung

**[0001]** Die Erfindung betrifft ein Halbleiterspeicherelement; ein Speichersystem und ein Daten-Sende-/Empfangssystem.

**[0002]** Ein herkömmliches Halbleiterspeicherelement mit einer ODIC-Padstruktur weist eine Struktur auf, bei der Datenpads auf beiden Seiten des Halbleiterspeicherelements angeordnet sind und bei der ein Befehlspad an einer Innenseite des Halbleiterspeicherelements angeordnet ist. Daten, die über das Datenpad, welches auf einer Seite angeordnet ist, eingegeben und ausgegeben werden, können nur durch einen Speicherzellenfeldblock eingegeben und ausgegeben werden, der auf einer Seite angeordnet ist, und Daten, die über das Datenpad eingegeben und ausgegeben werden, das auf der anderen Seite angeordnet ist, können nur durch einen Speicherzellenfeldblock eingegeben und ausgegeben werden, der auf der anderen Seite angeordnet ist.

**[0003]** Fig. 1 ist ein Blockschaltbild eines herkömmlichen Halbleiterspeicherelements, das eine ODIC-Padstruktur aufweist, welches beispielsweise vier Speicherzellenfeldblöcke **BLK1** bis **BLK4** umfassen kann.

**[0004]** Bezugnehmend auf Fig. 1 können eine erste Gruppe von Datenpads **DQ1**, Adress-/Befehl-Signalanlegepads **CMD/ADD** und eine zweite Gruppe von Datenpads **DQ2** in einer Reihe zwischen einem Bereich, in dem die Speicherzellenblöcke **BL1** und **BLK3** angeordnet sind, und einem Bereich, in dem die Speicherzellenblöcke **BLK2** und **BLK4** angeordnet sind, angeordnet sein. Wie in Fig. 1 gezeigt, kann das Halbleiterspeicherelement, welches die ODIC-Padstruktur aufweist, eine erste Gruppe von Datenpads **DQ1** und eine zweite Gruppe von Datenpads **DQ2** aufweisen, die auf beiden Seiten des Halbleiterspeicherelements angeordnet sind, und die Adress-/Befehl-Signalanlegepads **CMD/ADD** können in der Mitte des Halbleiterspeicherelements angeordnet sein.

**[0005]** Bezugnehmend auf Fig. 1 können Daten **DO1** und **DO2**, die in die Speicherzellenfeldblöcke **BLK1** und **BLK2** eingegeben oder aus den Speicherzellenfeldblöcken **BLK1** und **BLK2** ausgegeben werden, über die erste Gruppe von Datenpads **DQ1** eingegeben/ausgegeben werden und Daten **DO3** und **DO4**, die in die Speicherzellenfeldblöcke **BLK3** und **BLK4** eingegeben oder aus den Speicherzellenfeldblöcken **BLK3** und **BLK4** ausgegeben werden, können über die zweite Gruppe von Datenpads **DQ2** eingegeben/ausgegeben werden.

**[0006]** Herkömmliche Halbleiterspeicherelemente können Daten sowie einen Fehlererkennungscode übertragen, der während einer Datenübertragung zu

den Daten hinzugefügt wird. Zu diesem Zweck können herkömmliche Halbleiterspeicherelemente einen Fehlererkennungscodegenerator aufweisen. Ein Fehlererkennungscodegenerator kann Fehlererkennungscode für Daten betreffend alle Bits erzeugen, die von den Speicherzellenfeldblöcken **BLK1** bis **BLK4** ausgegeben werden. Wenn Daten selektiv von den Speicherzellenfeldblöcken **BLK1** und **BLK3** oder den Speicherzellenfeldblöcken **BLK2** und **BLK4** ausgegeben werden, kann der Generator beispielsweise einen Fehlererkennungscode für die Daten **DO1** und **DO4** oder für die Daten **DO2** und **DO3** erzeugen.

**[0007]** Wenn jedoch der Fehlererkennungscodegenerator in einem Bereich A zwischen Bereichen angeordnet ist, in denen die Blöcke **BLK1** bis **BLK4** angeordnet sind, und wenn die Daten von den Daten **DO1** und **DO4** oder den Daten **DO2** und **DO3** ausgegeben werden, sollten die Daten **DO1** und **DO4** oder die Daten **DO2** und **DO3** in den Fehlererkennungscodegenerator eingegeben werden. Dementsprechend sind Signalleitungen zum Übertragen der Daten **DO1** und **DO2** und Signalleitungen zum Übertragen der Daten **DO3** und **DO4** regelmäßig auf der Seite des Fehlererkennungscodegenerators angeordnet. Dies führt nicht nur zu einer Vergrößerung der Layoutfläche herkömmlicher Halbleiterspeicherelemente, sondern führt auch zu einer Verzögerung der Signalübertragung aufgrund einer vergrößerten Länge der Signalleitung oder Signalleitungen, die dazu verwendet wird/werden, die Daten **DO1**, **DO2**, **DO3** und **DO4** an dem Fehlererkennungscodegenerator bereitzustellen.

**[0008]** Vergleichbare Probleme treten auch in herkömmlichen Halbleiterspeicherelementen auf, selbst wenn der Fehlererkennungscodegenerator in einem Bereich B zwischen Bereichen angeordnet ist, in denen die Speicherzellenfeldblöcke **BLK1** und **BLK2** angeordnet sind, oder in einem Bereich C zwischen Bereichen, in denen die Speicherzellenfeldblöcke **BLK3** und **BLK4** angeordnet sind.

Die US 6 076 182 A offenbart ein Halbleiterspeicherelement, das eine Mehrzahl von Fehlererkennungscodegeneratoren aufweist, die jeweils für Teile bzw. Subwörter eines Datenworts Fehlererkennungscode berechnen. Die Subwörter und die Fehlererkennungscode werden in mehreren Speicherbausteinen eines Speichermoduls abgelegt.

**[0009]** Der Erfindung liegt das technische Problem zugrunde, ein Halbleiterspeicherelement, ein Speichersystem und ein Daten-Sende-/Empfangssystem anzugeben, die in der Lage sind, einen Fehlererkennungscode zu erzeugen ohne die Fehlererkennungs-fähigkeit zu beeinträchtigen und ohne eine Layoutfläche zu vergrößern.

**[0010]** Die Erfindung löst dieses Problem mittels eines Halbleiterspeicherelements mit den Merkmalen

des Patentanspruchs 1 oder des Patentanspruchs 26, eines Speichersystems mit den Merkmalen des Patentanspruchs 38 oder des Patentanspruchs 40 und eines Daten-Sende-/Empfangssystems mit den Merkmalen des Patentanspruchs 42.

**[0011]** Vorteilhafte Ausgestaltungen der Erfindung sind in den Unteransprüchen angegeben, deren Wortlaut hiermit durch Bezugnahme in die Beschreibung aufgenommen wird, um unnötige Textwiederholungen zu vermeiden.

**[0012]** Gemäß einer beispielhaften Ausgestaltung kann sowohl eine erste als auch eine zweite Fehlererkennungscodeerzeugungsschaltung eine XOR-Schaltung umfassen, die unter Verwendung eines Cyclic-Redundancy-Check-Codes implementiert sind und die dasselbe Fehlererkennungscodeerzeugungspolynom verwenden. Die erste und die zweite Fehlererkennungscodeerzeugungsschaltung können jeweils eine XOR-Schaltung umfassen, die unter Verwendung eines Cyclic-Redundancy-Check-Codes implementiert ist und die ein anderes Fehlererkennungscodeerzeugungspolynom als die jeweils andere Schaltung verwendet.

**[0013]** Eine weitere beispielhafte Ausgestaltung schafft ein Speichersystem, welches weiterhin ein Halbleiterspeicherelement, eine Speichersteuereinheit und einen Fehlerdetektor umfassen kann. Das Halbleiterspeicherelement kann umfassen: einen ersten Speicherzellenfeldblock, der  $2n$  Bits erster Daten in einem ersten Bitstrukturmodus erzeugt und der  $n$  Bits der ersten Daten in einem zweiten Bitstrukturmodus erzeugt; einen zweiten Speicherzellenfeldblock, der  $2n$  Bits zweiter Daten in dem ersten Bitstrukturmodus erzeugt und der  $n$  Bits der zweiten Daten in dem zweiten Bitstrukturmodus erzeugt; einen ersten Fehlererkennungscodegenerator, der die oberen  $n$  Bits der  $2n$  Bits erster Daten in dem ersten Bitstrukturmodus und einen ersten Fehlererkennungscode für die oberen  $n$  Bits der  $2n$  Bits erster Daten in dem zweiten Bitstrukturmodus erzeugt und der einen Anteil von Bits des ersten Fehlererkennungscodees mit einem Anteil von Bits eines dritten Fehlererkennungscodees kombiniert, um einen Anteil von Bits eines ersten endgültigen Fehlererkennungscodees zu erzeugen; einen zweiten Fehlererkennungscodegenerator, der einen zweiten Fehlererkennungscode für untere  $n$  Bits der  $2n$  Bits erster Daten in dem ersten Bitstrukturmodus erzeugt und der einen Anteil von Bits des zweiten Fehlererkennungscodees mit einem Anteil von Bits eines vierten Fehlererkennungscodees kombiniert, um die verbleibenden Bits in Ergänzung zu dem Anteil von Bits des ersten endgültigen Fehlererkennungscodees zu erzeugen; einen dritten Fehlererkennungscodegenerator, der obere  $n$  Bits der  $2n$  Bits zweiter Daten in dem ersten Bitstrukturmodus und den dritten Fehlererkennungscode für die oberen  $n$  Bits der  $2n$  Bits zweiter Daten in dem zweiten

Bitstrukturmodus erzeugt und der die verbleibenden Bits in Ergänzung zu dem Anteil von Bits des dritten Fehlererkennungscodees mit den verbleibenden Bits in Ergänzung zu dem Anteil von Bits des ersten Fehlererkennungscodees kombiniert, um einen Anteil von Bits eines zweiten endgültigen Fehlererkennungscodees zu erzeugen; und einen vierten Fehlererkennungscodegenerator, der einen vierten Fehlererkennungscode für die unteren  $n$  Bits der  $2n$  Bits zweiter Daten in dem ersten Bitstrukturmodus erzeugt und der die verbleibenden Bits in Ergänzung zu dem Anteil von Bits des vierten Fehlererkennungscodees mit den verbleibenden Bits in Ergänzung zu dem Anteil von Bits des zweiten Fehlererkennungscodees kombiniert, um die verbleibenden Bits in Ergänzung zu dem Anteil von Bits des zweiten endgültigen Fehlererkennungscodees zu erzeugen. Die Speichersteuereinheit kann einen fünften Fehlererkennungscodegenerator aufweisen, der die oberen  $n$  Bits der  $2n$  Bits erster Daten, die von dem Halbleiterspeicherelement in dem ersten Bitstrukturmodus ausgegeben werden, und einen fünften Fehlererkennungscode für die  $n$  Bits der ersten Daten, die von dem Halbleiterspeicherelement in dem zweiten Bitstrukturmodus ausgegeben werden, erzeugt und der einen Anteil von Bits des fünften Fehlererkennungscodees mit einem Anteil von Bits eines siebten Fehlererkennungscodees kombiniert, um einen Anteil von Bits eines dritten endgültigen Fehlererkennungscodees zu erzeugen; einen sechsten Fehlererkennungscodegenerator, der einen sechsten Fehlererkennungscode für die unteren  $n$  Bits der  $2n$  Bits erster Daten in dem ersten Bitstrukturmodus erzeugt und der einen Anteil von Bits des sechsten Fehlererkennungscodees mit einem Anteil von Bits eines achten Fehlererkennungscodees kombiniert, um die verbleibenden Bits in Ergänzung zu dem Anteil von Bits des dritten endgültigen Fehlererkennungscodees zu erzeugen; einen siebten Fehlererkennungscodegenerator, der obere  $n$  Bits der  $2n$  Bits zweiter Daten, die von dem Halbleiterspeicherelement in dem ersten Strukturmodus ausgegeben werden, und den siebten Fehlererkennungscode für  $n$  Bits der zweiten Daten in dem zweiten Bitstrukturmodus erzeugt und der die verbleibenden Bits in Ergänzung zu dem Anteil von Bits des siebten Fehlererkennungscodees mit den verbleibenden Bits in Ergänzung zu dem Anteil von Bits des fünften Fehlererkennungscodees kombiniert, um einen Anteil von Bits eines vierten endgültigen Fehlererkennungscodees zu erzeugen; und einen achten Fehlererkennungscodegenerator, der den achten Fehlererkennungscode für untere  $n$  Bits der  $2n$  Bits zweiter Daten in dem ersten Bitstrukturmodus erzeugt und der die verbleibenden Bits in Ergänzung zu dem Anteil von Bits des achten Fehlererkennungscodees mit den verbleibenden Bits in Ergänzung zu dem Anteil von Bits des sechsten Fehlererkennungscodees kombiniert, um die verbleibenden Bits in Ergänzung zu dem Anteil von Bits des vierten endgültigen Fehlererkennungscodees zu erzeugen. Der Fehlererkennungscodegenerator kann das erste

bis vierte endgültige Fehlererkennungssignal vergleichen, die von dem Halbleiterspeicherelement ausgegeben werden, um ein Fehlererkennungssignal zu erzeugen.

**[0014]** Gemäß einer beispielhaften Ausgestaltung kann der erste bis achte Fehlererkennungscodegenerator unter Verwendung eines Cyclic-Redundancy-Check-Codes implementiert sein. Sowohl der erste als auch der fünfte Fehlererkennungscodegenerator kann eine erste XOR-Schaltung aufweisen, die dasselbe erste Fehlererkennungscodeerzeugungspolynom verwendet. Sowohl der zweite als auch der sechste Fehlererkennungscodegenerator kann eine zweite XOR-Schaltung aufweisen, die dasselbe zweite Fehlererkennungscodeerzeugungspolynom verwendet. Sowohl der dritte als auch der siebte Fehlererkennungscodegenerator kann eine dritte XOR-Schaltung aufweisen, die dasselbe dritte Fehlererkennungscodeerzeugungspolynom verwendet, und sowohl der vierte als auch der achte Fehlererkennungscodegenerator können eine vierte XOR-Schaltung aufweisen, die dasselbe vierte Fehlererkennungscodeerzeugungspolynom verwendet.

**[0015]** Vorteilhafte Ausgestaltungen der Erfindung, die nachfolgend detailliert beschrieben sind, sowie die zur Erleichterung des Verständnisses der Erfindung diskutierten Ausgestaltungen des Standes der Technik sind in den Zeichnungen dargestellt. Es zeigt:

**Fig. 1** ein Blockschaltbild eines herkömmlichen Halbleiterspeicherelements mit einer ODIC-Padstruktur;

**Fig. 2** ein Blockschaltbild einer beispielhaften Ausgestaltung des Halbleiterspeicherelements mit einer ODIC-Padstruktur;

**Fig. 3** eine beispielhafte Konfiguration eines ersten Fehlererkennungscodegenerators in **Fig. 2** gemäß einer beispielhaften Ausgestaltung;

**Fig. 4** eine beispielhafte Konfiguration einer Fehlererkennungscodeerzeugungsschaltung in **Fig. 3** gemäß einer beispielhaften Ausgestaltung;

**Fig. 5** beispielhafte Strukturen von Parallel-Seriell-Wandlern in **Fig. 2** gemäß einer beispielhaften Ausgestaltung;

**Fig. 6** ein Blockschaltbild eines Halbleiterspeicherelements mit einer ODIC-Padstruktur gemäß einer anderen beispielhaften Ausgestaltung;

**Fig. 7** ein Blockschaltbild eines Halbleiterspeicherelements mit einer ODIC-Padstruktur gemäß einer weiteren beispielhaften Ausgestaltung;

**Fig. 8** ein Blockschaltbild eines Halbleiterspeicherelements mit einer ODIC-Padstruktur gemäß einer weiteren beispielhaften Ausgestaltung;

**Fig. 9** eine beispielhafte Konfiguration eines Speichersystems gemäß einer beispielhaften Ausgestaltung; und

**Fig. 10** eine beispielhafte Konfiguration von Parallel-Seriell-Wandlern in **Fig. 9** gemäß einer beispielhaften Ausgestaltung.

**[0016]** Es sei darauf hingewiesen, dass ein Element, welches als mit einem anderen Element „verbunden“ oder „gekoppelt“ beschrieben ist, direkt mit dem anderen Element verbunden oder gekoppelt sein kann, oder dass Zwischenelemente vorhanden sein können. Wenn dagegen ein Element als mit einem anderen Element „direkt verbunden“ oder „direkt gekoppelt“ beschrieben ist, sind keine Zwischenelemente vorhanden. Andere Wörter, die verwendet werden, um die Beziehung zwischen den Elementen zu beschreiben, sollten in gleicher Weise interpretiert werden (z.B. „zwischen“ gegenüber „direkt zwischen“, „benachbart“ gegenüber „direkt benachbart“ usw.).

**[0017]** **Fig. 2** zeigt ein Halbleiterspeicherelement mit einer ODIC-Padstruktur gemäß einer beispielhaften Ausgestaltung. Wie in **Fig. 2** gezeigt, kann das Halbleiterspeicherelement erste bis vierte Speicherzellenfeldblöcke **BLK1** bis **BLK4**, erste und zweite Fehlererkennungscodegeneratoren **ECCG1** und **ECCG2**, erste und zweite Parallel-Seriell-Wandler **1P2S** und **2P2S** und einen Taktsignalgenerator **CLKG** aufweisen. Der erste Parallel-Seriell-Wandler **1P2S** kann **k** Parallel-Seriell-Wandler **1P2S1** bis **1P2Sk** beinhalten und der zweite Parallel-Seriell-Wandler **2P2S** kann **k** Parallel-Seriell-Wandler **2P2S1** bis **2P2Sk** beinhalten.

**[0018]** Die Pads in **Fig. 2** können ODIC-Padstrukturen aufweisen. In der beispielhaften Ausgestaltung in **Fig. 2** entsprechen **DQ11** bis **DQ1k** in **Fig. 2** einer ersten Gruppe von Datenpads und **DQ21** bis **DQ2k** entsprechen einer zweiten Gruppe von Datenpads. Des Weiteren entsprechen **DO1** bis **DO4** **n**-Bit-Daten, die jeweils von dem ersten bis vierten Speicherzellenfeldblock **BLK1** bis **BLK4** ausgegeben werden.

**[0019]** Das in **Fig. 2** dargestellte Halbleiterspeicherelement kann selektiv eine Gesamtheit von **2n** Datenbits **DO1** und **DO4** von den ersten und vierten Speicherzellenfeldblöcken **BLK1** und **BLK4** ausgeben. Alternativ kann das Halbleiterspeicherelement selektiv eine Gesamtheit von **2n** Datenbits **DO2** und **DO3** von den zweiten und dritten Speicherzellenfeldblöcken **BLK2** und **BLK3** ausgeben.

**[0020]** Beispielhafte Funktionen der Bestandteile des Halbleiterspeicherelements in **Fig. 2** sind nachfolgend beschrieben.

**[0021]** Der erste Fehlererkennungscodegenerator **ECCG1** kann n-Bit-Daten **DO1** oder **DO2** empfangen, die von dem ersten oder zweiten Speicherzellenfeldblock **BLK1** oder **BLK2** ausgegeben werden, kann einen ersten 2m-Bit-Fehlererkennungscode erzeugen und kann obere m Bits eines ersten 2m-Bit-Fehlererkennungscode mit einem zweiten Obere-m-Bit-Fehlererkennungscode kombinieren, der von dem zweiten Fehlererkennungscodegenerator **ECCG2** übertragen wird, um einen endgültigen Obere-m-Bit-Fehlererkennungscode zu erzeugen. Der zweite Fehlererkennungscodegenerator **ECCG2** kann n-Bit-Daten **DO3** oder **DO4** empfangen, die von dem dritten oder vierten Speicherzellenfeldblock **BLK3** oder **BLK4** ausgegeben werden, kann den zweiten 2m-Bit-Fehlererkennungscode erzeugen und kann untere m Bits des zweiten 2m-Bit-Fehlererkennungscode mit einem ersten Untere-n-Bit-Fehlererkennungscode kombinieren, der von dem ersten Fehlererkennungscodegenerator **ECCG1** übertragen wird, um einen endgültigen Untere-m-Bit-Fehlererkennungscode zu erzeugen. Der Taktsignalgenerator **CLKG** kann  $n/k+m/k$  Taktsignale **P1** -  $P(n/k+m/k)$  mit unterschiedlichen Phasen in Abhängigkeit von einem extern angelegten Taktsignal (nicht gezeigt) erzeugen. Alternativ kann der Taktsignalgenerator **CLKG** intern  $n/k+m/k$  Taktsignale **P1**  $\sim P(n/k+m/k)$  mit unterschiedlichen Phasen erzeugen. Die Parallel-Seriell-Wandler **1P2S1** bis **1P2Sk** können  $n/k$  Datenbits und obere  $m/k$  Bits des endgültigen Fehlererkennungscode in Abhängigkeit von den entsprechenden  $n/k+m/k$  Taktsignalen **P1** bis  $P(n/k+m/k)$  in serielle Daten umwandeln und können die seriellen Daten über die entsprechenden Pads **DQ11** bis **DQ1k** ausgeben. Die Parallel-Seriell-Wandler **2P2S1** bis **2P2Sk** können  $n/k$  Datenbits und obere  $m/k$  Bits des endgültigen Fehlererkennungscode in Abhängigkeit von den entsprechenden  $n/k+m/k$  Taktsignalen **P1** bis  $P(n/k+m/k)$  in serielle Daten umwandeln und können die seriellen Daten über die entsprechenden Pads **DQ21** bis **DQ2k** ausgeben.

**[0022]** Das Halbleiterspeicherelement mit der in **Fig. 2** gezeigten ODIC-Padstruktur kann einen ersten Fehlererkennungscodegenerator **ECCG1** zum Erzeugen eines Fehlererkennungscode für Daten, die von dem ersten oder zweiten Speicherzellenfeldblock **BLK1** oder **BLK2** ausgegeben werden, und einen zweiten Fehlererkennungscodegenerator **ECCG2** zum Erzeugen eines Fehlererkennungscode für Daten, die von dem dritten oder vierten Speicherzellenfeldblock **BLK3** oder **BLK4** ausgegeben werden, aufweisen. Entsprechend können Signalleitungen zum Übertragen der Daten **DO1** und **DO2** und Signalleitungen zum Übertragen der Daten **DO3** und **DO4** im Vergleich zu denen, die in her-

kömmlichen Halbleiterelementen verwendet werden, relativ kurz sein.

**[0023]** Das Halbleiterspeicherelement mit der in **Fig. 2** gezeigten ODIC-Padstruktur kann die oberen m Bits eines ersten Fehlererkennungscode von dem ersten Fehlererkennungscodegenerator **ECCG1** zu dem zweiten Fehlererkennungscodegenerator **ECCG2** übertragen und kann die unteren m Bits eines zweiten Fehlererkennungscode von dem zweiten Fehlererkennungscodegenerator **ECCG2** zu dem ersten Fehlererkennungscodegenerator **ECCG1** übertragen. Dementsprechend können 2 m Signalleitungen relativ lang sein. Selbst wenn die 2 m Signalleitungen relativ lang sind, ist die Anzahl der relativ langen Signalleitungen klein im Vergleich zu der Anzahl von Signalleitungen zum Übertragen von Daten. Daher kann das Halbleiterspeicherelement gemäß der beispielhaften Ausgestaltung, die vorstehend beschrieben wurde und welche die ODIC-Padstruktur aufweist, eine verringerte Layoutfläche im Vergleich mit herkömmlichen Halbleiterspeicherelementen aufweisen.

**[0024]** **Fig. 3** zeigt eine beispielhafte Konfiguration des ersten Fehlererkennungscodegenerators in **Fig. 2**. Wie in **Fig. 3** gezeigt, kann der erste Fehlererkennungscodegenerator beispielsweise eine Fehlererkennungscodeerzeugungsschaltung **10** und eine **XOR**-Schaltung **12** aufweisen.

**[0025]** Beispielhafte Funktionen der Fehlererkennungscodeerzeugungsschaltung **10** und der **XOR**-Schaltung **12** werden nachfolgend beschrieben.

**[0026]** Die Fehlererkennungscodeerzeugungsschaltung **10** kann bei Empfang von n-Bit-Daten **D1** erste 2m-Bit-Fehlererkennungscode **1ec1** und **1ec2** für die n-Bit-Daten erzeugen. Das **XOR**-Gatter **12** kann eine **XOR**-Operation an dem ersten Fehlererkennungscode **1ec1** aus den ersten 2m-Bit-Fehlererkennungscode **1ec1** und **1ec2** und an einem zweiten Obere-m-Bit-Fehlererkennungscode **2ec1** durchführen, um einen endgültigen Obere-m-Bit-Fehlererkennungscode **fec1** zu erzeugen.

**[0027]** Obwohl nicht gezeigt, kann ein zweiter Fehlererkennungscodegenerator dieselbe Struktur wie in **Fig. 3** aufweisen.

**[0028]** **Fig. 4** zeigt eine beispielhafte Konfiguration einer Fehlererkennungscodeerzeugungsschaltung in **Fig. 3**, die beispielsweise acht **XOR**-Schaltungen aufweisen kann.

**[0029]** Die beispielhaften Schaltung, die in **Fig. 4** gezeigt ist, kann auf einem Cyclic-Redundancy-Check(**CRC**)-Code basieren, wobei ein Fehlererkennungscodeerzeugungspolynom  $X^8 + X^5 + X^3 + X^2 + X + 1$  ist.

**[0030]** Zusätzlich kann der Fehlererkennungscodegenerator in **Fig. 4** den Rest als Fehlererkennungscode **r1** bis **r8** erzeugen. Der Rest kann den 40-Bit-Daten **d32...d10...0** entsprechen, bei denen es sich um die niedrigsten Bitdaten von parallelen 32-Bit-Ausgabedaten **d32** bis **d1** zuzüglich „00000000“ handelt, die durch dem Polynom entsprechende Daten „10010111“ geteilt werden.

**[0031]** Bezugnehmend auf **Fig. 4** kann eine erste **XOR**-Schaltung eine **XOR**-Operation an Daten **d1**, **d2**, **d5**, **d7** bis **d11**, **d17**, **d20** bis **d25**, **d27**, **d29** und **d32** durchführen, um einen Fehlererkennungscode **r1** zu erzeugen; eine zweite **XOR**-Schaltung kann eine **XOR**-Operation an Daten **d2**, **d4** bis **d6**, **d11**, **d16**, **d17**, **d19**, **d25** bis **d29**, **d31** und **d32** durchführen, um einen Fehlererkennungscode **r2** zu erzeugen; eine dritte **XOR**-Schaltung kann eine **XOR**-Operation an Daten **d2** bis **d4**, **d7** bis **d9**, **d11**, **d16** bis **d18**, **d20** bis **d23**, **d26** und **d28** bis **d32** durchführen, um einen Fehlererkennungscode **r3** zu erzeugen; und eine vierte **XOR**-Schaltung kann eine **XOR**-Operation an Daten **d3**, **d5**, **d6**, **d9**, **d11**, **d14** bis **d16**, **d19**, **d23**, **d24**, **d28** und **d30** bis **d32** durchführen, um einen Fehlererkennungscode **r4** zu erzeugen. Eine fünfte **XOR**-Schaltung kann eine **XOR**-Operation an Daten **d2**, **d4**, **d5**, **d8**, **d10**, **d13** bis **d15**, **d18**, **d22**, **d23**, **d27** und **d29** bis **d31** durchführen, um einen Fehlererkennungscode **r5** zu erzeugen; eine sechste **XOR**-Schaltung kann eine **XOR**-Operation an Daten **d2** bis **d5**, **d8**, **d10** bis **d14**, **d20**, **d23** bis **d28**, **d30** und **d32** durchführen, um einen Fehlererkennungscode **r6** zu erzeugen; eine siebte **XOR**-Schaltung kann eine **XOR**-Operation an Daten **d1** bis **d4**, **d7**, **d9** bis **d13**, **d19**, **d22** bis **d27**, **d29** und **d31** durchführen, um einen Fehlererkennungscode **r7** zu erzeugen; und eine achte **XOR**-Schaltung kann eine **XOR**-Operation an Daten **d1** bis **d3**, **d6**, **d8** bis **d12**, **d18**, **d21** bis **d26**, **d28** und **d30** durchführen, um einen Fehlererkennungscode **r8** zu erzeugen.

**[0032]** Die beispielhafte Fehlererkennungscodeerzeugungsschaltung, die in **Fig. 4** gezeigt ist, kann erste 2m-Bit-Fehlererkennungscode (**c1** bis **c8**; **1ec**) erzeugen und somit kann der erste Fehlererkennungscodegenerator einen endgültigen Obere-m-Bit-Fehlererkennungscode (**fec**) erzeugen.

**[0033]** Das in **Fig. 1** gezeigte herkömmliche Halbleiterspeicherelement weist einen Fehlererkennungscodegenerator auf, um einen endgültigen 8-Bit-Fehlererkennungscode für beispielsweise insgesamt 64-Bit-Daten zu erzeugen, wohingegen die in **Fig. 2** gezeigte beispielhafte Ausgestaltung des Halbleiterspeicherelements zwei Fehlererkennungscodegeneratoren aufweist, von denen jeder eine Fehlererkennungscodeerzeugungsschaltung zum Erzeugen eines 8-Bit-Fehlererkennungscode für 32-Bit-Daten aufweist.

**[0034]** Somit kann jeder der Fehlererkennungscodegeneratoren des Halbleiterspeicherelements gemäß einer beispielhaften Ausgestaltung einen Fehlererkennungscode für Daten mit weniger Bits unter Verwendung eines Fehlererkennungscodeerzeugungspolynoms derselben Ordnung erzeugen, wie es in dem Fehlererkennungscodegenerator des herkömmlichen Halbleiterspeicherelements verwendet wird, so dass die Leistungsfähigkeit der Fehlererkennung verbessert wird.

**[0035]** **Fig. 5** zeigt eine beispielhafte Konfiguration, die für jeden der in **Fig. 2** gezeigten Parallel-Seriell-Wandler verwendet werden kann. Wie in **Fig. 5** gezeigt, kann die beispielhafte Konfiguration beispielsweise **AND**-Gatter **AND1** bis **AND(n/k)**, **AND(n/k+1)** bis **AND(n/k+m/k)** und ein **OR**-Gatter **OR1** aufweisen.

**[0036]** Bei der in **Fig. 5** gezeigten beispielhaften Konfiguration bezeichnen **do11** bis **do(n/k)** n/k-Bit-Daten, **1ec11** bis **1ec1(m/k)** bezeichnen einen m/k-Bit-Fehlererkennungscode, **P1** bis **P(n/k+m/k)** bezeichnen Taktsignale mit jeweils unterschiedlichen Phasen und **DOUT** bezeichnet Ausgangs- oder Ausgabedaten.

**[0037]** Jedes der **AND**-Gatter **AND1** bis **AND(n/k+m/k)** kann sequentiell (n/k+m/k)-Bit-Daten und einen Fehlererkennungscode in Abhängigkeit von den betreffenden (n/k+m/k) Taktsignalen **P1** bis **P(n/k+m/k)** ausgeben. Das **OR**-Gatter **OR1** kann eine **XOR**-Operation an Daten ausführen, die von allen der **AND**-Gatter **AND1** bis **AND(n/k+m/k)** ausgegeben werden, um die Ausgabedaten **DOUT** zu erzeugen. Entsprechend können die (n/k+m/k)-Bit-Daten und die Fehlererkennungscode in serielle Daten umgewandelt und dann ausgegeben werden.

**[0038]** Der in **Fig. 5** gezeigte beispielhafte Parallel-Seriell-Wandler kann n/k-Bit-Daten und anschließend den m/k-Bit-Fehlererkennungscode ausgeben. Alternativ kann der beispielhafte Parallel-Seriell-Wandler zunächst den Fehlererkennungscode und anschließend Daten übertragen. Gemäß einer weiteren Alternative kann der beispielhafte Parallel-Seriell-Wandler den Fehlererkennungscode übertragen während die Daten übertragen werden.

**[0039]** **Fig. 6** ist ein Blockschaltbild eines Halbleiterspeicherelements mit einer **ODIC**-Padstruktur gemäß einer weiteren beispielhaften Ausgestaltung. Das Halbleiterspeicherelement kann beispielsweise erste bis vierte Speicherzellenfeldblöcke **BLK1** bis **BLK4**, erste bis vierte Fehlererkennungscodegeneratoren **ECCG1** bis **ECCG4**, erste und zweite Parallel-Seriell-Wandler **1P2S** und **2P2S** und einen Taktsignalgenerator **CLKG** aufweisen. Die ersten und zweiten Parallel-Seriell-Wandler **1P2S** und **2P2S** können dieselbe Struktur wie diejenigen in **Fig. 2** aufweisen.

**[0040]** Die Pads in **Fig. 6** können mit derselben **ODIC**-Struktur angeordnet sein, wie in **Fig. 2** gezeigt. Entsprechend bezeichnen in der beispielhaften Ausgestaltung gemäß **Fig. 6** **DQ11** bis **DQ1k** eine erste Gruppe von Datenpads und **DQ21** bis **DQ2k** bezeichnen eine zweite Gruppe von Datenpads. Des Weiteren bezeichnen Daten **DO1** bis **DO4** 2n-Bit-Daten, die jeweils von dem ersten bis vierten Speicherzellenfeldblock **BLK1** bis **BLK4** ausgegeben werden.

**[0041]** Das in **Fig. 6** gezeigte Halbleiterspeicherelement kann selektiv eine Gesamtheit von  $4n$  Datenbits ausgeben, die von den ersten und vierten Speicherzellenfeldblöcken **BLK1** und **BLK4** ausgegeben werden, oder eine Gesamtheit von  $4n$  Datenbits, die von den zweiten und dritten Speicherzellenfeldblöcken **BLK2** und **BLK3** ausgegeben werden.

**[0042]** Beispielhafte Funktionen der Bestandteile des Halbleiterspeicherelements in **Fig. 6** sind nachfolgend beschrieben.

**[0043]** Der erste Fehlererkennungscodegenerator **ECCG1** kann einen ersten 2m-Bit-Fehlererkennungscode für die unteren n-Bits von 2n-Bit-Daten **DO1** oder **DO2** erzeugen, die von dem ersten oder zweiten Speicherzellenfeldblock **BLK1** oder **BLK2** ausgegeben werden, und kann die oberen m Bits der 2m Bits eines ersten Fehlererkennungscode mit den oberen m Bits der 2m Bits eines dritten Fehlererkennungscode kombinieren, der von dem dritten Fehlererkennungscodegenerator **ECCG3** ausgegeben wird, um die unteren m Bits der oberen 2m Bits eines endgültigen Fehlererkennungscode zu erzeugen. Der zweite Fehlererkennungscodegenerator **ECCG2** kann einen zweiten 2m-Bit-Fehlererkennungscode für die oberen n Bits von 2n-Bit-Daten **DO1** oder **DO2** erzeugen, die von dem ersten oder zweiten Speicherzellenfeldblock **BLK1** oder **BLK2** ausgegeben werden, und kann die oberen m Bits des zweiten 2m-Bit-Fehlererkennungscode mit den oberen m Bits des 2m-Bit vierten Fehlererkennungscode kombinieren, der von dem vierten Fehlererkennungscodegenerator **ECCG4** ausgegeben wird, um die oberen m Bits des endgültigen Obere-2m-Bit-Fehlererkennungscode zu erzeugen. Der dritte Fehlererkennungscodegenerator **ECCG3** kann 2m Bits des dritten Fehlererkennungscode für die unteren n Bits von 2n-Bit-Daten **DO3** oder **DO4** erzeugen, die von dem dritten oder vierten Speicherzellenfeldblock **BLK3** oder **BLK4** ausgegeben werden, und kann die unteren m Bits der 2m Bits des dritten Fehlererkennungscode mit den unteren m Bits der 2m Bits des ersten Fehlererkennungscode kombinieren, der von dem ersten Fehlererkennungscodegenerator **ECCG1** ausgegeben wird, um die unteren m Bits der unteren 2m Bits eines endgültigen Fehlererkennungscode zu erzeugen. Der vierte Fehlererkennungscodegenerator **ECCG4** kann einen vierten 2m-Bit-Fehlererkennungscode für die oberen n

Bits von 2n-Bit-Daten **DO3** oder **DO4** erzeugen, die von dem dritten oder vierten Speicherzellenfeldblock **BLK3** oder **BLK4** ausgegeben werden, und kann die oberen m Bits der 2m Bits des vierten Fehlererkennungscode mit den oberen m Bits der 2m Bits des zweiten Fehlererkennungscode kombinieren, der von dem zweiten Fehlererkennungscodegenerator **ECCG2** ausgegeben wird, um die oberen m Bits der unteren 2m Bits eines endgültigen Fehlererkennungscode zu erzeugen. Der Taktsignalgenerator **CLKG** kann in Abhängigkeit von einem extern angelegten Taktsignal (nicht gezeigt)  $2n/k+2m/k$  Taktsignale mit unterschiedlichen Phasen erzeugen. Alternativ kann der Taktsignalgenerator intern  $2n/k+2m/k$  Taktsignale mit unterschiedlichen Phasen erzeugen. Jeder der k Parallel-Seriell-Wandler **1P2S1** bis **1P2Sk** kann Daten **DO1** oder **DO2** in Einheiten von  $2n/k$  Bits und einen oberen endgültigen Fehlererkennungscode in Einheiten von  $2m/k$  Bits in Abhängigkeit von den jeweiligen  $2n/k+2m/k$  Taktsignalen mit unterschiedlichen Phasen in serielle Daten umwandeln und kann die umgewandelten Daten durch die entsprechenden Pads **DQ11** bis **DQ1k** ausgeben. Jeder der k Parallel-Seriell-Wandler **2P2S1** bis **2P2Sk** kann Daten **DO3** oder **DO4** in Einheiten von  $2n/k$  Bits und einen unteren Fehlererkennungscode in Einheiten von  $2m/k$  Bits in Abhängigkeit von den jeweiligen Taktsignalen mit unterschiedlichen Phasen in serielle Daten umwandeln und kann die umgewandelten Daten durch die entsprechenden Pads **DQ21** bis **DQ2k** ausgeben.

**[0044]** Gemäß der in **Fig. 6** gezeigten beispielhaften Ausgestaltung kann das Halbleiterspeicherelement erste und zweite Fehlererkennungscodegeneratoren **ECCG1** und **ECCG2** aufweisen, die Fehlererkennungscode für Daten erzeugen, die von den ersten oder zweiten Speicherzellenfeldblöcken **BLK1** oder **BLK2** ausgegeben werden, und kann dritte und vierte Fehlererkennungscodegeneratoren **ECCG3** und **ECCG4** aufweisen, die Fehlererkennungscode für Daten erzeugen können, die von den dritten oder vierten Speicherzellenfeldblöcke **BLK3** oder **BLK4** ausgegeben werden. Bei dieser beispielhaften Ausgestaltung sind relativ lange Signalleitungen zum Übertragen der Daten **DO1** oder **DO2** und der Daten **DO3** oder **DO4** nicht erforderlich. Entsprechend sind nur 4 m Signalleitungen zum Übertragen der m Bits des Fehlererkennungscode, der von jedem der Fehlererkennungscodegeneratoren ausgegeben wird, relativ lang. Die Anzahl von 4 m Signalleitungen ist im Vergleich zu der Anzahl von Signalleitungen zum Übertragen der Daten immer noch relativ gering, so dass eine Layoutfläche des Halbleiterspeicherelements, welches die **ODIC**-Padstruktur gemäß dieser beispielhaften Ausgestaltung aufweist, im Vergleich zu herkömmlichen Halbleiterspeicherelementen reduziert sein kann.

[0045] Der Fehlererkennungscodegenerator in **Fig. 6** kann dieselbe Struktur wie der Fehlererkennungscode in **Fig. 3** aufweisen und die Fehlererkennungscodeerzeugungsschaltung kann dieselbe Struktur wie die Fehlererkennungscodeerzeugungsschaltung in **Fig. 4** aufweisen. Obwohl nicht dargestellt, können die Parallel-Seriell-Wandler dieselbe Struktur wie die Parallel-Seriell-Wandler in **Fig. 5** aufweisen.

[0046] Wenn eine Gesamtheit von 64 Datenbits **DO1** oder **DO2** von dem ersten oder zweiten Speicherzellenfeldblock **BLK1** oder **BLK2** ausgegeben wird, kann die Fehlererkennungscodeerzeugungsschaltung in **Fig. 4** bewirken, dass der erste bis vierte Fehlererkennungscodeerzeuger **ECCG1** bis **ECCG4** Fehlererkennungscode in Einheiten von 8 Bits für Daten aus 32 Bits erzeugen und endgültige Fehlererkennungscode können in Einheiten von 4 Bits durch das **XOR**-Gatter von **Fig. 3** erzeugt werden.

[0047] Die beispielhafte Ausgestaltung des Halbleiterspeicherelements in **Fig. 6** kann den Fehlererkennungscode erzeugen ohne die Leistungsfähigkeit der Fehlererkennung zu beeinträchtigen.

[0048] **Fig. 7** ist ein Blockschaltbild eines Halbleiterspeicherelements mit einer **ODIC**-Padstruktur gemäß einer weiteren beispielhaften Ausgestaltung. Wie in **Fig. 7** gezeigt, kann die beispielhaften Ausgestaltung des Halbleiterspeicherelements erste bis vierte Speicherzellenblöcke **BLK1** bis **BLK4**, erste bis achte Auswähler **SEL1** bis **SEL8**, erste bis vierte Fehlererkennungscodegeneratoren **ECCG1** bis **ECCG4**, erste und zweite Parallel-Seriell-Wandler **1P2S** und **2P2S**, einen Taktsignalgenerator **CLKG** und einen Steuersignalgenerator **CONP** aufweisen. Jeder der ersten bis vierten Speicherzellenfeldblöcke **BLK1** bis **BLK4** kann erste und zweite Unterspeicherzellenfeldblöcke (**BLK11**, **BLK12**) bis (**BLK41**, **BLK42**) aufweisen. Der erste Parallel-Seriell-Wandler **1P2S** kann  $k$  Parallel-Seriell-Wandler **1P2S1** bis **1P2Sk** aufweisen und der zweiten Parallel-Seriell-Wandler **2P2S** kann  $k$  Parallel-Seriell-Wandler **2P2S1** bis **2P2Sk** aufweisen.

[0049] Die Pads in **Fig. 7** weisen eine **ODIC**-Padstruktur auf. In **Fig. 7** können **DQ11** bis **DQ1k** eine erste Gruppe von Datenpads bezeichnen, und **DQ21** bis **DQ2k** können eine zweite Gruppe von Datenpads bezeichnen.  $2n$  Datenbits (**DO11**, **DO12**), (**DO21**, **DO22**), (**DO31**, **DO32**), und (**DO41**, **DO42**) können erste und zweite  $n$ -Bit-Daten bezeichnen, die jeweils von dem ersten bis vierten Speicherzellenfeldblock **BLK1** bis **BLK4** ausgegeben werden.

[0050] Das Halbleiterspeicherelement in **Fig. 7** kann in einem ersten Bitstrukturmodus oder in einem zweiten Bitstrukturmodus arbeiten. In einem ersten Bitstrukturmodus kann das Halbleiterspeicherelement

selektiv eine Gesamtheit von  $4n$  Datenbits **DO11**, **DO12**, **DO41** und **DO42** von dem ersten und vierten Speicherzellenfeldblock **BLK1** und **BLK4** oder eine Gesamtheit von  $4n$  Datenbits **DO21**, **DO22**, **DO31** und **DO32** aus dem zweiten und dritten Speicherzellenfeldblock **BLK2** und **BLK3** ausgeben. In einem zweiten Bitstrukturmodus kann das Halbleiterspeicherelement selektiv  $2n$  Datenbits aus den Unterspeicherzellenfeldblöcken ausgeben, die beliebige der ersten und zweiten Unterspeicherzellenfeldblöcke **BLK11** und **BLK12** des ersten Speicherzellenfeldblocks **BLK1** und beliebige der ersten und zweiten Unterspeicherzellenfeldblöcke **BLK41** und **BLK42** des vierten Speicherzellenfeldblocks **BLK4** einschließen, Alternativ kann das Halbleiterspeicherelement in einem zweiten Bitstrukturmodus selektiv zwei  $2n$  Datenbits aus den Unterspeicherzellenfeldblöcken ausgeben, die beliebige der ersten und zweiten Unterspeicherzellenfeldblöcke **BLK21** und **BLK22** des zweiten Speicherzellenfeldblocks **BLK2** und beliebige der ersten und zweiten Unterspeicherzellenfeldblöcke **BLK31** und **BLK32** des dritten Speicherzellenfeldblocks **BLK3** einschließen.

[0051] Beispielhafte Funktionen der Bestandteile der Blöcke des Halbleiterspeicherblocks in **Fig. 7** sind nachfolgend beschrieben.

[0052] Wenn ein Befehlssignal **COM**, das von den Adress- und Befehlssignalanlegepads **CMD/ADD** angelegt wird, ein Moduseinstellbefehl ist, decodiert der Steuersignalgenerator **CONP** den angelegten Moduseinstellcode, um ein Modussignal einzustellen, das den ersten Bitstrukturmodus oder den zweiten Bitstrukturmodus bezeichnet. Der Steuersignalgenerator **CONP** kann sowohl erste als auch zweite Steuersignale **con1** und **con2** aktivieren, wenn das Befehlssignal **COM** ein Schreib- oder Lesebefehl ist und wenn das Modussignal den ersten Bitstrukturmodus bezeichnet. Der Steuersignalgenerator **CONP** kann in Abhängigkeit von einer Spalten- (oder Zeilen-)Adresse aus einem Bit das zweite Steuersignal **con2** deaktivieren und das erste Steuersignal **con1** aktivieren oder deaktivieren, wenn das Modussignal den zweiten Bitstrukturmodus bezeichnet. Die Spalten- oder (Zeilen-)Adresse aus einem Bit kann dazu verwendet werden, erste und zweite Unterspeicherzellenfeldblöcke von jedem der Speicherzellenfeldblöcke **BLK1** bis **BLK4** zu unterteilen. Die ersten, vierten, fünften und achten Auswähler **SEL1**, **SEL4**, **SEL5** und **SEL8** können erste Daten **DO11**, **DO31**, **DO21** bzw. **DO41** selektieren und ausgeben, wenn das erste Steuersignal **con1** aktiviert ist, und können zweite Daten **DO12**, **DO32**, **DO22** bzw. **DO42** selektieren und ausgeben, wenn das erste Steuersignal **con1** deaktiviert ist. Die zweiten, dritten, sechsten und sieben Auswähler **SEL2**, **SEL3**, **SEL6** und **SEL7** können zweite Daten **DO12**, **DO32**, **DO22** bzw. **DO42** selektieren und ausgeben, wenn das zweite Steuersignal **con2** aktiviert ist, und können die Ausgabe

der Daten blockieren, wenn das zweite Steuersignal **con2** deaktiviert ist. Der erste bis vierte Fehlererkennungscodegenerator **ECCG1** bis **ECCG4** können im Wesentlichen dieselben Funktionen wie der erste bis vierte Fehlererkennungscodegenerator ausüben, die weiter oben unter Bezugnahme auf **Fig. 6** beschrieben wurden. In gleicher Weise können die ersten und zweiten Parallel-Seriell-Wandler **1P2S** und **2P2S** im Wesentlichen dieselben Funktionen ausüben wie die ersten und zweiten Parallel-Seriell-Wandler, die weiter oben unter Bezugnahme auf **Fig. 6** beschrieben wurden. Der Taktsignalgenerator **CLKG** kann ebenfalls im Wesentlichen dieselbe Funktion wie der Taktsignalgenerator ausüben, der weiter oben unter Bezugnahme auf **Fig. 6** beschrieben wurde. Gemäß der beispielhaften Ausgestaltung in **Fig. 7** können Daten durch die ersten und zweiten Gruppen von Datenpads **DQ11** bis **DQ1k** und **DQ21** bis **DQ2k** in dem ersten Bitstrukturmodus ausgegeben werden und können durch die ersten und zweiten Gruppen von Datenpads **DQ11** bis **DQ1(k/2)** und **DQ21** bis **DQ2(k/2)** in dem zweiten Bitstrukturmodus ausgegeben werden.

**[0053]** Die Layoutfläche des Halbleiterspeicherelements, welches die ODIC-Padstruktur gemäß **Fig. 7** aufweist, kann darüber hinaus im Vergleich mit herkömmlichen Halbleiterspeicherelementen reduziert sein.

**[0054]** Der Fehlererkennungscodegenerator in **Fig. 7** kann im Wesentlichen dieselbe Struktur wie der Fehlererkennungscodegenerator in **Fig. 3** aufweisen und die Fehlererkennungscodeerzeugungsschaltung kann im Wesentlichen dieselbe Struktur wie die Fehlererkennungscodeerzeugungsschaltung in **Fig. 4** aufweisen. Obwohl nicht dargestellt, können die Parallel-Seriell-Wandler auch im Wesentlichen dieselbe Struktur wie die Parallel-Seriell-Wandler in **Fig. 5** aufweisen.

**[0055]** Das Halbleiterspeicherelement in **Fig. 7** kann einen Fehlererkennungscode erzeugen und kann die Leistungsfähigkeit der Fehlererkennung gemäß einer beispielhaften Ausgestaltung erhalten.

**[0056]** Der erste und der vierte Fehlererkennungscodegenerator in **Fig. 6** und **Fig. 7** können derart ausgestaltet sein, dass sie dasselbe Fehlererkennungscodeerzeugungspolynom wie der zweite und der dritte Fehlererkennungscodegenerator verwenden. Alternativ können der erste und der vierte Fehlererkennungscodegenerator in **Fig. 6** und **Fig. 7** so ausgestaltet sein, dass sie Fehlererkennungscodeerzeugungspolynome verwenden, die sich von denen des zweiten und des dritten Fehlererkennungscodegenerators unterscheiden. Wenn unterschiedliche Fehlererkennungscodeerzeugungspolynome verwendet werden, ist die Struktur des ersten und des vierten Fehlererkennungscodegenerators auch unterschied-

lich von der Struktur des zweiten und des dritten Fehlererkennungscodegenerators.

**[0057]** **Fig. 8** ist ein Blockschaltbild eines Halbleiterspeicherelements, das eine ODIC-Padstruktur aufweist, gemäß einer weiteren beispielhaften Ausgestaltung. Das Halbleiterspeicherelement kann im Wesentlichen dieselbe Struktur wie das Halbleiterspeicherelement in **Fig. 7** aufweisen, mit der Ausnahme, dass der erste bis vierte Fehlererkennungscodegenerator **ECCG1** bis **ECCG4** durch zwei erste bis vierte Fehlererkennungscodegeneratoren **ECCG11**, **ECCG12** bis **ECCG41**, **ECCG42** und der Steuersignalgenerator **CONP** durch einen Steuersignalgenerator **CONP'** ersetzt sein können.

**[0058]** Eine Beschreibung von Funktionen der Bestandteile des Halbleiterspeicherelements in **Fig. 8**, welche im Wesentlichen den Bestandteilen des Halbleiterspeicherelements in **Fig. 7** entsprechen, werden zur Verkürzung ausgelassen, so dass nachfolgend nur eine Beschreibung beispielhafter Funktionen der ersetzten Bestandteile gemäß **Fig. 8** angegeben ist.

**[0059]** Die beiden ersten bis vierten Fehlererkennungscodegeneratoren **ECCG11**, **ECCG12** bis **ECCG41**, **ECCG42** können derart ausgestaltet sein, dass sie unterschiedliche Fehlererkennungscodeerzeugungspolynome verwenden. Beispielsweise können die Fehlererkennungscodegeneratoren **ECCG11**, **ECCG21**, **ECCG31** und **ECCG41** derart ausgestaltet sein, dass sie dasselbe Fehlererkennungscodeerzeugungspolynom verwenden, und die Fehlererkennungscodegeneratoren **ECCG12**, **ECCG22**, **ECCG32** und **ECCG42** können derart ausgestaltet sein, dass sie andere Fehlererkennungscodeerzeugungspolynome als die Fehlererkennungscodegeneratoren **ECCG11**, **ECCG21**, **ECCG31** und **ECCG41** verwenden. Alternativ können die beiden Fehlererkennungscodegeneratoren **ECCG11** und **ECCG41** derart ausgebildet sein, dass sie dasselbe Fehlererkennungscodeerzeugungspolynom verwenden, die Fehlererkennungscodegeneratoren **ECCG21** und **ECCG31** können derart ausgebildet sein, dass sie andere Fehlererkennungscodeerzeugungspolynome als die Fehlererkennungscodegeneratoren **ECCG11** und **ECCG41** verwenden, die Fehlererkennungscodegeneratoren **ECCG12** und **ECCG42** können derart ausgebildet sein, dass andere Fehlererkennungscodeerzeugungspolynome als die Fehlererkennungscodegeneratoren **ECCG11**, **ECCG41**, **ECCG21** und **ECCG31** verwenden, und die Fehlererkennungscodegeneratoren **ECCG22** und **ECCG32** können derart ausgebildet sein, dass sie andere Fehlererkennungscodeerzeugungspolynome als die Fehlererkennungscodegeneratoren **ECCG11**, **ECCG41**, **ECCG21**, **ECCG31**, **ECCG12** und **ECCG42** verwenden. Der Steuersignalgenerator **CONP'** kann zusätzlich zu den beispielhaften Funktionen des Steuersignalgenerators

**CONP**, der weiter oben unter Bezugnahme auf **Fig. 7** beschrieben wurde, eine gewünschte und/oder vorbestimmte Zeit in Abhängigkeit von einem Taktsignal zählen, das von dem Taktsignalgenerator **CLKG** erzeugt wird, und kann periodisch ein drittes Steuersignal **con3** basierend auf dem Zählwert aktivieren. Die Fehlererkennungscodegeneratoren **ECCG11**, **ECCG21**, **ECCG31** und **ECCG41** können im Wesentlichen dieselbe Operation wie die Fehlererkennungscodegeneratoren ausüben, die weiter oben unter Bezugnahme auf **Fig. 7** beschrieben wurden, wenn das dritte Steuersignal **con3** aktiviert ist. Die Fehlererkennungscodegeneratoren **ECCG12**, **ECCG22**, **ECCG32** und **ECCG42** können im Wesentlichen dieselbe Operation wie die Fehlererkennungscodegeneratoren ausüben, die weiter oben unter Bezugnahme auf **Fig. 7** beschrieben wurden, wenn das dritte Steuersignal **con3** deaktiviert ist.

**[0060]** Das Halbleiterspeicherelement in **Fig. 8** kann einen endgültigen Fehlererkennungscode erzeugen, indem die Fehlererkennungscodegeneratoren verwendet werden, die periodisch unterschiedliche Strukturen gemäß einer beispielhaften Ausgestaltung aufweisen.

**[0061]** Das Halbleiterspeicherelement in **Fig. 8** kann mehr und längere Signalleitungen als die Halbleiterspeicherelemente aufweisen, die weiter oben unter Bezugnahme auf die **Fig. 2**, **Fig. 6** und **Fig. 7** beschrieben wurden. Jedoch weist das Halbleiterspeicherelement in **Fig. 8** im Vergleich mit der Anzahl von Signalleitungen, die im Allgemeinen in herkömmlichen Halbleiterspeicherelementen vorhanden sind, eine geringere Anzahl an Signalleitungen auf. Somit kann die Layoutfläche eines Halbleiterspeicherelements gemäß der beispielhaften Ausgestaltung in **Fig. 8** gegenüber herkömmlichen Halbleiterspeicherelementen reduziert sein.

**[0062]** Darüber hinaus kann das Halbleiterspeicherelement in **Fig. 8** unter Verwendung von Fehlererkennungscodegeneratoren mit periodisch unterschiedlichen Fehlererkennungscodeerzeugungspolynomen betrieben werden, so dass die Leistungsfähigkeit der Fehlererkennung gegenüber herkömmlichen Halbleiterspeicherelementen erhöht werden kann.

**[0063]** Die vorstehend beschriebenen beispielhaften Ausgestaltungen wurden anhand des Halbleiterspeicherelements mit einer **ODIC**-Padstruktur beschrieben. Jedoch erkennt der Fachmann, dass die Lehren der beispielhaften Ausgestaltungen auch auf Halbleiterspeicherelemente angewendet werden können, die eine andere Padstruktur als die **ODIC**-Padstruktur aufweisen.

**[0064]** Gemäß einer beispielhaften Ausgestaltung kann das dritte Steuersignal **con3** in Abhängigkeit von einem Auffrischbefehl (oder -zyklus) er-

zeugt werden, wenn das Halbleiterspeicherelement in **Fig. 8** ein Dynamic Random Access Memory (**DRAM**) ist. Beispielsweise kann ein Verfahren gemäß einer beispielhaften Ausgestaltung ein Aktivieren eines dritten Steuersignals **con3** in Abhängigkeit von einem ersten Auffrischbefehl (oder -zyklus) und ein Deaktivieren des dritten Steuersignals **con3** in Abhängigkeit von einem zweiten Auffrischbefehl (oder -zyklus) beinhalten. Das bedeutet, dass das dritte Steuersignal **con3** nur in Abhängigkeit von dem Auffrischbefehl (oder -zyklus) ausgelöst werden kann.

**[0065]** **Fig. 9** zeigt eine beispielhafte Konfiguration eines Speichersystems gemäß einer beispielhaften Ausgestaltung. Das Speichersystem kann ein Halbleiterspeicherelement **100** und eine Speichersteuereinheit **200** aufweisen. Die Speichersteuereinheit **200** kann einen ersten und einen zweiten Seriell-Parallel-Wandler **1S2P** und **2S2P**, einen Taktsignalgenerator **CLKG**, einen ersten bis vierten Fehlererkennungscodegenerator **ECCG1** bis **ECCG4**, einen Fehlerdetektor **ED** und einen Dateneingabeteil **20** aufweisen. Der Fehlerdetektor **ED** kann **XOR**-Gatter **XOR1** bis **XOR4** und ein **OR**-Gatter **OR** aufweisen, wie in **Fig. 9** gezeigt. Der erste Seriell-Parallel-Wandler **1S2P** kann **k** Seriell-Parallel-Wandler **1S2P1** bis **1S2Pk** umfassen, und der zweite Seriell-Parallel-Wandler **2S2P** kann **k** Seriell-Parallel-Wandler **2S2P1** bis **2S2Pk** umfassen.

**[0066]** Es sei für die nachfolgende Beschreibung angenommen, dass das Halbleiterspeicherelement **100** des Speichersystems in **Fig. 9** dieselbe Struktur wie die beispielhafte Ausgestaltung des Halbleiterspeicherelements in **Fig. 7** aufweist.

**[0067]** Die Fehlererkennungscodegeneratoren **ECCG1** bis **ECCG4** der Speichersteuereinheit **200** in **Fig. 9** können eine Struktur aufweisen, die an den **CRC**-Code angepasst ist und die dasselbe Fehlererkennungscodeerzeugungspolynom verwendet, wie zuvor unter Bezugnahme auf den ersten bis vierten Fehlererkennungscodegenerator **ECCG1** bis **ECCG4** in **Fig. 6** beschrieben. Demgemäß können die Fehlererkennungscodegeneratoren **ECCG1** bis **ECCG4** in **Fig. 9** im Wesentlichen dieselbe Struktur wie die Fehlererkennungscodegeneratoren **ECCG1** bis **ECCG4** in **Fig. 6** aufweisen.

**[0068]** Bezugnehmend auf **Fig. 9** können Daten **dout11** bis **dout1k** Daten bezeichnen, die von den Datenpads **DQ11** bis **DQ1k** jeweils ausgegeben werden, und Daten **dout21** bis **dout2k** können Daten bezeichnen, die jeweils von den Datenpads **DQ21** bis **DQ2k** ausgegeben werden.

**[0069]** Beispielhafte Funktionen der Blöcke in **Fig. 9** sind nachfolgend beschrieben.

**[0070]** Das Halbleiterspeicherelement **100** kann im Wesentlichen dieselben Funktionen durchführen, wie weiter oben unter Bezugnahme auf das Halbleiterspeicherelement in **Fig. 6** beschrieben. Beispielsweise kann der Taktsignalgenerator **CLKG** dieselbe Funktion wie der Taktsignalgenerator **CLKG** in **Fig. 6** ausüben und somit  $n/k+m/k$  Taktsignale **P1** bis **P(n/k+m/k)** mit unterschiedlichen Phasen erzeugen. Die Seriell-Parallel-Wandler **1S2P1** bis **1S2Pk** können obere Daten in Einheiten von  $n/k$  Bits und einen oberen endgültigen Fehlererkennungscode in Einheiten von  $m/k$  Bits empfangen, die seriell angelegt werden, und können die Daten und den oberen endgültigen Fehlererkennungscode in Abhängigkeit von den entsprechenden  $n/k+m/k$  Taktsignalen **P1** bis **P(n/k+m/k)** in parallele Daten umwandeln, um so obere  $2n$  Datenbits und obere endgültige Fehlererkennungscode mit  $2m$ -Bits zu erzeugen. Die Seriell-Parallel-Wandler **2S2P1** bis **2S2Pk** können untere Daten in Einheiten von  $n/k$  Bits und einen unteren endgültigen Fehlererkennungscode in Einheiten  $m/k$  Bits empfangen, die seriell angelegt werden, und können die unteren Daten und den unteren endgültigen Fehlererkennungscode in Abhängigkeit von den jeweiligen  $n/k+m/k$ -Taktsignalen **P1** bis **P(n/k+m/k)** in parallele Daten umwandeln, um so untere  $2n$  Datenbits und untere endgültige Fehlererkennungscode mit  $2m$  Bits zu erzeugen. Der erste bis vierte Fehlererkennungscodegenerator **ECCG1** bis **ECCG4** können den endgültigen Fehlererkennungscode in Einheiten von  $m$  Bits erzeugen. Die **XOR**-Gatter **XOR1** bis **XOR4** können eine **XOR**-Operation an dem endgültigen Fehlererkennungscode in Einheiten von  $m$  Bits, die von dem ersten bis vierten Fehlererkennungscodegenerator **ECCG1** bis **ECCG4** jeweils ausgegeben werden, und an dem endgültigen Fehlererkennungscode in Einheiten von  $m$  Bits, die von dem ersten und zweiten Seriell-Parallel-Wandler ausgegeben werden, ausführen, um Fehlererkennungscode **e1** bis **e4** zu erzeugen. Dies bedeutet, dass jedes der **XOR**-Gatter **XOR1** bis **XOR4** „0“-Daten ausgeben kann, wenn die  $m$  Bits des endgültigen Fehlererkennungscode, der von dem Fehlererkennungscodegenerator ausgegeben wird, gleich den entsprechenden  $m$  Bits des endgültigen Fehlererkennungscode sind, der von dem Seriell-Parallel-Wandler ausgegeben wird, und dass sie anderenfalls „1“-Daten ausgeben können. Das **OR**-Gatter **OR** kann eine **OR**-Operation an den Fehlererkennungscode **e1** bis **e4** ausführen, um ein Fehlererkennungscode **ER** mit dem Wert „1“ zu erzeugen, wenn wenigstens eines der Signale „1“ anzeigt. Der Dateneingabeteil **20** kann  $4n$  Datenbits übertragen, die von dem ersten und zweiten Seriell-Parallel-Wandler **1S2P** und **2S2P** ausgegeben werden, wenn das Fehlererkennungscode **ER** mit dem Wert „0“ auftritt, und kann die Übertragung der  $4n$  Datenbits, die von dem ersten und zweiten Seriell-Parallel-Wandler **1S2P** und **2S2P** ausgegeben werden, blockieren, wenn das Fehlererkennungscode **ER** mit dem Wert „1“ auftritt.

**[0071]** Obwohl nicht dargestellt, kann das Fehlererkennungscode **ER** zu dem Halbleiterspeicherelement **100** übertragen werden. Darüber hinaus kann das Halbleiterspeicherelement **100** die Übertragung der Ausgabedaten blockieren, wenn das Halbleiterspeicherelement **100** ein Fehlererkennungscode **ER** mit dem Wert „1“ empfängt.

**[0072]** **Fig. 10** zeigt eine beispielhafte Konfiguration eines jeweiligen Seriell-Parallel-Wandlers in **Fig. 9** gemäß einer beispielhaften Ausgestaltung. Die beispielhafte Ausgestaltung eines jeweiligen Seriell-Parallel-Wandlers in **Fig. 9** umfasst **D**-Flip-Flops **DF11** bis **DF1(n/k+n/k-1)** und **DF21** bis **DF2(n/k+m/k)**.

**[0073]** In der beispielhaften Ausgestaltung gemäß **Fig. 10** bezeichnet **dout1**  $n/k$  Bits serieller Eingabedaten und  $m/k$  Bits eines seriellen Eingabefehlererkennungscode, **D111** bis **D111(n/k)** bezeichnen  $n/k$ -Bits paralleler Ausgabedaten, **1ec11** bis **1ec1(m/k)** bezeichnen  $m/k$ -Bits parallelen Ausgabefehlererkennungscode und **P1** bis **P(n/k+m/k)** bezeichnen Taktsignale mit jeweils unterschiedlichen Phasen.

**[0074]** Die **D**-Flip-Flops **DF11** bis **DF1(n/k+m/k-1)** können  $n/k$ -Datenbits und erste bis  $(m/k-1)$ -te Fehlererkennungscode speichern und ausgeben, die jeweils seriell in Abhängigkeit von  $(n/k+m/k-1)$  Taktsignalen eingegeben werden können. Die **D**-Flip-Flops **DF21** bis **DF2(n/k+m/k)** können Daten, die von den jeweiligen **D**-Flip-Flops **DF11** bis **DF1(n/k+m/k-1)** ausgegeben werden, und einen  $(m/k)$ -ten Eingabefehlererkennungscode in Abhängigkeit von den jeweiligen  $(n/k+m/k)$  Taktsignalen speichern und können im Wesentlichen gleichzeitig  $n/k$  Bits paralleler Ausgabedaten und  $m/k$  Bits des parallelen Ausgabefehlererkennungscode ausgeben. Entsprechend können  $(n/k+m/k)$  Datenbits und die Fehlererkennungscode in parallele Daten umgewandelt und dann ausgegeben werden.

**[0075]** Zusätzlich kann das Halbleiterspeicherelement des Speichersystems in der vorstehend beschriebenen beispielhaften Ausgestaltung Fehlererkennungscodegeneratoren und Parallel-Seriell-Wandler aufweisen, um Fehlererkennungscode für Ausgabedaten zu erzeugen und zu übertragen. Jedoch kann das Halbleiterspeicherelement auch Seriell-Parallel-Wandler, die Fehlererkennungscodegeneratoren und den Fehlerdetektor aufweisen, die in der Speichersteuereinheit angeordnet sind, um ein Fehlererkennungscode für Eingabedaten zu erzeugen. Die Speichersteuereinheit kann ebenfalls die Fehlererkennungscodegeneratoren und die Parallel-Seriell-Wandler aufweisen, die in dem Halbleiterspeicherelement angeordnet sein können, um einen Fehlererkennungscode für Ausgabedaten zu erzeugen und zu übertragen.

**[0076]** Die Speichersteuereinheit des Speichersystems gemäß den vorstehend beschriebenen beispielhaften Ausgestaltungen kann derart konfiguriert sein, dass sie eine Struktur aufweist, deren Struktur derjenigen in **Fig. 6** entspricht. Die Speichersteuereinheit kann auch derart ausgestaltet sein, dass sie eine Struktur aufweist, die den beispielhaften Ausgestaltungen der Halbleiterspeicherelemente in **Fig. 2, Fig. 7** oder 8 entspricht.

**[0077]** Das Halbleiterspeicherelement der vorstehend beschriebenen beispielhaften Ausgestaltung kann auf ein Halbleiterspeicherelement angewendet werden, das keine ODIC-Padstruktur aufweist.

**[0078]** Das Halbleiterspeicherelement der vorstehend beschriebenen beispielhaften Ausgestaltungen kann Parallel-Seriell-Wandler zum Ausgeben von Daten aufweisen, es braucht jedoch keine Parallel-Seriell-Wandler aufzuweisen, um parallele Daten auszugeben. In derselben Weise kann die Speichersteuereinheit Seriell-Parallel-Wandler zum Eingeben von Daten aufweisen, es braucht jedoch keine Seriell-Parallel-Wandler aufzuweisen, um parallele Daten auszugeben.

**[0079]** Ein Datenübertragungsverfahren oder ein Daten-Sende-/Empfangsverfahren gemäß einer beispielhaften Ausgestaltung kann zwischen einem Halbleiterspeicherelement und einer Speichersteuereinheit sowie auf andere Daten-Sende-/Empfangssysteme angewendet werden.

**[0080]** Eine Layoutfläche eines Halbleiterspeicherelements gemäß der beispielhaften Ausgestaltungen, die oben beschrieben wurden, kann im Vergleich zu herkömmlichen Halbleiterspeicherelementen, die eine **ODIC**-Padstruktur aufweisen, reduziert sein, so dass Halbleiterspeicherelemente gemäß beispielhaften Ausgestaltungen Fehlererkennungscode erzeugen können, ohne die Leistungsfähigkeit der Fehlererkennung zu beeinträchtigen.

**[0081]** Zusätzlich kann ein Speichersystem gemäß einer beispielhaften Ausgestaltung separate Fehlererkennungscodegeneratoren zum Erzeugen von Fehlererkennungscode verwenden, ohne die Leistungsfähigkeit der Fehlererkennung zu beeinträchtigen.

### Patentansprüche

1. Halbleiterspeicherelement, aufweisend:  
einen ersten Speicherzellenfeldblock (BLK1), der erste Daten (DO1) erzeugt;  
einen zweiten Speicherzellenfeldblock (BLK3), der zweite Daten (DO3) erzeugt;  
einen ersten Fehlererkennungscodegenerator (ECCG1), der einen ersten Fehlererkennungscode (1ec1) für die ersten Daten (DO1) erzeugt und der

einen Anteil von Bits des ersten Fehlererkennungscode (1ec1) mit einem Anteil von Bits eines zweiten Fehlererkennungscode (2ec1) kombiniert, um ein erstes endgültiges Fehlererkennungscode (fec1) zu erzeugen; und

einen zweiten Fehlererkennungscodegenerator (ECCG2), der den zweiten Fehlererkennungscode (2ec1) für die zweiten Daten (DO3) erzeugt und der verbleibende Bits des zweiten Fehlererkennungscode (2ec1), die sich von dem Anteil von Bits des zweiten Fehlererkennungscode (2ec1) unterscheiden, mit verbleibenden Bits des ersten Fehlererkennungscode (1ec1), die sich von dem Anteil von Bits des ersten Fehlererkennungscode (1ec1) unterscheiden, kombiniert, um ein zweites endgültiges Fehlererkennungscode zu erzeugen.

2. Halbleiterspeicherelement nach Anspruch 1, **dadurch gekennzeichnet**, dass der erste Fehlererkennungscodegenerator aufweist:

eine erste Fehlererkennungscodeerzeugungsschaltung (10), die den Fehlererkennungscode für die ersten Daten erzeugt; und

eine erste Kombinierschaltung (12), die den Anteil von Bits des ersten Fehlererkennungscode mit dem Anteil von Bits des zweiten Fehlererkennungscode kombiniert, um das erste endgültige Fehlererkennungscode zu erzeugen.

3. Halbleiterspeicherelement nach Anspruch 2, **dadurch gekennzeichnet**, dass die erste Kombinierschaltung ein XOR-Gatter (12) ist, das eine XOR-Operation an oberen Bits des ersten Fehlererkennungscode und oberen Bits des zweiten Fehlererkennungscode ausführt, um das erste endgültige Fehlererkennungscode zu erzeugen.

4. Halbleiterspeicherelement nach Anspruch 2 oder 3, **dadurch gekennzeichnet**, dass der zweite Fehlererkennungscodegenerator aufweist:

eine zweite Fehlererkennungscodeerzeugungsschaltung (10), die den zweiten Fehlererkennungscode für die zweiten Daten erzeugt; und

eine zweite Kombinierschaltung (12), die die verbleibenden Bits des ersten Fehlererkennungscode, die sich von dem Anteil von Bits des ersten Fehlererkennungscode unterscheiden, mit den verbleibenden Bits des zweiten Fehlererkennungscode, die sich von dem Anteil von Bits des zweiten Fehlererkennungscode unterscheiden, kombiniert, um das zweite endgültige Fehlererkennungscode zu erzeugen.

5. Halbleiterspeicherelement nach Anspruch 4, **dadurch gekennzeichnet**, dass die zweite Kombinierschaltung ein XOR-Gatter (12) ist, das eine XOR-Operation an unteren Bits des zweiten Fehlererkennungscode und unteren Bits des ersten Fehlererkennungscode ausführt, um das zweite endgültige Fehlererkennungscode zu erzeugen.

6. Halbleiterspeicherelement nach Anspruch 4 oder 5, **dadurch gekennzeichnet**, dass die erste und die zweite Fehlererkennungscodeerzeugungsschaltung jeweils eine XOR-Schaltung beinhalten, wobei die XOR-Schaltungen auf einem Cyclic-Redundancy-Check-Code basieren und identische Fehlererkennungscodeerzeugungspolynome aufweisen.

7. Halbleiterspeicherelement nach Anspruch 4 oder 5, **dadurch gekennzeichnet**, dass die erste und die zweite Fehlererkennungscodeerzeugungsschaltung jeweils eine XOR-Schaltung aufweisen, die auf einem Cyclic-Redundancy-Check-Code basiert, wobei die erste Fehlererkennungscodeerzeugungsschaltung ein anderes Fehlererkennungscodeerzeugungspolynom als die zweite Fehlererkennungscodeerzeugungsschaltung aufweist.

8. Halbleiterspeicherelement nach einem der Ansprüche 1 bis 7, weiterhin aufweisend:  
einen ersten Parallel-Seriell-Wandler (1P2S), der die ersten Daten und das erste endgültige Fehlererkennungssignal in serielle Daten umwandelt und die seriellen Daten ausgibt; und  
einen zweiten Parallel-Seriell-Wandler (2P2S), der die zweiten Daten und das zweite endgültige Fehlererkennungssignal in serielle Daten umwandelt und die seriellen Daten ausgibt.

9. Halbleiterspeicherelement nach einem der Ansprüche 1 bis 8, weiterhin aufweisend:  
eine Mehrzahl von Pads (DQ), die in einer Outer Data Inner Control(ODIC)-Padstruktur angeordnet sind, wobei der erste Fehlererkennungscodegenerator benachbart zu einem ersten Datenpad angeordnet ist, das die ersten Daten und das erste endgültige Fehlererkennungssignal ausgibt, und wobei der zweite Fehlererkennungscodegenerator benachbart zu einem zweiten Datenpad angeordnet ist, das die zweiten Daten und das zweite endgültige Fehlererkennungssignal ausgibt.

10. Halbleiterspeicherelement nach einem der Ansprüche 1 bis 9, weiterhin aufweisend:  
einen dritten Fehlererkennungscodegenerator (ECCG3), der einen dritten Fehlererkennungscode für dritte Daten erzeugt, die extern angelegt werden, und der einen Anteil von Bits des dritten Fehlererkennungscode mit einem Anteil von Bits eines vierten Fehlererkennungscode kombiniert, um ein drittes endgültiges Fehlererkennungssignal zu erzeugen;  
einen vierten Fehlererkennungscodegenerator (ECCG4), der den vierten Fehlererkennungscode für vierte Daten erzeugt, die extern angelegt werden, und der verbleibende Bits des vierten Fehlererkennungscode, die sich von dem Anteil von Bits des vierten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des dritten Fehlererkennungscode, die sich von dem Anteil von Bits des dritten Fehlererkennungscode unterscheiden, kombiniert, um

ein viertes endgültiges Fehlererkennungssignal zu erzeugen; und  
einen Fehlerdetektor (ED), der ein extern angelegtes drittes endgültiges Fehlererkennungssignal mit dem dritten endgültigen Fehlererkennungssignal vergleicht und der ein extern angelegtes viertes endgültiges Fehlererkennungssignal, das extern angelegt ist, mit dem vierten endgültigen Fehlererkennungssignal vergleicht, um ein Fehlererkennungssignal zu erzeugen.

11. Halbleiterspeicherelement nach Anspruch 10, **dadurch gekennzeichnet**, dass der dritte Fehlererkennungscodegenerator aufweist:  
eine erste Fehlererkennungscodeerzeugungsschaltung (10), die den dritten Fehlererkennungscode für die dritten Daten erzeugt; und  
eine erste Kombinierschaltung (12), die den Anteil von Bits des dritten Fehlererkennungscode mit dem Anteil von Bits des vierten Fehlererkennungscode kombiniert, um das dritte endgültige Fehlererkennungssignal zu erzeugen.

12. Halbleiterspeicherelement nach Anspruch 11, **dadurch gekennzeichnet**, dass die erste Kombinierschaltung ein XOR-Gatter (12) ist, das eine XOR-Operation an oberen Bits des vierten Fehlererkennungscode und oberen Bits des dritten Fehlererkennungscode ausführt, um das dritte endgültige Fehlererkennungssignal zu erzeugen.

13. Halbleiterspeicherelement nach Anspruch 11 oder 12, **dadurch gekennzeichnet**, dass der vierte Fehlererkennungscodegenerator aufweist:  
eine zweite Fehlererkennungscodeerzeugungsschaltung (10), die den vierten Fehlererkennungscode für die vierten Daten erzeugt; und  
eine zweite Kombinierschaltung (12), die die verbleibenden Bits des vierten Fehlererkennungscode, die sich von dem Anteil von Bits des vierten Fehlererkennungscode unterscheiden, mit den verbleibenden Bits des dritten Fehlererkennungscode, die sich von dem Anteil von Bits des dritten Fehlererkennungscode unterscheiden, kombiniert, um das vierte endgültige Fehlererkennungssignal zu erzeugen.

14. Halbleiterspeicherelement nach Anspruch 13, **dadurch gekennzeichnet**, dass die zweite Kombinierschaltung ein XOR-Gatter (12) ist, das eine XOR-Operation an unteren Bits des vierten Fehlererkennungscode und unteren Bits des dritten Fehlererkennungscode ausführt, um das vierte endgültige Fehlererkennungssignal zu erzeugen.

15. Halbleiterspeicherelement nach Anspruch 13 oder 14, **dadurch gekennzeichnet**, dass die erste und die zweite Fehlererkennungscodeerzeugungsschaltung jeweils auf einem Cyclic-Redundancy-Check-Code basieren und dass die erste und die zweite Fehlererkennungscodeerzeugungsschal-

tung jeweils eine XOR-Schaltung aufweisen, wobei die XOR-Schaltungen ein identisches Fehlererkennungscodeerzeugungspolynom aufweisen.

16. Halbleiterspeicherelement nach Anspruch 13 oder 14, **dadurch gekennzeichnet**, dass die erste und die zweite Fehlererkennungscodeerzeugungsschaltung jeweils auf einem Cyclic-Redundancy-Check-Code basieren und dass die erste und die zweite Fehlererkennungscodeerzeugungsschaltung jeweils eine XOR-Schaltung aufweisen, wobei die XOR-Schaltungen unterschiedliche Fehlererkennungscodeerzeugungspolynome aufweisen.

17. Halbleiterspeicherelement nach einem der Ansprüche 10 bis 16, weiterhin aufweisend: einen ersten Seriell-Parallel-Wandler (1S2P), der die extern angelegten dritten Daten und das dritte endgültige Fehlererkennungssignal in parallele Daten umwandelt und die parallelen Daten ausgibt; und einen zweiten Seriell-Parallel-Wandler (1S2P), der die extern angelegten vierten Daten und das vierte endgültige Fehlererkennungssignal in parallele Daten umwandelt und die parallelen Daten ausgibt.

18. Halbleiterspeicherelement nach einem der Ansprüche 10 bis 17, **dadurch gekennzeichnet**, dass der Fehlerdetektor (ED) aufweist: ein erstes XOR-Gatter (XOR1), welches das extern angelegte dritte endgültige Fehlererkennungssignal mit dem dritten endgültigen Fehlererkennungssignal vergleicht, das von dem dritten Fehlererkennungscodegenerator ausgegeben wird, um ein erstes Vergleichsübereinstimmungssignal zu erzeugen; ein zweites XOR-Gatter (XOR2), welches das extern angelegte vierte endgültige Fehlererkennungssignal, das extern angelegt wird, mit dem vierten endgültigen Fehlererkennungssignal vergleicht, das von dem vierten Fehlererkennungscodegenerator ausgegeben wird, um ein zweites Vergleichsübereinstimmungssignal zu erzeugen; und ein OR-Gatter (OR), das eine OR-Operation an dem ersten und dem zweiten Vergleichsübereinstimmungssignal ausführt, um das Fehlererkennungssignal zu erzeugen.

19. Halbleiterspeicherelement nach einem der Ansprüche 4 bis 18, weiterhin aufweisend: einen dritten Fehlererkennungscodegenerator (ECCG3), der einen dritten Fehlererkennungscode für die ersten Daten erzeugt und der einen Anteil von Bits des dritten Fehlererkennungscodees mit einem Anteil von Bits eines vierten Fehlererkennungscodees kombiniert, um das erste endgültige Fehlererkennungssignal zu erzeugen; und einen vierten Fehlererkennungscodegenerator (ECCG4), der den vierten Fehlererkennungscode für die zweiten Daten erzeugt und der verbleibende Bits des vierten Fehlererkennungscodees, die sich von dem Anteil von Bits des vierten Fehlererkennungs-

codes unterscheiden, mit verbleibenden Bits des dritten Fehlererkennungscodees, die sich von dem Anteil von Bits des dritten Fehlererkennungscodees unterscheiden, kombiniert, um das zweite endgültige Fehlererkennungssignal zu erzeugen.

20. Halbleiterspeicherelement nach Anspruch 19, **dadurch gekennzeichnet**, dass der dritte Fehlererkennungscodegenerator aufweist: eine dritte Fehlererkennungscodeerzeugungsschaltung (10), die den dritten Fehlererkennungscode für die ersten Daten erzeugt; und eine dritte Kombinierschaltung (12), die den Anteil von Bits des dritten Fehlererkennungscodees mit dem Anteil von Bits des vierten Fehlererkennungscodees kombiniert, um das erste endgültige Fehlererkennungssignal zu erzeugen.

21. Halbleiterspeicherelement nach Anspruch 20, **dadurch gekennzeichnet**, dass die dritte Kombinierschaltung ein XOR-Gatter (12) ist, das eine XOR-Operation an oberen Bits des dritten Fehlererkennungscodees und an oberen Bits des vierten Fehlererkennungscodees ausführt, um das erste endgültige Fehlererkennungssignal zu erzeugen.

22. Halbleiterspeicherelement nach einem der Ansprüche 19 bis 21, **dadurch gekennzeichnet**, dass der vierte Fehlererkennungscodegenerator aufweist: eine vierte Fehlererkennungscodeerzeugungsschaltung (10), die den vierten Fehlererkennungscode für die zweiten Daten erzeugt; und eine vierte Kombinierschaltung (12), welche die verbleibenden Bits des vierten Fehlererkennungscodees, die sich von dem Anteil von Bits des vierten Fehlererkennungscodees unterscheiden, mit den verbleibenden Bits des dritten Fehlererkennungscodees, die sich von dem Anteil von Bits des dritten Fehlererkennungscodees unterscheiden, kombiniert, um das zweite endgültige Fehlererkennungssignal zu erzeugen.

23. Halbleiterspeicherelement nach Anspruch 22, **dadurch gekennzeichnet**, dass die vierte Kombinierschaltung ein XOR-Gatter (12) ist, das eine XOR-Operation an unteren Bits des vierten Fehlererkennungscodees und an unteren Bits des dritten Fehlererkennungscodees ausführt, um das zweite endgültige Fehlererkennungssignal zu erzeugen.

24. Halbleiterspeicherelement nach Anspruch 22 oder 23, **dadurch gekennzeichnet**, dass die erste bis vierte Fehlererkennungscodeerzeugungsschaltung jeweils auf einem Cyclic-Redundancy-Check-Code basieren, dass die erste und die dritte Fehlererkennungscodeerzeugungsschaltung jeweils eine erste XOR-Schaltung mit einem unterschiedlichen Fehlererkennungscodeerzeugungspolynom aufweisen und dass die zweite und die vierte Fehlererkennungscodeerzeugungsschaltung jeweils eine zweite XOR-Schaltung mit einem anderen unterschiedlichen

Fehlererkennungscodeerzeugungspolynom aufweisen.

25. Halbleiterspeicherelement nach Anspruch 24, weiterhin aufweisend:

eine Steuereinheit zum periodischen und abwechselnden Aktivieren des ersten und des zweiten Fehlererkennungscodegenerators sowie des dritten und des vierten Fehlererkennungscodegenerators.

26. Halbleiterspeicherelement, aufweisend:

- einen ersten Speicherzellenfeldblock (BLK1), der  $2n$  Bits erster Daten in einem ersten Bitstrukturmodus erzeugt und der  $n$  Bits der ersten Daten in einem zweiten Bitstrukturmodus erzeugt;

einen zweiten Speicherzellenfeldblock (BLK3), der  $2n$  Bits zweiter Daten in dem ersten Bitstrukturmodus erzeugt und der  $n$  Bits der zweiten Daten in dem zweiten Bitstrukturmodus erzeugt;

einen ersten Fehlererkennungscodegenerator (ECCG1), der obere  $n$  Bits der  $2n$  Bits der ersten Daten in dem ersten Bitstrukturmodus erzeugt, der einen ersten Fehlererkennungscode für die  $n$  Bits der ersten Daten in dem zweiten Bitstrukturmodus erzeugt und der einen Anteil von Bits des ersten Fehlererkennungscode mit einem Anteil von Bits eines dritten Fehlererkennungscode kombiniert, um einen Anteil von Bits eines ersten endgültigen Fehlererkennungscode zu erzeugen;

einen zweiten Fehlererkennungscodegenerator (ECCG2), der einen zweiten Fehlererkennungscode für untere  $n$  Bits der  $2n$  Bits der ersten Daten in dem ersten Bitstrukturmodus erzeugt und der einen Anteil von Bits des zweiten Fehlererkennungscode mit einem Anteil von Bits eines vierten Fehlererkennungscode kombiniert, um verbleibende Bits des ersten endgültigen Fehlererkennungscode zu erzeugen; die sich von dem Anteil von Bits des ersten endgültigen Fehlererkennungscode unterscheiden, zu erzeugen; einen dritten Fehlererkennungscodegenerator (ECCG3), der obere  $n$  Bits der  $2n$  Bits der zweiten Daten in dem ersten Bitstrukturmodus erzeugt, der den dritten Fehlererkennungscode für die  $n$  Bits der zweiten Daten in dem zweiten Bitstrukturmodus erzeugt und der verbleibende Bits des dritten Fehlererkennungscode, die sich von dem Anteil von Bits des dritten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des ersten Fehlererkennungscode, die sich von dem Anteil von Bits des ersten Fehlererkennungscode unterscheiden, kombiniert, um einen Anteil von Bits eines zweiten endgültigen Fehlererkennungscode zu erzeugen; und einen vierten Fehlererkennungscodegenerator (ECCG4), der den vierten Fehlererkennungscode für untere  $n$  Bits der  $2n$  Bits der zweiten Daten in dem ersten Bitstrukturmodus erzeugt und der verbleibende Bits des vierten Fehlererkennungscode, die sich von dem Anteil von Bits des vierten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des zweiten Fehlererkennungscode, die

sich von dem Anteil von Bits des zweiten Fehlererkennungscode unterscheiden, kombiniert, um verbleibende Bits des zweiten endgültigen Fehlererkennungscode zu erzeugen; die sich von dem Anteil von Bits des zweiten endgültigen Fehlererkennungscode unterscheiden, zu erzeugen.

27. Halbleiterspeicherelement nach Anspruch 26, **dadurch gekennzeichnet**, dass der erste Fehlererkennungscodegenerator aufweist:

eine erste Fehlererkennungscodeerzeugungsschaltung (10), die obere  $n$  Bits von den  $2n$  Bits der ersten Daten in dem ersten Bitstrukturmodus erzeugt und die  $2m$  Bits des ersten Fehlererkennungscode für die  $n$  Bits der ersten Daten in dem zweiten Bitstrukturmodus erzeugt; und

eine erste Kombinierschaltung (12), die obere  $m$  Bits von  $2m$  Bits des ersten Fehlererkennungscode mit oberen  $m$  Bits der  $2m$  Bits des dritten Fehlererkennungscode kombiniert, um obere  $m$  Bits des ersten endgültigen Fehlererkennungscode zu erzeugen.

28. Halbleiterspeicherelement nach Anspruch 27, **dadurch gekennzeichnet**, dass der zweite Fehlererkennungscodegenerator aufweist:

eine zweite Fehlererkennungscodeerzeugungsschaltung (10), die  $2m$  Bits des zweiten Fehlererkennungscode für untere  $n$  Bits der  $2n$  Bits der ersten Daten in dem ersten Bitstrukturmodus erzeugt; und eine zweite Kombinierschaltung (12), die obere  $m$  Bits von  $2m$  Bits des zweiten Fehlererkennungscode mit oberen  $m$  Bits von  $2m$  Bits des vierten Fehlererkennungscode kombiniert, um untere  $m$  Bits des zweiten endgültigen Fehlererkennungscode zu erzeugen.

29. Halbleiterspeicherelement nach Anspruch 28, **dadurch gekennzeichnet**, dass der dritte Fehlererkennungscodegenerator aufweist:

eine dritte Fehlererkennungscodeerzeugungsschaltung (10), die obere  $n$  Bits der  $2n$  Bits der zweiten Daten in dem ersten Bitstrukturmodus erzeugt und die  $2m$  Bits des dritten Fehlererkennungscode für die  $n$  Bits der zweiten Daten in dem zweiten Bitstrukturmodus erzeugt; und

eine dritte Kombinierschaltung (12), die untere  $m$  Bits von  $2m$  Bits des dritten Fehlererkennungscode mit unteren  $m$  Bits von  $2m$  Bits des ersten Fehlererkennungscode kombiniert, um obere  $m$  Bits des zweiten endgültigen Fehlererkennungscode zu erzeugen.

30. Halbleiterspeicherelement nach Anspruch 29, **dadurch gekennzeichnet**, dass der vierte Fehlererkennungscodegenerator aufweist:

eine vierte Fehlererkennungscodeerzeugungsschaltung (10), die  $2m$  Bits des vierten Fehlererkennungscode für untere  $n$  Bits der  $2n$  Bits der zweiten Daten in dem ersten Bitstrukturmodus erzeugt; und eine vierte Kombinierschaltung (12), die untere  $m$  Bits von  $2m$  Bits des vierten Fehlererkennungscode mit

unteren  $m$  Bits von  $2m$  Bits des zweiten Fehlererkennungscode kombiniert, um untere  $m$  Bits des zweiten endgültigen Fehlererkennungssignals zu erzeugen.

31. Halbleiterspeicherelement nach einem der Ansprüche 26 bis 30, weiterhin aufweisend:

einen ersten Parallel-Seriell-Wandler (1P2S), der die  $2n$  Bits der ersten Daten und  $2m$  Bits des ersten endgültigen Fehlererkennungssignals in dem ersten Bitstrukturmodus in serielle Daten umwandelt und die seriellen Daten ausgibt, und der  $n$  Bits der ersten Daten und  $m$  Bits des ersten endgültigen Fehlererkennungssignals in dem zweiten Bitstrukturmodus in serielle Daten umwandelt und die seriellen Daten ausgibt; und

einen zweiten Parallel-Seriell-Wandler (2P2S), der die  $2n$  Bits der zweiten Daten und  $2m$  Bits des zweiten endgültigen Fehlererkennungssignals in dem ersten Bitstrukturmodus in serielle Daten umwandelt und die seriellen Daten ausgibt.

32. Halbleiterspeicherelement nach Anspruch 30 oder 31, weiterhin aufweisend:

einen fünften Fehlererkennungscodegenerator, der die oberen  $n$  Bits der  $2n$  Bits der ersten Daten in dem ersten Bitstrukturmodus und einen fünften Fehlererkennungscode für die  $n$  Bits der ersten Daten in dem zweiten Bitstrukturmodus erzeugt und der einen Anteil von Bits des fünften Fehlererkennungscode mit einem Anteil von Bits eines siebten Fehlererkennungscode kombiniert, um einen weiteren Anteil von Bits des ersten endgültigen Fehlererkennungssignals zu erzeugen;

einen sechsten Fehlererkennungscodegenerator, der einen sechsten Fehlererkennungscode für die unteren  $n$  Bits der  $2n$  Bits der ersten Daten in dem ersten Bitstrukturmodus erzeugt und der einen Anteil von Bits des sechsten Fehlererkennungscode mit einem Anteil von Bits eines achten Fehlererkennungscode kombiniert, um verbleibende Bits des ersten endgültigen Fehlererkennungssignals, die sich von dem anderen Anteil von Bits des ersten endgültigen Fehlererkennungssignals unterscheiden, zu erzeugen;

einen siebten Fehlererkennungscodegenerator, der die oberen  $n$  Bits der  $2n$  Bits zweiter Daten in dem ersten Bitstrukturmodus und

den siebten Fehlererkennungscode für die  $n$  Bits der zweiten Daten in dem zweiten Bitstrukturmodus erzeugt und der verbleibende Bits des siebten Fehlererkennungscode, die sich von dem Anteil von Bits des siebten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des fünften Fehlererkennungscode, die sich von dem Anteil von Bits des fünften Fehlererkennungscode unterscheiden, kombiniert, um einen weiteren Anteil von Bits des zweiten endgültigen Fehlererkennungssignals zu erzeugen; und

einen achten Fehlererkennungscodegenerator, der den achten Fehlererkennungscode für die unteren  $n$

Bits der  $2n$  Bits der zweiten Daten in dem ersten Bitstrukturmodus erzeugt und der verbleibende Bits des achten Fehlererkennungscode, die sich von dem Anteil von Bits des achten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des sechsten Fehlererkennungscode, die sich von dem Anteil von Bits des sechsten Fehlererkennungscode unterscheiden, kombiniert, um verbleibende Bits des zweiten endgültigen Fehlererkennungssignals, die sich von dem weiteren Anteil von Bits des zweiten endgültigen Fehlererkennungssignals unterscheiden, zu erzeugen.

33. Halbleiterspeicherelement nach Anspruch 32, **dadurch gekennzeichnet**, dass der erste bis achte Fehlererkennungscodegenerator jeweils auf einem Cyclic-Redundancy-Check-Code basieren, dass der erste bis vierte Fehlererkennungscodegenerator jeweils eine erste XOR-Schaltung mit demselben ersten Fehlererkennungscodeerzeugungspolynom aufweisen und dass der fünfte bis achte Fehlererkennungscodegenerator jeweils eine zweite XOR-Schaltung mit demselben zweiten Fehlererkennungscodeerzeugungspolynom aufweisen, das von dem ersten Fehlererkennungscodeerzeugungspolynom verschieden ist.

34. Halbleiterspeicherelement nach Anspruch 33, weiterhin aufweisend:

eine Steuereinheit zum periodischen und abwechselnden Aktivieren des ersten bis vierten Fehlererkennungscodegenerators und des fünften bis achten Fehlererkennungscodegenerators.

35. Halbleiterspeicherelement nach einem der Ansprüche 32 bis 34, weiterhin aufweisend:

eine Mehrzahl von Pads (DQ), die in einer Outer Data Inner Control(ODIC)-Padstruktur angeordnet sind, wobei der erste, der zweite, der fünfte und der sechste Fehlererkennungscodegenerator benachbart zu einem ersten Datenpad angeordnet sind, das die ersten Daten und das erste endgültige Fehlererkennungssignal ausgibt; und

wobei der dritte, der vierte, der siebte und der achte Fehlererkennungscodegenerator benachbart zu einem zweiten Datenpad angeordnet sind, das die zweiten Daten und das zweite endgültige Fehlererkennungssignal ausgibt.

36. Halbleiterspeicherelement nach Anspruch 26, weiterhin aufweisend:

einen fünften Fehlererkennungscodegenerator, der obere  $n$  Bits von  $2n$  Bits extern angelegter dritter Daten in dem ersten Bitstrukturmodus und einen fünften Fehlererkennungscode für  $n$  Bits der extern angelegten dritten Daten in dem zweiten Bitstrukturmodus erzeugt und der einen Anteil von Bits des fünften Fehlererkennungscode und einen Anteil von Bits eines siebten Fehlererkennungscode kombiniert, um

einen Anteil von Bits eines dritten endgültigen Fehlererkennungssignals zu erzeugen;

einen sechsten Fehlererkennungscodegenerator, der einen sechsten Fehlererkennungscode für untere  $n$  Bits der  $2n$  Bits der extern angelegten dritten Daten in dem ersten Bitstrukturmodus erzeugt und der einen Anteil von Bits des sechsten Fehlererkennungscode mit einem Anteil von Bits eines achten Fehlererkennungscode kombiniert, um verbleibende Bits des dritten endgültigen Fehlererkennungssignals, die sich von dem Anteil von Bits des dritten endgültigen Fehlererkennungssignals unterscheiden, zu erzeugen;

einen siebten Fehlererkennungscodegenerator, der obere  $n$  Bits von  $2n$  Bits extern angelegter vierter Daten in dem ersten Bitstrukturmodus und den siebten Fehlererkennungscode für  $n$  Bits der extern angelegten vierten Daten in dem zweiten Bitstrukturmodus erzeugt und der verbleibende Bits des siebten Fehlererkennungscode, die sich von dem Anteil von Bits des siebten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des fünften Fehlererkennungscode, die sich von dem Anteil von Bits des fünften Fehlererkennungscode unterscheiden, kombiniert, um einen Anteil von Bits eines vierten endgültigen Fehlererkennungssignals zu erzeugen;

einen achten Fehlererkennungscodegenerator, der den achten Fehlererkennungscode für untere  $n$  Bits der  $2n$  Bits der extern angelegten vierten Daten in dem ersten Bitstrukturmodus erzeugt und der verbleibende Bits des achten Fehlererkennungscode, die sich von dem Anteil von Bits des achten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des sechsten Fehlererkennungscode, die sich von dem Anteil von Bits des sechsten Fehlererkennungscode unterscheiden, kombiniert, um verbleibende Bits des vierten endgültigen Fehlererkennungssignals, die sich von dem Anteil von Bits des vierten endgültigen Fehlererkennungssignals unterscheiden, zu erzeugen; und

einen Fehlerdetektor (ED), der das dritte endgültige Fehlererkennungssignal mit einem extern angelegten dritten endgültigen Fehlererkennungssignal vergleicht und der das vierte endgültige Fehlererkennungssignal mit einem extern angelegten vierten endgültigen Fehlererkennungssignal vergleicht, um ein Fehlererkennungssignal zu erzeugen.

37. Halbleiterspeicherelement nach Anspruch 36, weiterhin aufweisend:

einen ersten Seriell-Parallel-Wandler (1S2P), der die  $2n$  Bits der extern angelegten dritten Daten und  $2m$  Bits des dritten endgültigen Fehlererkennungssignals in dem ersten Bitstrukturmodus in parallele Daten umwandelt und die parallelen Daten ausgibt und der die  $n$  Bits der extern angelegten dritten Daten und  $m$  Bits des dritten endgültigen Fehlererkennungssignals in dem zweiten Bitstrukturmodus in parallele Daten umwandelt und die parallelen Daten ausgibt; und

einen zweiten Seriell-Parallel-Wandler (2S2P), der die  $2n$  Bits der extern angelegten vierten Daten und  $2m$  Bits des vierten endgültigen Fehlererkennungssignals in dem ersten Bitstrukturmodus in parallele Daten umwandelt und die parallelen Daten ausgibt.

38. Speichersystem, aufweisend:  
das Halbleiterspeicherelement nach einem der Ansprüche 1 bis 37; und

eine Speichersteuereinheit (200), umfassend:

einen dritten Fehlererkennungscodegenerator (ECCG3), der einen dritten Fehlererkennungscode für die ersten Daten erzeugt, die von dem Halbleiterspeicherelement ausgegeben werden, und der einen Anteil von Bits des dritten Fehlererkennungscode mit einem Anteil von Bits eines vierten Fehlererkennungscode kombiniert, um ein drittes endgültiges Fehlererkennungssignal zu erzeugen;

einen vierten Fehlererkennungscodegenerator (ECCG4), der den vierten Fehlererkennungscode für die zweiten Daten erzeugt, die von dem Halbleiterspeicherelement ausgegeben werden, und der verbleibende Bits des vierten Fehlererkennungscode, die sich von dem Anteil von Bits des vierten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des dritten Fehlererkennungscode, die sich von dem Anteil von Bits des dritten Fehlererkennungscode unterscheiden, kombiniert, um ein viertes endgültiges Fehlererkennungssignal zu erzeugen; und  
einen Fehlerdetektor (ED), der das erste endgültige Fehlererkennungssignal, das von dem Halbleiterspeicherelement ausgegeben wird, mit dem dritten endgültigen Fehlererkennungssignal vergleicht und der das zweite endgültige Fehlererkennungssignal, das von dem Halbleiterspeicherelement ausgegeben wird, mit dem vierten endgültigen Fehlererkennungssignal vergleicht, um ein Fehlererkennungssignal zu erzeugen.

39. Speichersystem nach Anspruch 38, **dadurch gekennzeichnet**, dass der erste bis vierte Fehlererkennungscodegenerator jeweils auf einem Cyclic-Redundancy-Check-Code basieren, dass der erste und der dritte Fehlererkennungscodegenerator jeweils eine erste XOR-Schaltung mit demselben ersten Fehlererkennungscodeerzeugungspolynom aufweisen und dass der zweite und der vierte Fehlererkennungscodegenerator jeweils eine zweite XOR-Schaltung mit demselben zweiten Fehlererkennungscodeerzeugungspolynom aufweisen, das von dem ersten Fehlererkennungscodeerzeugungspolynom verschieden ist.

40. Speichersystem, aufweisend:

das Halbleiterspeicherelement nach einem der Ansprüche 26 bis 37; und

eine Speichersteuereinheit (200), umfassend:

einen fünften Fehlererkennungscodegenerator, der die oberen  $n$  Bits von  $2n$  Bits der ersten Daten erzeugt, die von dem Halbleiterspeicherelement in dem

ersten Bitstrukturmodus ausgegeben werden, der einen fünften Fehlererkennungscode für die  $n$  Bits der ersten Daten, die von dem Halbleiterspeicherelement in dem zweiten Bitstrukturmodus ausgegeben werden, erzeugt und der einen Anteil von Bits des fünften Fehlererkennungscode mit einem Anteil von Bits eines siebten Fehlererkennungscode kombiniert, um einen Anteil von Bits eines dritten endgültigen Fehlererkennungssignals zu erzeugen;

einen sechsten Fehlererkennungscodegenerator, der einen sechsten Fehlererkennungscode für die unteren  $n$  Bits der  $2n$  Bits erster Daten in dem ersten Bitstrukturmodus erzeugt und der einen Anteil von Bits des sechsten Fehlererkennungscode mit einem Anteil von Bits eines achten Fehlererkennungscode kombiniert, um verbleibende Bits des dritten endgültigen Fehlererkennungssignals, die sich von dem Anteil von Bits des dritten endgültigen Fehlererkennungssignals unterscheiden, zu erzeugen;

einen siebten Fehlererkennungscodegenerator, der die oberen  $n$  Bits der  $2n$  Bits der zweiten Daten, die von dem Halbleiterspeicherelement in dem ersten Bitstrukturmodus ausgegeben werden, und den siebten Fehlererkennungscode für die  $n$  Bits der zweiten Daten in dem zweiten Bitstrukturmodus erzeugt und der verbleibende Bits des siebten Fehlererkennungscode, die sich von dem Anteil von Bits des siebten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des fünften Fehlererkennungscode, die sich von dem Anteil von Bits des fünften Fehlererkennungscode unterscheiden, kombiniert, um einen Anteil von Bits eines vierten endgültigen Fehlererkennungssignals zu erzeugen,

einen achten Fehlererkennungscodegenerator, der den achten Fehlererkennungscode für die unteren  $n$  Bits der  $2n$  Bits der zweiten Daten in dem ersten Bitstrukturmodus erzeugt und der verbleibende Bits des achten Fehlererkennungscode, die sich von dem Anteil von Bits des achten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des sechsten Fehlererkennungscode, die sich von dem Anteil von Bits des sechsten Fehlererkennungscode unterscheiden, kombiniert, um verbleibende Bits des vierten endgültigen Fehlererkennungssignals, die sich von dem Anteil von Bits des vierten endgültigen Fehlererkennungssignals unterscheiden, zu erzeugen; und

einen Fehlererkennungssignalgenerator, der das erste bis vierte endgültige Fehlererkennungssignal vergleicht, die von dem Halbleiterspeicherelement ausgegeben werden, um ein Fehlererkennungssignal zu erzeugen.

41. Speichersystem nach Anspruch 40, **dadurch gekennzeichnet**, dass der erste bis achte Fehlererkennungscodegenerator jeweils auf einem Cyclic-Redundancy-Check-Code basieren, dass der erste und der fünfte Fehlererkennungscodegenerator jeweils eine erste XOR-Schaltung mit demselben ersten Fehlererkennungscodeerzeugungspolynom auf-

weisen, dass der zweite und sechste Fehlererkennungscodegenerator jeweils eine zweite XOR-Schaltung mit demselben zweiten Fehlererkennungscodeerzeugungspolynom aufweisen, dass der dritte und siebte Fehlererkennungscodegenerator jeweils eine dritte XOR-Schaltung mit demselben dritten Fehlererkennungscodeerzeugungspolynom aufweisen und dass der vierte und der achte Fehlererkennungscodegenerator jeweils eine vierte XOR-Schaltung mit demselben vierten Fehlererkennungscodeerzeugungspolynom aufweisen.

42. Daten-Sende-/Empfangssystem, aufweisend:  
 einen Datensender; und  
 einen Datenempfänger,  
 wobei der Datensender aufweist:  
 einen ersten Fehlererkennungscodegenerator, der einen ersten Fehlererkennungscode für obere  $n$  Bits von  $2n$  Bits erster Daten erzeugt und der einen Anteil von Bits des ersten Fehlererkennungscode mit einem Anteil von Bits eines dritten Fehlererkennungscode kombiniert, um einen Anteil von Bits eines ersten endgültigen Fehlererkennungssignals zu erzeugen;  
 einen zweiten Fehlererkennungscodegenerator, der einen zweiten Fehlererkennungscode für untere  $n$  Bits der  $2n$  Bits erster Daten erzeugt und der einen Anteil von Bits des zweiten Fehlererkennungscode mit einem Anteil von Bits eines vierten Fehlererkennungscode kombiniert, um verbleibende Bits des ersten endgültigen Fehlererkennungssignals, die sich von dem Anteil von Bits des ersten endgültigen Fehlererkennungssignals unterscheiden, zu erzeugen;  
 einen dritten Fehlererkennungscodegenerator, der den dritten Fehlererkennungscode für obere  $n$  Bits von  $2n$  Bits zweiter Daten erzeugt und der verbleibende Bits des dritten Fehlererkennungscode, die sich von dem Anteil von Bits des dritten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des ersten Fehlererkennungscode, die sich von dem Anteil von Bits des ersten Fehlererkennungscode unterscheiden, kombiniert, um einen Anteil von Bits eines zweiten endgültigen Fehlererkennungssignals zu erzeugen; und  
 einen vierten Fehlererkennungscodegenerator, der einen vierten Fehlererkennungscode für untere  $n$  Bits der  $2n$  Bits zweiter Daten erzeugt und der verbleibende Bits des vierten Fehlererkennungscode, die sich von dem Anteil von Bits des vierten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des zweiten Fehlererkennungscode, die sich von dem Anteil von Bits des zweiten Fehlererkennungscode unterscheiden, kombiniert, um verbleibende Bits des zweiten endgültigen Fehlererkennungssignals, die sich von dem Anteil von Bits des zweiten endgültigen Fehlererkennungssignals unterscheiden, zu erzeugen, wobei der Datensender die ersten und die zweiten Daten in Einheiten von  $2n$  Bits

und die ersten und die zweiten endgültigen Fehlererkennungssignale ausgibt; und wobei der Datenempfänger aufweist:

- einen fünften Fehlererkennungscodegenerator, der einen fünften Fehlererkennungscode für die oberen  $n$  Bits der  $2n$  Bits der ersten Daten erzeugt und der einen Anteil von Bits des fünften Fehlererkennungscode mit einem Anteil von Bits eines siebten Fehlererkennungscode kombiniert, um einen Anteil von Bits eines dritten endgültigen Fehlererkennungssignals zu erzeugen;
- einen sechsten Fehlererkennungscodegenerator, der einen sechsten Fehlererkennungscode für die unteren  $n$  Bits der  $2n$  Bits erster Daten erzeugt und der einen Anteil von Bits des sechsten Fehlererkennungscode mit einem Anteil von Bits eines achten Fehlererkennungscode kombiniert, um verbleibende Bits des dritten endgültigen Fehlererkennungssignals, die sich von dem Anteil von Bits des dritten endgültigen Fehlererkennungssignals unterscheiden, zu erzeugen;
- einen siebten Fehlererkennungscodegenerator, der den siebten Fehlererkennungscode für die oberen  $n$  Bits der  $2n$  Bits der zweiten Daten erzeugt und der verbleibende Bits des siebten Fehlererkennungscode, die sich von dem Anteil von Bits des siebten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des fünften Fehlererkennungscode, die sich von dem Anteil von Bits des fünften Fehlererkennungscode unterscheiden, kombiniert, um einen Anteil von Bits eines vierten endgültigen Fehlererkennungssignals zu erzeugen;
- einen achten Fehlererkennungscodegenerator, der den achten Fehlererkennungscode für die unteren  $n$  Bits der  $2n$  Bits der zweiten Daten erzeugt und der verbleibende Bits des achten Fehlererkennungscode, die sich von dem Anteil von Bits des achten Fehlererkennungscode unterscheiden, mit verbleibenden Bits des sechsten Fehlererkennungscode, die sich von dem Anteil von Bits des sechsten Fehlererkennungscode unterscheiden, kombiniert, um verbleibende Bits des vierten endgültigen Fehlererkennungssignals, die sich von dem Anteil von Bits des vierten endgültigen Fehlererkennungssignals unterscheiden, zu erzeugen; und
- einen Fehlererkennungssignalgenerator, der das erste bis vierte endgültige Fehlererkennungssignal vergleicht, um ein Fehlererkennungssignal zu erzeugen.

43. Daten-Sende-/Empfangssystem nach Anspruch 42, **dadurch gekennzeichnet**, dass der erste bis achte Fehlererkennungscodegenerator jeweils auf einem Cyclic-Redundancy-Check-Code basieren, dass der erste und der fünfte Fehlererkennungscodegenerator jeweils eine erste XOR-Schaltung mit demselben ersten Fehlererkennungscodeerzeugungspolynom aufweisen, dass der zweite und der sechste Fehlererkennungscodegenerator jeweils eine zweite XOR-Schaltung mit demselben zweiten Fehlererkennungscodeerzeugungspolynom auf-

weisen, dass der dritte und der siebte Fehlererkennungscodegenerator jeweils eine dritte XOR-Schaltung mit demselben dritten Fehlererkennungscodeerzeugungspolynom aufweisen und dass der vierte und der achte Fehlererkennungscodegenerator jeweils eine vierte XOR-Schaltung mit demselben vierten Fehlererkennungscodeerzeugungspolynom aufweisen.

Es folgen 10 Seiten Zeichnungen

Anhängende Zeichnungen

FIG. 1  
(Stand der Technik)

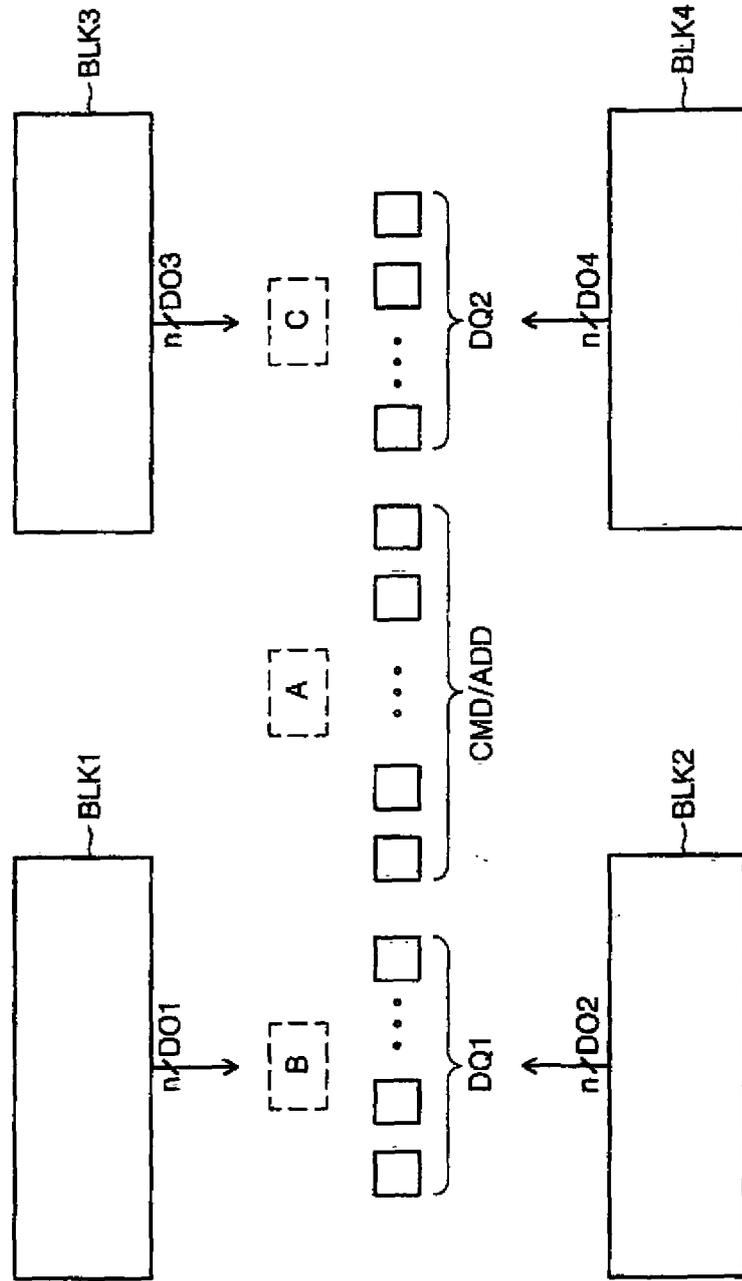


FIG. 2

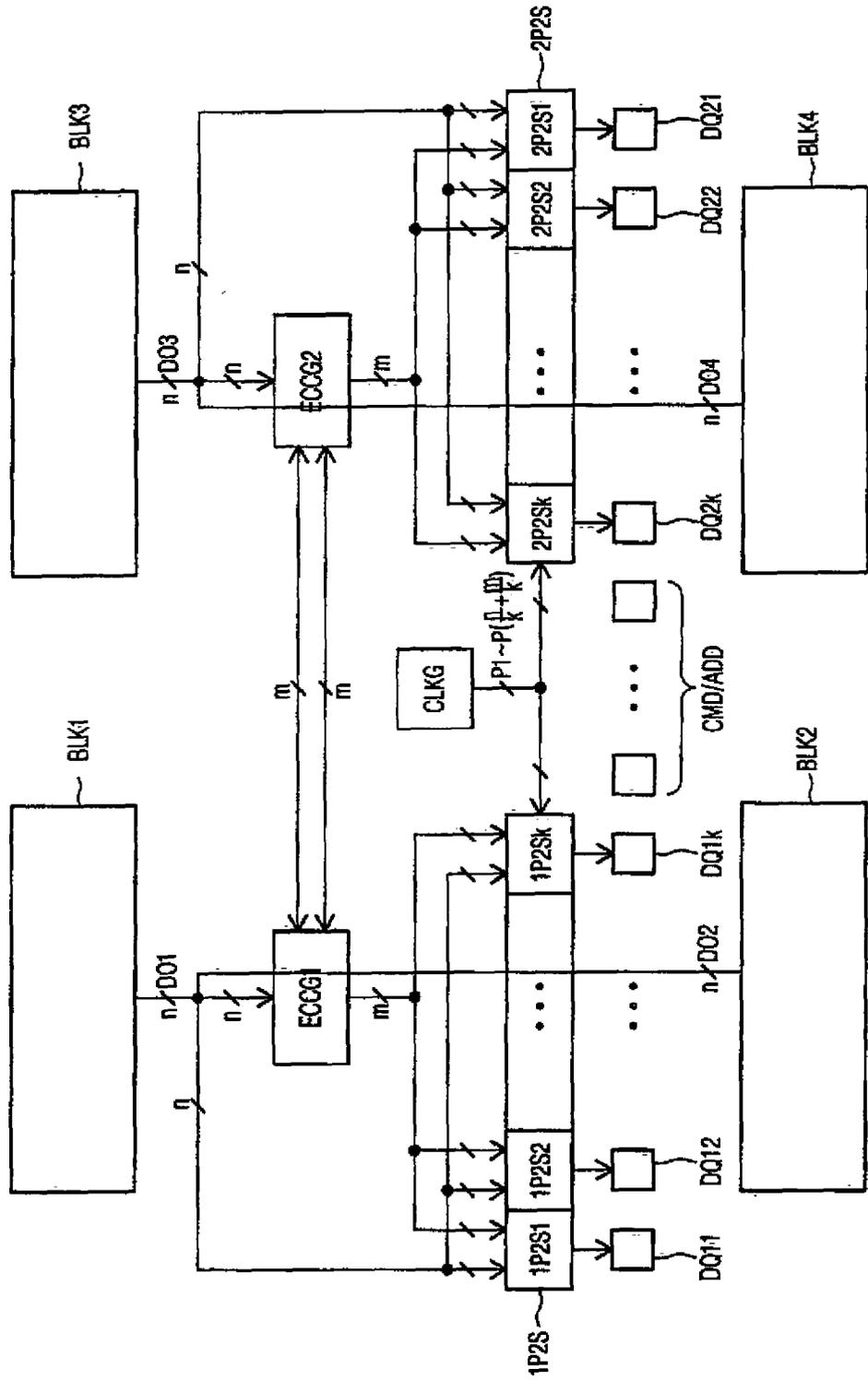


FIG. 3

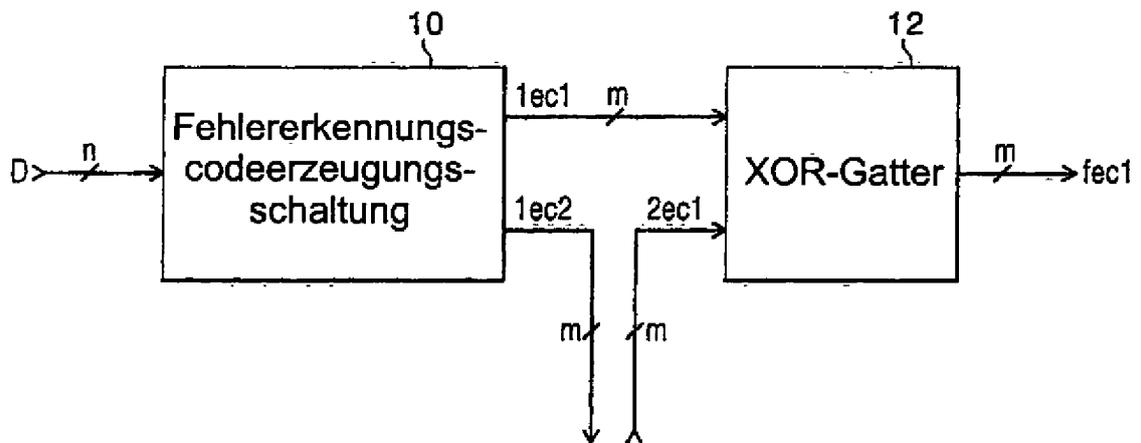


FIG. 4

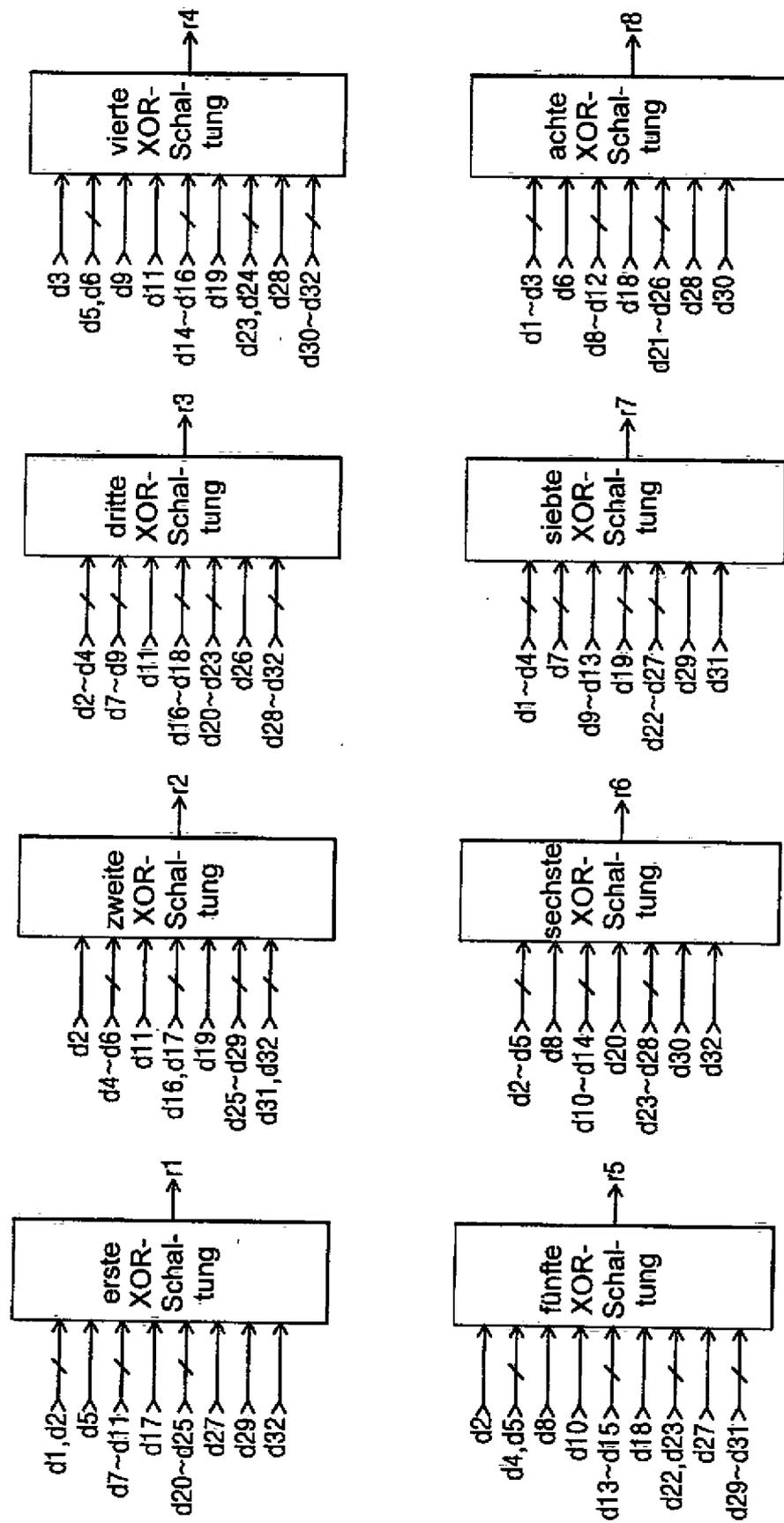


FIG. 5

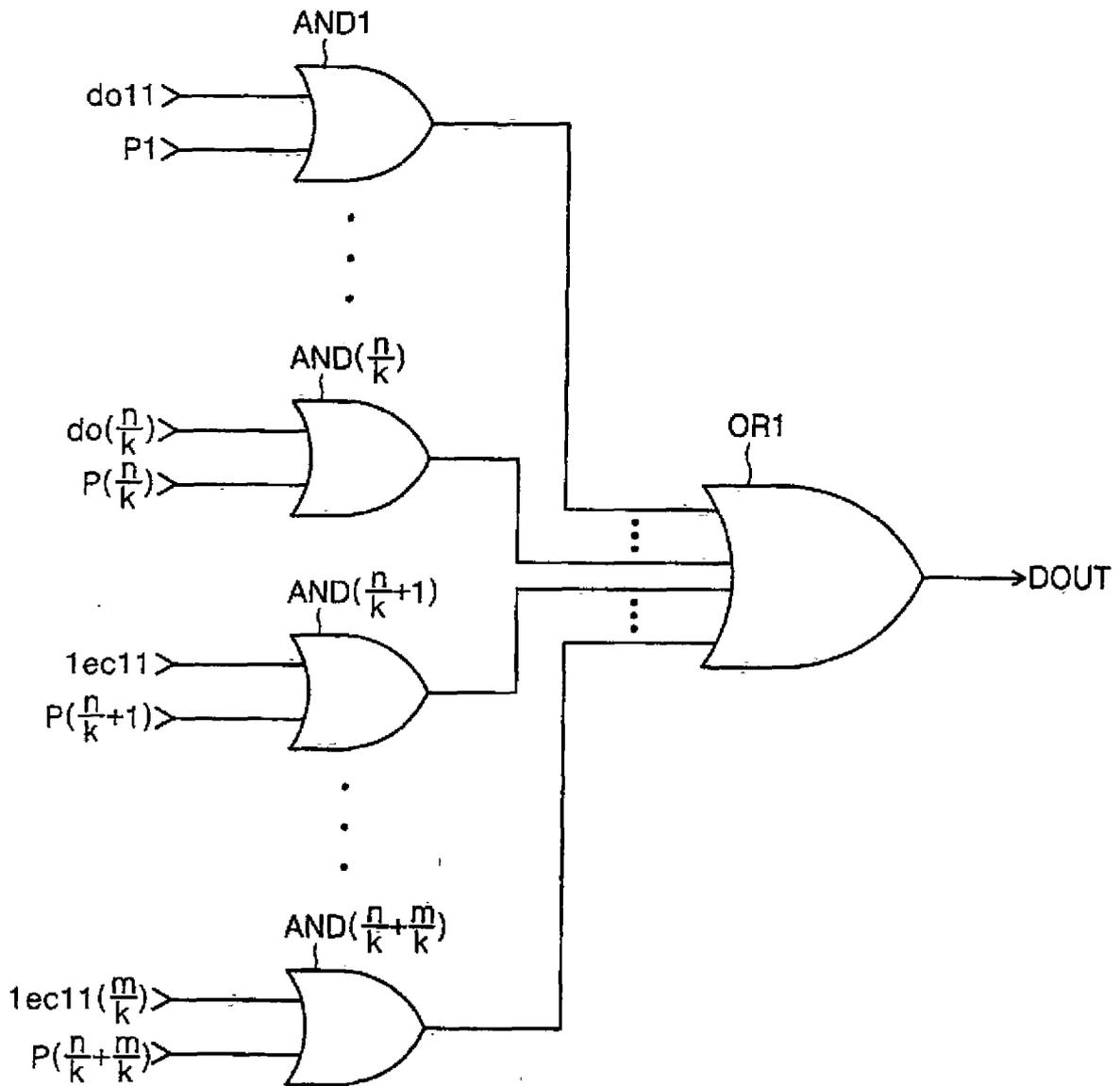






FIG. 8

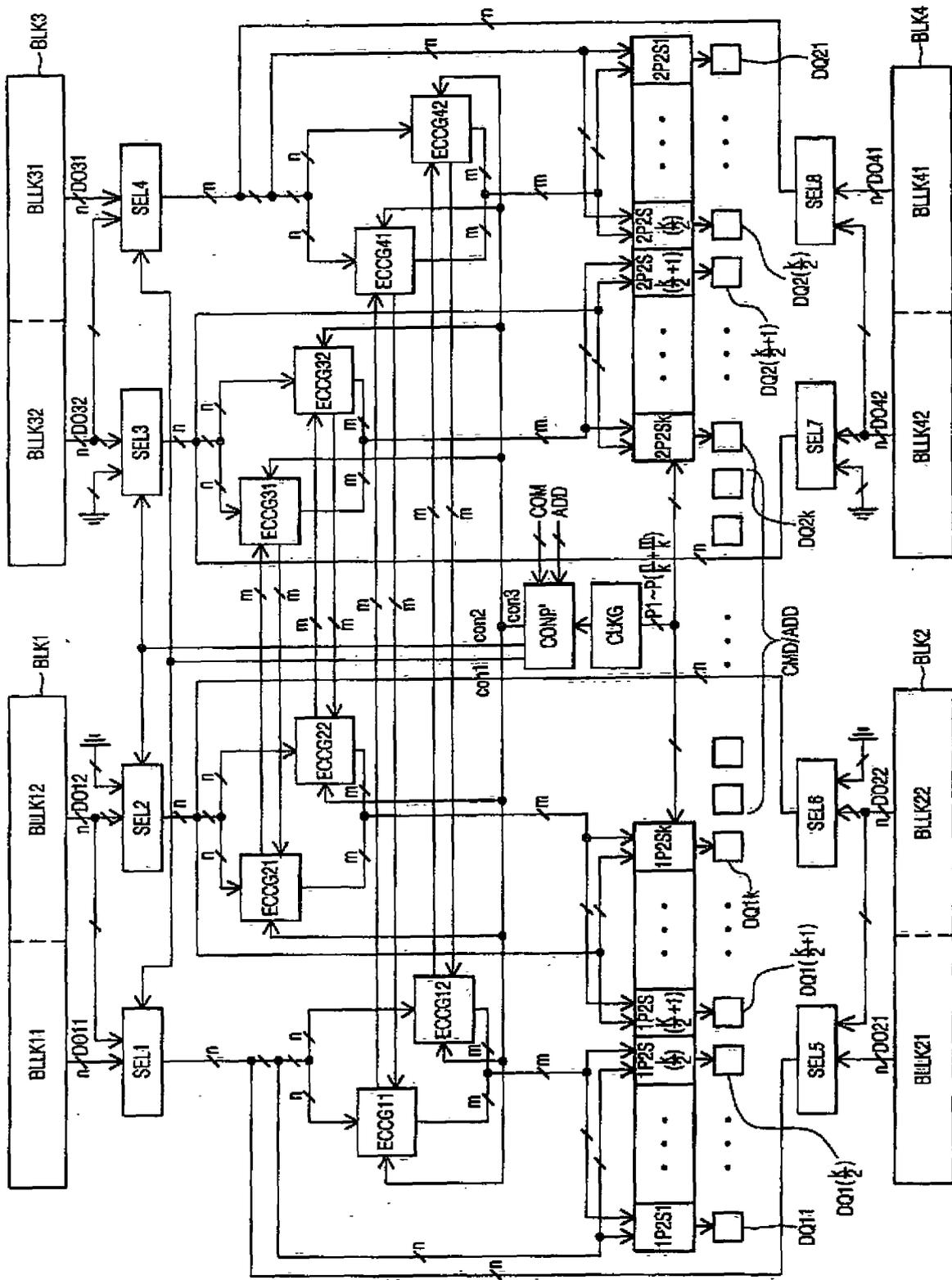


FIG. 9

