

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2005 年 2 月 24 日 (24.02.2005)

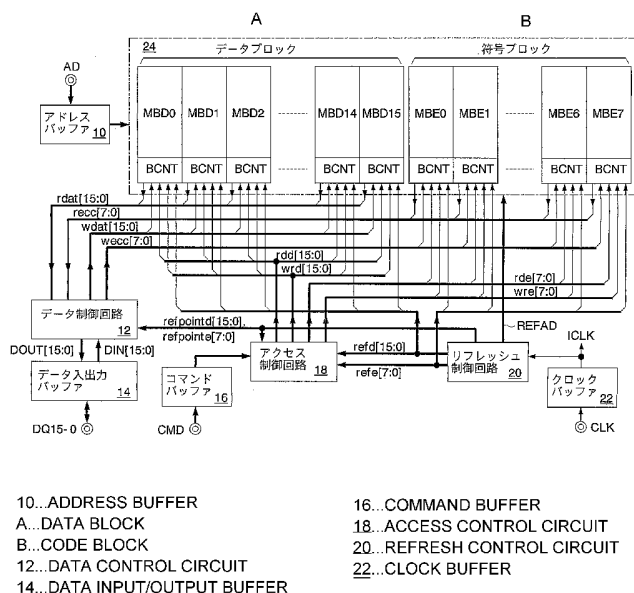
PCT

(10) 国際公開番号
WO 2005/017914 A1

- (51) 国際特許分類: G11C 29/00, 11/401 原区上小田中 4 丁目 1 番 1 号 富士通株式会社内 Kanagawa (JP).
- (21) 国際出願番号: PCT/JP2003/010412
- (22) 国際出願日: 2003 年 8 月 18 日 (18.08.2003)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (71) 出願人 (米国を除く全ての指定国について): 富士通株式会社 (FUJITSU LIMITED) [JP/JP]; 〒211-8588 神奈川県 川崎市中原区上小田中 4 丁目 1 番 1 号 Kanagawa (JP).
- (74) 代理人: 古谷 史旺 (FURUYA, Fumio); 〒160-0023 東京都 新宿区 西新宿 1 丁目 1 9 番 5 号 第 2 明宝ビル 9 階 Tokyo (JP).
- (81) 指定国 (国内): CN, JP, US.
- (84) 指定国 (広域): ヨーロッパ特許 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR).
- 添付公開書類:
— 国際調査報告書
- (72) 発明者; および
- (75) 発明者/出願人 (米国についてのみ): 江渡 聡 (ETO, Satoshi) [JP/JP]; 〒211-8588 神奈川県 川崎市 中
- 2 文字コード及び他の略語については、定期発行される各 PCT ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(54) Title: SEMICONDUCTOR MEMORY AND OPERATION METHOD OF SEMICONDUCTOR MEMORY

(54) 発明の名称: 半導体メモリおよび半導体メモリの動作方法



(57) Abstract: A data adder circuit adds a plurality of types of predictive data to be read from a refresh block to respective data read from another block to produce a plurality of read data sequences. An error correction circuit detects errors of the respective read data sequences and regards an error detection result, which is most likely to be correct, as true. The error correction circuit decodes, based on the error detection result regarded as true, the data to be read from the refresh block. Further, the error correction circuit corrects the error of the read data sequence corresponding to the error detection result regarded as true. As a result, the refresh operation can be hidden without extending the read cycle time, and at the same time the errors can be corrected. Correction of the errors of the data read from memory cells having bad data retention characteristics can extend the refresh request interval and hence reduce the power consumption in the standby mode.

[続 葉 有]



(57) 要約: データ付加回路は、リフレッシュブロックから読み出されるべき複数種の予想データを、他のブロックから読み出されるデータにそれぞれ付加して複数の読み出しデータ列を生成する。エラー訂正回路は、読み出しデータ列についてそれぞれエラーを検出し、最も確からしいエラー検出結果の一つを真とする。エラー訂正回路は、真としたエラー検出結果に基づいてリフレッシュブロックから読み出されるべきデータを復号する。さらに、エラー訂正回路は、真としたエラー検出結果に対応する読み出しデータ列のエラーを訂正する。この結果、読み出しサイクル時間を延ばすことなくリフレッシュ動作を隠すことができ、同時にエラーを訂正できる。データリテンション特性の悪いメモリセルから読み出されるデータのエラーを訂正することで、リフレッシュ要求間隔を延ばすことができ、スタンバイ中の消費電力を削減できる。

明細書

半導体メモリおよび半導体メモリの動作方法

5 技術分野

本発明は、リフレッシュ動作が必要なダイナミックメモリセルを有する半導体メモリおよびその動作方法に関する。

背景技術

- 10 近年、擬似 SRAM (Pseudo-SRAM) と呼ばれる半導体メモリが注目されている。擬似 SRAM は、DRAM のメモリセルを有し、メモリセルのリフレッシュ動作を内部で自動的に実行することで SRAM として動作する。ダイナミックメモリセルは、面積が小さいため、ビットコストが低く、かつ大容量の擬似 SRAM を開発できる。

- 一方、擬似 SRAM は、メモリセルのキャパシタに蓄えられる電荷によってデータを記憶するために、データは、電荷のリークなどによりやがて失われる。これを避けるために、擬似 SRAM は、所定期間毎にリフレッシュ動作が必要である。リフレッシュ動作は、メモリセルに保持されているデータをビット線に読み出し、ビット線上のデータをセンスアンプで増幅し、メモリセルに再書き込みする動作である。

- 20 リフレッシュ動作を実行するためのリフレッシュ要求は、外部アクセス要求と無関係に発生する。このため、リフレッシュ動作は、アクセス動作と競合する場合がある。擬似 SRAM は、SRAM インタフェースを有するため、この競合が発生したときにも、外部のシステムに対してリフレッシュ動作を隠す必要がある。近時、リフレッシュ動作を隠すために、書き込みデータと共にエラー訂正符号を記憶する半導体メモリが提案されている（例えば、特開 2003-51186 号公報）。
- 25 この半導体メモリでは、エラー訂正符号を用いることで、リフレッシュ動作中のメモリブロックからデータを読み出すことなく、読み出しデータを再生できる。また、読み出し要求とリフレッシュ要求が競合しないため、読み出しサイクル時間にリフレッシュ動作時間を含める必要はない。この結果、読み出しサイクル時

間を SRAM と同等にできる。

一方、擬似 SRAM は、リフレッシュ動作が必要なため、メモリセルがラッチで構成される SRAM に比べ、スタンバイ期間中の消費電力が大きい。スタンバイ期間中に消費される電力として、DC 成分と AC 成分がある。DC 成分は、リーク成分であり、トランジスタのオフ電流および内部回路（特に、電源回路）の動作のために意図的に流している電流等である。これ等、DC 成分は、半導体プロセスおよび回路構成に依存するため、擬似 SRAM と SRAM との差異は、議論できない。

擬似 SRAM のスタンバイ期間の消費電流のうち AC 成分は、主にリフレッシュ動作による電流である。このため、リフレッシュ動作電流の削減は、AC 成分を削減するために有効である。近時、リフレッシュ動作電流を削減するために、書き込みデータと共にエラー訂正符号を記憶する半導体メモリが提案されている。この半導体メモリは、エラー訂正符号によりメモリセルのデータ保持特性を実質的に向上し、リフレッシュ間隔を延ばすことで、リフレッシュ動作電流を削減している。（例えば、特開平 5-41084 号公報、特開 2002-56671 号公報、特開 2003-59290 号公報）。

しかしながら、上述した従来の擬似 SRAM で用いられるエラー訂正手法では、読み出しサイクル時間の削減またはスタンバイ電流の削減の一方しかできない。具体的には、読み出しサイクル時間（AC 特性）は、SRAM と同等にできても、スタンバイ電流（DC 特性）は SRAM と同等にできない。あるいは、スタンバイ電流は SRAM と同等にできても、読み出しサイクル時間は、SRAM と同等にできない。すなわち、SRAM と完全に互換性を有する擬似 SRAM は、実現されていない。この結果、例えば、システムに搭載されている SRAM をコスト削減のために擬似 SRAM に単純に置き換えるだけでは、システムを動作させることをは困難であった。

以下、本発明に関連する先行技術文献を列記する。

25 （特許文献）

- (1) 特開 2003-51186 号公報
- (2) 特開平 5-41084 号公報
- (3) 特開 2002-56671 号公報
- (4) 特開 2003-59290 号公報

発明の開示

本発明の目的は、アクセスサイクル時間を延ばすことなくリフレッシュ動作を隠すことができ、同時にスタンバイ期間の消費電力を削減できる擬似 SRAM を提供
5 することにある。

本発明の別の目的は、SRAM と完全な互換性を有する擬似 SRAM を提供することにある。

本発明の半導体メモリの一形態では、複数のデータブロックは、情報データを記憶するメモリセルを有する。複数の符号ブロックは、データブロックに記憶されたデータを復号するための符号データを記憶するメモリセルを有する。リフレッシュ制御回路は、データブロックおよび符号ブロックにリフレッシュ動作を順次実行させるためにリフレッシュ要求信号を出力する。データ付加回路は、リフレッシュブロックから読み出されるべき複数種の予想データを、他のブロックから読み出されるデータにそれぞれ付加して複数の読み出しデータ列を生成する。
10 ここで、リフレッシュブロックは、データブロックおよび符号ブロックのうちリフレッシュ動作によりデータを読み出せないブロックである。

エラー訂正回路は、読み出しデータ列についてそれぞれエラーを検出し、これ等のエラー検出結果から最も確からしい一つを真とする。また、エラー訂正回路は、真としたエラー検出結果に対応する読み出しデータ列のエラーを訂正する。
20 符号データは、例えば、水平垂直パリティ符号または拡大ハミング符号であり、符号データの最小ハミング距離は、4 以上である。

このように、データの一部を予想した複数の読み出しデータ列のエラー検出結果を互いに比較して真のエラーを検出し、真のエラーに基づいてブロックのいずれかで実際に発生したエラーを訂正することで、リフレッシュ動作を外部のシステムに対して隠すこと、およびエラー訂正を同時に達成できる。リフレッシュ動作を読み出し要求と競合することなく隠すことができるため、読み出しサイクル時間が長くなることを防止できる（AC 特性の改善）。
25

エラー訂正機能により、例えば、データを保持できる最小時間であるデータリテンション時間が短いメモリセルによって発生するエラーを訂正できる。このた

- め、リフレッシュ要求間隔を延ばすことができ、スタンバイ中の消費電力を削減できる（DC特性の改善）。具体的には、各メモリセルのリフレッシュ間隔を、ワーストのメモリセルのデータリテンション時間より長く設定しても、誤ったデータが読み出されることはない。したがって、本発明を、例えば、擬似SRAMに適用した場合、AC特性およびDC特性の両方をSRAMと同等以下にできる。換言すれば、SRAMと完全に互換性を有する擬似SRAMをユーザに提供できる。ユーザは、システム上に搭載されるSRAMを本発明が適用された擬似SRAMに置き換えるだけで、コストを削減できる。

- 本発明の半導体メモリの別の形態では、リフレッシュ制御回路は、データブロックおよび符号ブロックのいずれか1つにリフレッシュ動作を実行させるために、データブロックおよび符号ブロックにリフレッシュ要求信号を順次出力する。リフレッシュ要求に応答するリフレッシュ動作を1つのブロックでのみ実行することで、リフレッシュ動作により読み出せないデータを、容易に復元できる。

- 本発明の半導体メモリの別の形態では、アクセス制御回路は、読み出しコマンドに応答してリフレッシュブロックを除くデータブロックおよび符号ブロックに読み出し要求信号を出力する。すなわち、リフレッシュブロックは、読み出し動作を実行しない。このため、リフレッシュ要求と読み出し要求が競合することを防止でき、読み出しアクセス時間が増加することを防止できる。また、アクセス制御回路による読み出し動作の制御を容易にできる。

- 本発明の半導体メモリの別の形態では、アクセス制御回路は、書き込みコマンドに応答してデータブロックおよび符号ブロックに書き込み要求信号を出力する。データブロックおよび符号ブロック内の各ブロック制御回路は、リフレッシュ要求信号と書き込み要求信号が競合するときに、リフレッシュ動作を実行した後書き込み動作を遅れて実行する。このため、書き込みデータをリフレッシュブロックに対して確実に書き込むことができる。

本発明の半導体メモリの別の形態では、書き込みコマンドの最小供給間隔である外部書き込みサイクル時間は、各ブロックの実際の書き込み動作時間である内部書き込みサイクル時間より長く設定されている。このため、ブロック制御回路は、連続して供給される書き込み要求信号の間にリフレッシュ要求が発生した

場合にも、書き込み動作の遅れを徐々に取り戻すことができる。すなわち、書き込みデータをリフレッシュブロックに確実に書き込むことができる。

本発明の半導体メモリの別の形態では、あるリフレッシュブロックの最後のリフレッシュ要求信号から次のリフレッシュブロックに切り替わるまでの切り替え期間に、 n 回の外部書き込みサイクル時間が挿入可能であり、かつ1回のリフレッシュ動作と n 回の書き込み動作を実行可能である。例えば、切り替え期間は、外部書き込みサイクル時間 T_{cyc} と内部書き込みサイクル時間との差を δ とするときに、少なくとも $(T_{cyc}/\delta) \times T_{cyc}$ に設定される。このため、あるブロックにおいてリフレッシュ要求により遅れた書き込み動作は、そのブロックがリフレッシュブロックの期間に完了する。リフレッシュブロックから通常のブロックに切り替わった後、リフレッシュブロック中に供給された書き込み要求に対応する未完了の書き込み動作はない。このため、リフレッシュブロック以外のブロックにおいて、リフレッシュ動作と書き込み動作の競合に伴って読み出し動作が遅れることを防止でき、読み出しアクセス時間が長くなることを防止できる。

15 本発明の半導体メモリの別の形態では、符号ブロックは、不良を救済するロウ冗長回路およびコラム冗長回路の少なくともいずれかの代わりに形成される。不良のメモリセルから読み出される誤ったデータは、エラー訂正回路により訂正できる。冗長回路の形成が不要になるため、チップサイズを削減できる。

本発明の半導体メモリの別の形態では、エラー訂正回路は、読み出しデータ列のうちエラーの少ない読み出しデータ列を真として選択する。例えば、エラー訂正回路は、「エラー無し」、「データブロックのいずれかの1ビットエラー」、「符号ブロックのいずれかの1ビットエラーである符号エラー」および「2ビット以上のエラー」のいずれかを検出し、エラー判定の優先度を、エラー無し $>$ 1ビットエラー $>$ 符号エラー $>$ 2ビット以上のエラーとする。そして、エラー訂正回路は、優先度の高い項目を真とする。このため、エラー訂正回路のエラー判定論理を簡易に構成できる。

本発明の半導体メモリの別の形態では、エラー訂正回路は、訂正不可能なエラーを検出したときに、エラー信号をエラー端子に出力する。例えば、エラー訂正回路は、「2ビット以上のエラー」を検出したときに、エラー信号を出力する。

エラー信号により、半導体メモリをアクセスする外部のシステムは、読み出し動作のリトライ要求およびデータの廃棄等、エラーを回避する動作を実行できる。この結果、システムの信頼性を向上できる。

本発明の半導体メモリの別の形態では、書き戻し制御回路は、エラー訂正回路が訂正したデータをデータブロックおよび符号ブロックに書き戻す。メモリセル内のデータが電源ノイズ、ソフトエラー等により破壊されたときに、正しいデータを書き込むことで、半導体メモリの信頼性を向上できる。

本発明の半導体メモリの別の形態では、データブロックは、情報データを入出力する複数のデータ端子にそれぞれ対応して形成されている。このため、半導体メモリに一度に入出力されるデータ単位でエラーを訂正できる。この結果、符号データの生成および読み出しデータのエラー訂正の論理を簡易にでき、読み出しサイクル時間および書き込みサイクル時間が増加することを防止できる。

本発明の半導体メモリの別の形態では、情報データを入出力する複数のデータ端子に一度に供給される書き込みデータは、データブロックの一部に書き込まれる。エラー訂正回路は、書き込み動作中に、データブロックおよび符号ブロックからの読み出しデータを用いて情報データを復号し、復号した情報データの一部を書き込みデータに入れ替えて書き換えデータ列を生成する。エラー訂正回路は、書き換えデータ列から新たな符号データを求め、書き換えデータ列および新たな符号データをデータブロックおよび符号ブロックにそれぞれ書き込む。情報データのビット数を増やすことで、情報データのビット数に対する符号データのビット数を相対的に減らすことができる。この結果、符号ブロックの数を、データブロックの数に対して相対的に減らすことができ、チップサイズを削減できる。

図面の簡単な説明

図 1 は、本発明の第 1 の実施形態を示すブロック図である。

図 2 は、図 1 に示したデータ制御回路の詳細を示すブロック図である。

図 3 は、図 1 に示したリフレッシュ制御回路の動作を示すタイミング図である。

図 4 は、図 1 に示したリフレッシュ制御回路の動作の詳細を示すタイミング図である。

図 5 は、第 1 の実施形態の読み出し動作の概要を示すタイミング図である。

図 6 は、第 1 の実施形態の書き込み動作の概要を示すタイミング図である。

図 7 は、第 1 の実施形態のパリティ符号の生成規則を示す説明図である。

5 図 8 は、図 2 に示したエラー判定回路の動作の概要を示すフローチャートである。

図 9 は、図 2 に示したエラー訂正回路の動作の概要を示す説明図である。

図 10 は、第 1 の実施形態の読み出し動作の一例を示すタイミング図である。

図 11 は、第 1 の実施形態の読み出し動作の別の例を示すタイミング図である。

図 12 は、本発明の第 2 の実施形態を示すブロック図である。

10 図 13 は、図 12 に示したデータ制御回路の詳細を示すブロック図である。

図 14 は、図 13 に示したエラー訂正回路の RTL 記述を示す説明図である。

図 15 は、図 14 の RTL 記述から論理合成した結果を示す回路図である。

図 16 は、図 13 に示したシンドローム生成回路の RTL 記述を示す説明図である。

15 図 17 は、図 13 に示した符号化回路の RTL 記述を示す説明図である。

図 18 は、図 12 に示したデータ制御回路のエラー訂正動作を示す説明図である。

図 19 は、図 12 に示したデータ制御回路の別のエラー訂正動作を示す説明図である。

20 図 20 は、本発明の半導体メモリの第 3 の実施形態を示すブロック図である。

図 21 は、本発明の半導体メモリの第 4 の実施形態を示すブロック図である。

図 22 は、図 21 に示したデータ制御回路の詳細を示すブロック図である。

図 23 は、本発明の半導体メモリの第 5 の実施形態を示すブロック図である。

図 24 は、図 23 に示した擬似 SRAM の書き込み動作を示すタイミング図である。

25

発明を実施するための最良の形態

以下、本発明の実施形態を図面を用いて説明する。図中の二重丸は、外部端子を示している。図中、太線で示した信号線は、複数本で構成されている。太線が接続されているブロックの一部は、複数の回路で構成されている。外部端子を介

して供給される信号には、端子名と同じ符号を使用する。信号が伝達される信号線には、信号名と同じ符号を使用する。信号名の末尾の“z”は、正論理を示している。

図 1 は、本発明の半導体メモリの第 1 の実施形態を示している。この半導体メモリは、シリコン基板上に CMOS プロセスを使用してクロック同期式のシンクロナ
5 ス擬似 SRAM として形成されている。

擬似 SRAM は、アドレスバッファ 10、データ制御回路 12、データ入出力バッファ 14、コマンドバッファ 16、アクセス制御回路 18、リフレッシュ制御回路 20、クロックバッファ 22 およびメモリコア 24 を有している。

10 メモリコア 24 は、データブロック MBD0-15、符号ブロック MBE0-7 およびブロック制御回路 BCNT を有している。データブロック MBD0-15 は、データ端子 DQ15-0 にそれぞれ対応して形成されており、データ端子 DQ15-0 を介して供給される情報データ（以下、単にデータとも称する）をそれぞれ保持する。符号ブロック MBE0-7 は、データブロック MBD0-15 に保持されるデータにより生成される 8 ビットの誤
15 り訂正符号の各ビットをそれぞれ保持する。誤り訂正符号は、データブロックに記憶された情報データを復号するための符号データとして使用される。なお、特に図示していないが、メモリコア 24 は、ワード線単位で不良を救済するためのロウ冗長回路およびビット線単位で不良を救済するためのコラム冗長回路を有している。

20 ブロック制御回路 BCNT は、ブロック MBD0-15、MBE0-7 に対応してそれぞれ形成され、これ等ブロックの動作をそれぞれ制御する。ブロック制御回路 BCNT は、書き込み要求信号と、リフレッシュ要求信号（refd[15:0]、refe[7:0]のいずれか）とを同時に受けたとき、対応する書き込み要求信号（wrd[15:0]、wde[7:0]のいずれか）を、リフレッシュ動作が挿入される期間だけ順次遅らせて出力する。

25 データブロック MBD0-15 および符号ブロック MBE0-7 は、互いに同じ構造を有し、同じメモリ容量を有している。各ブロック MBD0-15、MBE0-7 は、ダイナミックメモリセルと、これ等ダイナミックメモリセルに接続されたワード線、ビット線とを有している。メモリセルは、データの論理値を電荷として保持するキャパシタと、キャパシタをビット線に接続する転送トランジスタとを有している。転送ト

ランジスタのゲートは、ワード線に接続されている。なお、この実施形態では、後述する図 7 に示すように、水平垂直パリティが、エラー訂正符号として採用されている。

5 アドレスバッファ 10 は、アドレス端子 AD を介してアドレス信号 AD を受信する。この擬似 SRAM は、ロウアドレス信号およびコラムアドレス信号を一度に受信するアドレスノンマルチプレクス方式を採用している。ロウアドレス信号は、ブロック MBD0-15、MBE0-7 内のワード線の選択に使用される。コラムアドレスは、ブロック MBD0-15、MBE0-7 内のビット線の選択に使用される。

10 クロックバッファ 22 は、クロック端子 CLK を介してクロック信号 CLK を受信し、受信した信号を内部クロック信号 ICLK として出力する。クロック信号 CLK は、例えば、擬似 SRAM が搭載されるシステム基板のシステムクロックである。内部クロック信号 ICLK は、擬似 SRAM 内のクロック同期が必要な回路に供給される。

リフレッシュ制御回路 20 は、特に図示していないが、タイマ、カウンタ、シフトレジスタおよびリフレッシュ要求生成回路を有している。タイマは、内部クロック信号 ICLK に同期して動作し、周期信号（パルス信号）を生成する。タイマの周期は、リフレッシュ要求の発生間隔に一致する。カウンタは、タイマからの周期信号に同期してカウント動作し、リフレッシュアドレス信号 REFAD を生成する。リフレッシュアドレス信号 REFAD は、リフレッシュ動作中にワード線を選択するために出力される。カウンタのビット数は、カウンタが一周したときに、ブロック MBD0-15、MBE0-7 内の全てのワード線が選択される数に設定されている。

20 シフトレジスタは、縦続接続された 24 個のラッチを有しており、最終段のラッチの出力は初段のラッチの入力に帰還されている。パワーオンリセット中に、ラッチの一つは高レベルにセットされ、残りのラッチは低レベルにリセットされる。シフトレジスタは、カウンタのオーバフロー毎にシフト動作し、リフレッシュ動作を実行するブロック MBD0-15、MBE0-7 を示すためのリフレッシュポイント信号 refpointd[15:0]、refpointe[7:0] をラッチからそれぞれ出力する。すなわち、シフトレジスタは、リフレッシュポイント信号 refpointd[15:0]、refpointe[7:0] の一つを順次活性化する。このため、ブロック MBD0-15、MBE0-7 の一つのみは、リフレッシュ動作を実行するリフレッシュブロックとして順次動

作する。リフレッシュ要求生成回路は、リフレッシュポイント信号 $\text{refpointd}[15:0]$ 、 $\text{refpointe}[7:0]$ の活性化期間に、周期信号に同期して、リフレッシュポイント信号 $\text{refpointd}[15:0]$ 、 $\text{refpointe}[7:0]$ にそれぞれ対応するリフレッシュ要求信号 $\text{refd}[15:0]$ 、 $\text{refe}[7:0]$ を出力する。

- 5 コマンドバッファ 16 は、コマンド端子 CMD を介してコマンド信号 CMD を受信し、受信したコマンド信号 CMD をアクセス制御回路 18 に出力する。コマンド信号 CMD は、例えば、チップイネーブル信号/CE、アウトプットイネーブル信号/OE およびライトイネーブル信号/WE 等の擬似 SRAM の動作を制御する信号である。コマンド端子 CMD を介して供給されるコマンドとして書き込み動作を実行するための書き込みコマンドおよび読み出し動作を実行するための読み出しコマンドがある。

- 15 アクセス制御回路 18 は、コマンド信号 CMD を解釈し、解釈結果に応じて読み出し動作を実行する読み出し要求信号 $\text{rdd}[15:0]$ 、 $\text{rde}[7:0]$ 、または書き込み要求信号 $\text{wrd}[15:0]$ 、 $\text{wde}[7:0]$ を出力する。但し、アクセス制御回路 18 は、リフレッシュポイント信号 $\text{refpointd}[15:0]$ 、 $\text{refpointe}[7:0]$ のいずれかの活性化に
15 応答して、対応する読み出し要求信号 ($\text{rdd}[15:0]$ 、 $\text{rde}[7:0]$ のいずれか) の出力をマスクする。すなわち、読み出し動作は、リフレッシュブロックで禁止され、他のブロックで実行される。リフレッシュブロックの読み出し動作を常に禁止することで、アクセス制御回路 18 の読み出し制御を簡易にできる。

- 20 データ入出力バッファ 14 は、書き込み動作中に、データ端子 DQ15-0 を介して受信したデータ信号を入力データ $\text{DIN}[15:0]$ として出力し、読み出し動作中に、データ制御回路 12 から供給される出力データ $\text{DOUT}[15:0]$ をデータ端子 DQ15-0 に出力する。

- 25 データ制御回路 12 は、書き込み動作中に、入力データ $\text{DIN}[15:0]$ を書き込みデータ $\text{wdat}[15:0]$ としてメモリア 24 に出力し、同時に入力データ $\text{DIN}[15:0]$ から生成されるエラー訂正符号 (書き込み符号データ) $\text{wecc}[7:0]$ をメモリア 24 に出力する。また、データ制御回路 12 は、読み出し動作中に、メモリア 24 から読み出しデータ $\text{rdat}[15:0]$ およびエラー訂正符号 (読み出し符号データ) $\text{recc}[7:0]$ を受け、リフレッシュブロックから読み出されるべき 1 ビットのデータ

を復号する。データ制御回路 12 は、同時に、データ保持特性が悪くデータが失われたメモリセルからの読み出しデータ（1 ビット）を復号する。エラー訂正された読み出しデータは、出力データ DOUT[15:0]として出力される。

5 なお、書き込み要求信号 wrd[15:0]、読み出し要求信号 rdd[15:0]、リフレッシュポイント信号 refpointd[15:0]、リフレッシュ要求信号 refd[15:0]、書き込みデータ wdat[15:0]、読み出しデータ rdat[15:0]、入力データ DIN[15:0]および出力データ DOUT[15:0]のビット番号は、データブロック MBD15-0 のブロック番号にそれぞれ対応する。同様に、書き込み要求信号 wde[7:0]、読み出し要求信号 rde[7:0]、リフレッシュポイント信号 refpointe[7:0]、リフレッシュ要求信号
10 refe[7:0]、エラー訂正符号 wecc[7:0]およびエラー訂正符号 recc[7:0]のビット番号は、符号ブロック MBE7-0 のブロック番号にそれぞれ対応する。

図 2 は、図 1 に示したデータ制御回路 12 の詳細を示している。

データ制御回路 12 は、メモリコア 24 からの読み出しデータ rdat[15:0]、recc[7:0]をエラー訂正するデータ訂正部 12a およびメモリコア 24 への入力
15 データ DIN[15:0]からエラー訂正符号を生成するデータ符号化部 12b を有している。データ訂正部 12a は、データ付加回路 26 およびエラー訂正回路 12c を有している。エラー訂正回路 12c は、符号化回路 28、30、エラー判定回路 32、34 およびエラー復号回路 36 を有している。データ符号化部 12b は、符号化回路 38 を有している。

20 データ付加回路 26 は、読み出し動作を実行できないリフレッシュブロックとして動作しているデータブロック（MBD0-15 のいずれか）からの読み出しデータを“論理 1”および“論理 0”と仮定して、読み出しデータ rdat0[15:0]、rdat1[15:0]をそれぞれ生成する。あるいは、データ付加回路 26 は、読み出し動作を実行できないリフレッシュブロックとして動作している符号ブロック
25 (MBE0-7 のいずれか)からのエラー訂正符号を“論理 1”および“論理 0”と仮定して、読み出しデータ recc0[7:0]、recc1[7:0]を生成する。すなわち、データ付加回路 26 は、リフレッシュブロックから読み出されるべきデータを“論理 1”または“論理 0”の予想データとして、他のブロックからの読み出しデータに付加し、読み出しデータ列 rdat0[15:0]、recc0[7:0]および読み出しデータ列 rdat1[15:0]、

recc1[7:0]を生成する。

- 符号化回路 28 は、読み出しデータ rdat0[15:0] (情報データ列) からエラー訂正符号 rrecc0[7:0] (第 1 符号データ) を生成する。符号化回路 30 は、読み出しデータ rdat1[15:0] (情報データ列) からエラー訂正符号 rrecc1[7:0] (第 2 符号データ) を生成する。

- エラー判定回路 32 は、エラー訂正符号 rrecc0[7:0] をエラー訂正符号 recc0[7:0] (符号データ) と比較することでエラーの有無およびエラーの種類を判定し、判定結果を比較結果信号 res0 として出力する。エラー判定回路 34 は、エラー訂正符号 rrecc1[7:0] をエラー訂正符号 recc1[7:0] (符号データ) と比較することでエラーの有無およびエラーの種類を判定し、判定結果を比較結果信号 res1 として出力する。比較結果信号 res0、res1 は、一致 (エラー無し)、ECC エラー (符号ブロック MBD の 1 ビットエラー; 符号エラー)、1 ビットエラー (データブロック MBE の 1 ビットエラー)、2 ビット以上のエラーのいずれかをそれぞれ示す。エラー項目の詳細は、後述する図 8、図 9 で説明する。

- エラー復号回路 36 は、比較結果信号 res0、res1 が示すエラー内容に応じて、比較結果信号 res0、res1 のどちらが真かを判定し、判定した比較結果信号 res0 または res1 に応じて読み出しデータのエラーを訂正し、出力データ[15:0]として出力する。出力データ[15:0]は、データ端子 DQ15-0 を介して擬似 SRAM の外部に出力される。比較結果信号 res0 が真と判定された場合、リフレッシュブロックから読み出されるべきデータは、“論理 0”である。比較結果信号 res1 が真と判定された場合、リフレッシュブロックから読み出されるべきデータは、“論理 1”である。

- データ符号化部 12b の符号化回路 38 は、入力データ[15:0] (書き込みデータ) からエラー訂正符号 wecc[7:0]を生成し、書き込みデータ wdat[15:0]とともにメモリコア 24 に出力する。

図 3 は、図 1 に示したリフレッシュ制御回路 20 の動作を示している。

リフレッシュ制御回路 20 は、リフレッシュポイント信号 refpointd[15:0]、refpointe[7:0]を順次高レベルに活性化する。各リフレッシュポイント信号 refpointd[15:0]、refpointe[7:0]の活性化間隔 (例えば、リフレッシュポイント

信号 `refpointd[0]` の活性化間隔 $T1$ は、ワーストのメモリセルのデータリテンション時間より長く設定されている。データリテンション時間は、メモリセルがデータを失うことなく保持できる時間である。

活性化間隔 $T1$ は、各メモリセルのリフレッシュ間隔にも対応している。このため、この実施形態では、ワーストのメモリセルは、リフレッシュ動作を実行するときに既にデータを消失する。しかし、失われたデータは、データ制御回路 12 のエラー訂正機能により復号できる。このため、メモリセルのリフレッシュ間隔を長くでき、リフレッシュ要求間隔およびリフレッシュ動作間隔を長くできる。この結果、特に、スタンバイ期間の消費電力を削減できる。

- 10 図 4 は、図 1 に示したリフレッシュ制御回路 20 の動作の詳細を示している。図 4 の波形は、図 3 に一点鎖線で示した領域に対応する。

例えば、リフレッシュ制御回路 20 は、リフレッシュポイント信号 `refpointe[7]` の活性化期間 $T2$ に、対応するデータブロック `MBD7` のリフレッシュ要求信号 `refe[7]` を所定回数高レベルに活性化する。このように、リフレッシュ要求に応答するリフレッシュ動作は、活性化されたリフレッシュポイント信号 `refpointe` (または `refpointd`) に対応するブロック `MBE` (または `MBD`) でのみ実行される。換言すれば、同時にアクセスされる 24 個のブロック `MBD15-0`、`MBE7-0` のうち、常に一つのブロックのみが、リフレッシュ動作を実行するリフレッシュブロックとして動作する。

- 20 リフレッシュ要求信号 `refe[7]` の活性化回数は、ブロック `MBE` のメモリセルを全てリフレッシュするために必要な回数である。例えば、リフレッシュ要求毎に 1 本のワード線が選択される場合、活性化回数は、各ブロックのワード線の本数に等しい。リフレッシュポイント信号 `refpointd[15:0]`、`refpointe[7:0]` は、リフレッシュブロックの最後のリフレッシュ要求信号 `refd` (または `refe`) の出力から次のリフレッシュブロックの最初のリフレッシュ要求信号 `refd` (または `refe`) の出力までに切り替えられる。最後のリフレッシュ要求信号の出力からリフレッシュポイント信号の切り替えまでの期間 $T3$ は、リフレッシュ要求と書き込み要求とが競合したときの処理に必要である。期間 $T3$ については、後述する図 6 で説明する。

図 5 は、第 1 の実施形態の読み出し動作の概要を示している。図 5 の波形は、図 4 に一点鎖線で示した領域に対応する。

この例では、擬似 SRAM は、コマンド端子 CMD から読み出しコマンドを連続して受信し、読み出し動作を連続して実行する。図 1 に示したアクセス制御回路 18 は、リフレッシュポイント信号 `refpointd[15:0]`、`refpointe[7:0]` のいずれかの活性化を受けている間、対応する読み出し要求信号 `rdd[15:0]`、`rde[7:0]` の出力をマスクする。このため、リフレッシュブロックでは、読み出し動作は実行されず、リフレッシュ動作のみが実行される。読み出し動作を実行しないリフレッシュブロックから読み出されるべきデータは、他のブロックからの読み出しデータにより復号される。このため、読み出し動作は、リフレッシュ動作により遅れない。すなわち、読み出しアクセス時間を長くすることなくリフレッシュ動作を外部に対して完全に隠すことができる。

図 6 は、第 1 の実施形態の書き込み動作の概要を示している。

この例では、擬似 SRAM は、コマンド端子 CMD から書き込みコマンドを連続して受信し、書き込み動作を連続して実行する。ブロック MBD15-0、MBE7-0 は、書き込み要求信号 `wrd[15:0]`、`wre[7:0]` に応答して書き込み信号 `wrz` をそれぞれ生成し、書き込み動作を実行する。書き込み信号 `wrz` の高レベル期間は、ブロック MBD15-0、MBE7-0 の書き込み動作時間を示す。

書き込み要求に応答する書き込み動作は、全てのブロック MBD15-0、MBE7-0 で実行される。このため、リフレッシュ要求 (`refd` または `refe`) と書き込み要求 (`wrd` または `wre`) との競合が発生する。図 1 に示したアクセス制御回路 18 は、リフレッシュ要求と書き込み要求とが競合したとき、リフレッシュ動作を優先して実行する (図 6 (a))。アクセス制御回路 18 は、リフレッシュ動作の完了後、待たせていた書き込み要求に対応する書き込み動作を順次実行する (図 6 (b))。

書き込みコマンドの最小供給間隔 (`wrd` の出力間隔; 外部書き込みサイクル時間) は、メモリコア 24 の書き込み動作時間 (`wrz` の最小出力間隔; 内部書き込みサイクル時間) より長く設計されている。このため、リフレッシュ動作の割り込みに伴う書き込み動作の遅れは、書き込み動作を所定回数連続して実行することで、取り戻せる。この例では、あるリフレッシュブロックの最後のリフレッシュ

5 ュ要求信号 refd (または refe) から次のリフレッシュブロックに切り替わるまでの切り替え時間 T4 の間に、8 回の外部書き込みサイクル時間が挿入可能であり、かつ 1 回のリフレッシュ動作と 8 回の書き込み動作を実行できる。切り替え時間 T4 は、リフレッシュ動作の割り込みに伴う書き込み動作の遅れを取り戻すまでの回復時間である。このため、書き込みコマンドが連続して供給される場合にも、リフレッシュブロックの書き込み動作は、そのリフレッシュブロックが切り替わる前に必ず完了できる。

10 この結果、リフレッシュブロックから通常データブロックまたは符号ブロックに切り替わった直後に、読み出しコマンドが供給される場合にも、読み出しデータの出力が遅れることを防止できる。すなわち、読み出しアクセス時間および読み出しサイクル時間が、書き込み動作の影響を受けて長くなることを防止できる。なお、切り替え時間 T4 は、外部書き込みサイクル時間 Tcyce と内部書き込みサイクル時間 Tcyi との差を δ とするとき、 $(Tcyce/\delta) \times Tcyce$ 以上であればよい。

15 また、切り替え時間 T4 は、次式 (1) の関係を満たすように設計してもよい。差 δ は、切り替え時間 T4 を内部書き込みサイクル時間 Tcyi に対して十分に大きく設定できる場合、無視できる程度になる。このため、外部書き込みサイクル時間 Tcyce を内部書き込みサイクル時間 Tcyi とほぼ等しくできる。すなわち、リフレッシュ動作を書き込み動作に優先して実行するためのアクセスペナルティは、
20 ほとんどない。

$$(T4/Tcyi) - (T4/(Tcyi + \delta)) \geq Tcyi \quad \cdots \cdots (1)$$

図 7 は、第 1 の実施形態のパリティ符号の生成規則を示している。

25 この実施形態では、最小ハミング距離が "4" の水平垂直パリティが、エラー訂正符号として採用されている。パリティは、偶数パリティ、奇数パリティのいずれでもよい。この符号は、1 ビットのエラーを訂正し、同時に 2 ビットのエラーを検出できる。

データブロック MBD0-15 内の同じアドレスに対応するデータを 4 行 4 列で示すとき、データブロック MBD0-15 は、行アドレス L1-L0 および列アドレス C1-C0 でそれぞれ表すことができる。例えば、データブロック MBD0 は、L1-L0="00"、C1-C0

=“00”で表せ、データブロック MBD10 は、 $L1-L0=“10”$ 、 $C1-C0=“10”$ で表せる。

例えば、符号ブロック MBE0 には、行アドレス $L0$ が“0”のデータブロック MBD0-3、8-11 のパリティが格納される（パリティ LP0）。符号ブロック MBE1 には、行アドレス $L0$ が“1”のデータブロック MBD4-7、12-15 のパリティが格納される（パリティ LP1）。符号ブロック MBE2 には、行アドレス $L1$ が“0”のデータブロック MBD0-5
5 のパリティが格納される（パリティ LP2）。符号ブロック MBE3 には、行アドレス $L1$ が“1”のデータブロック MBD8-15 のパリティが格納される（パリティ LP3）。

同様に、符号ブロック MBE4 には、列アドレス $C0$ が“0”のデータブロック MBD0、4、8、12、2、6、10、14 のパリティが格納される（パリティ CP0）。符号ブロック MBE5 には、列アドレス $C0$ が“1”のデータブロック MBD1、5、9、13、3、7、11、
10 15 のパリティが格納される（パリティ CP1）。符号ブロック MBE6 には、列アドレス $C1$ が“0”のデータブロック MBD0、4、8、12、1、5、9、13 のパリティが格納される（パリティ CP2）。符号ブロック MBE7 には、列アドレス $C1$ が“1”のデータブロック MBD2、6、10、14、3、7、11、15 のパリティが格納される（パリティ CP3）。

15 上述した水平垂直パリティ符号は、符号化する情報データが 16 ビット（本実施形態）、32 ビット、64 ビット、128 ビット、256 ビットするとき、それぞれ 8 ビット、10 ビット、12 ビット、14 ビット、16 ビット必要になる。すなわち、情報ビットが 2 の n 乗するとき、符号データは $2n$ ビット必要になる。これに対して、一般的な水平垂直パリティ符号は、符号化する情報データが 16
20 ビット、32 ビット、64 ビット、128 ビット、256 ビットするとき、それぞれ 8 ビット、12 ビット、16 ビット、24 ビット、32 ビット必要になる。このため、上述した水平垂直パリティ符号は、特に、情報データが 32 ビット以上のときに符号ブロックの数を一般の水平垂直パリティ符号より減らすことができる。

25 図 8 は、図 2 に示したエラー判定回路 32、34 の動作の概要を示している。ここでは、符号ブロック MBE0-7 から読み出されたエラー訂正符号 $recc[0:7]$ を $lp[0:3]$ 、 $cp[0:3]$ と表し、データブロック MBD0-7 から読み出されるデータに応じて図 2 に示した符号化回路 28（または 30）により生成されるエラー訂正符号 $rrecc0[0:7]$ （または $rrecc1[0:7]$ ）を $LP[0:3]$ 、 $CP[0:3]$ と表す。

まず、ステップ S10 において、cp[0:3]と CP[0:3]、lp[0:3]と LP[0:3]とがそれぞれ比較される。cp[0:3]と CP[0:3]、lp[0:3]と LP[0:3]が両方とも一致する場合、エラー無しと判定され、処理はステップ S20 に移行する。cp[0:3]と CP[0:3]、lp[0:3]と LP[0:3]の少なくともいずれかが一致しない場合、エラー有り

5 判定され、処理はステップ S30 に移行する。ステップ S20 において、エラー無しを示す比較結果信号 res0（または res1）＝"1000"が出力され、処理は終了する。

ステップ S30 において、次式（2）～（5）が全て成立するか否かが判定される。全ての式が成立する場合、1ビットエラーと判定され、処理はステップ S50 に移行する。少なくともいずれかの式が不成立の場合、2ビット以上のエラーと

10 判定され、処理はステップ S40 に移行する。ステップ S40 において、2ビット以上のエラーを示す比較結果信号 res0（または res1）＝"0001"が出力され、処理は終了する。

$$\text{cp}[0]+\text{cp}[1]+\text{CP}[0]+\text{CP}[1]=1 \quad \cdots \cdots (2)$$

$$\text{cp}[2]+\text{cp}[3]+\text{CP}[2]+\text{CP}[3]=1 \quad \cdots \cdots (3)$$

15 $\text{lp}[0]+\text{lp}[1]+\text{LP}[0]+\text{LP}[1]=1 \quad \cdots \cdots (4)$

$$\text{lp}[2]+\text{lp}[3]+\text{LP}[2]+\text{LP}[3]=1 \quad \cdots \cdots (5)$$

ステップ S50 において、cp[0:3]と CP[0:3]、lp[0:3]と LP[0:3]とがそれぞれ比較される。これ等 8 ビットの比較のうち、1ビットのみ不一致の場合、ECC コードエラーと判定され、処理はステップ S60 に移行する。それ以外の場合、1ビットエラーと決定され、処理はステップ S70 に移行する。ステップ S60 において、ECC コードエラーを示す比較結果信号 res0（または res1）＝"0100"が出力され、処理は終了する。ステップ S70 において、1ビットエラーを示す比較結果信号 res0（または res1）＝"00100"が出力され、処理は終了する。

20

図 9 は、図 2 に示したエラー復号回路 3 6 の動作の概要を示している。

25 エラー復号回路 3 6 は、エラー判定回路 3 2、3 4 から比較結果信号 res0、res1 を受信する。比較結果信号 res0、res1 は、図 8 に示したように、エラーの内容により、2進数の"1000（エラー無し）"、"0100（ECC エラー；符号ブロックの 1 ビットエラー）"、"0010（1 ビットエラー；データブロックの 1 ビットエラー）"、"0001（2 ビット以上のエラー）"のいずれかを示す。

図9 (a)、(b)、(c)は、データブロック MBD0-15 のいずれかがリフレッシュブロックとして動作し、このリフレッシュブロックから読み出されるべきデータが“論理0”の場合を示している。図9 (a)において、読み出し動作でエラーがない場合、リフレッシュブロックからの読み出しデータを“論理0”とした比較結果信号 res0 は、エラー無しを示し、リフレッシュブロックからの読み出しデータを“論理1”とした比較結果信号 res1 は、1ビットエラー（リフレッシュブロックのみエラー）を示す。

図9 (b)において、読み出し動作でリフレッシュブロックを除くデータブロック MBD0-15 のいずれかに1ビットエラーがある場合、比較結果信号 res0 は、1ビットエラーを示し、比較結果信号 res1 は、2ビット以上のエラー（リフレッシュブロックの1ビットおよびデータブロックの1ビットエラー）を示す。図9 (c)において、読み出し動作で符号ブロック MBE0-7 のいずれかに1ビットエラーがある場合、比較結果信号 res0 は、ECC エラーを示し、比較結果信号 res1 は、2ビット以上のエラー（リフレッシュブロックの1ビットおよびECCエラー）を示す。

図9 (d)、(e)、(f)は、データブロック MBD0-15 のいずれかがリフレッシュブロックとして動作し、このリフレッシュブロックから読み出されるべきデータが“論理1”の場合を示している。図9 (d)、(e)、(f)では、リフレッシュブロックから読み出されるべきデータの論理値は、図9 (a)、(b)、(c)と逆になる。このため、比較結果信号 res0、res1 は、図9 (a)、(b)、(c)の逆になる。

図9 (g)、(h)、(i)は、符号ブロック MBE0-7 のいずれかがリフレッシュブロックとして動作し、このリフレッシュブロックから読み出されるべきデータが“論理0”の場合を示している。図9 (g)において、読み出し動作でエラーがない場合、リフレッシュブロックからの読み出しデータを“論理0”とした比較結果信号 res0 は、エラー無しを示し、リフレッシュブロックからの読み出しデータを“論理1”とした比較結果信号 res1 は、ECC エラー（リフレッシュブロックのみエラー）を示す。

図9 (h)において、読み出し動作でデータブロック MBD0-15 のいずれかに1ビットエラーがある場合、比較結果信号 res0 は、1ビットエラーを示し、比較結

果信号 res1 は、2 ビット以上のエラー（リフレッシュブロックの 1 ビットおよびデータブロックの 1 ビットエラー）を示す。図 9（i）において、読み出し動作でリフレッシュブロックを除く符号ブロック MBE0-7 のいずれかに 1 ビットエラーがある場合、比較結果信号 res0 は、ECC エラーを示し、比較結果信号 res1 は、
5 2 ビット以上のエラー（リフレッシュブロックの 1 ビットおよび ECC エラー）を示す。

図 9（j）、（k）、（l）は、符号ブロック MBE0-7 のいずれかがリフレッシュブロックとして動作し、このリフレッシュブロックから読み出されるべきデータが“論理 1”の場合を示している。図 9（j）、（k）、（l）では、リフレッシュブロックから読み出されるべきデータの論理値は、図 9（g）、（h）、（i）
10 と逆になる。このため、比較結果信号 res0、res1 は、図 9（g）、（h）、（i）の逆になる。

エラー復号回路 36 は、比較結果信号 res0、res1 のいずれかを真のエラーとし、この結果に基づいてエラーを訂正する。この際、エラー復号回路 36 は、エラー判定の優先度を高い順から、エラー無し＞ECC エラー＞1 ビットエラー＞2 ビット以上のエラーとする。すなわち、エラーの少ない比較結果信号 res0、res1 のいずれかが真とされる。このため、図 9（a）、（d）、（g）、（j）では、エラー無しと判定される。図 9（b）、（e）、（h）、（k）では、1 ビットエラーと判定される。図 9（c）、（f）、（i）、（l）では、ECC エラーと判定
15 20 される。

エラー復号回路 36 は、優先度の高い比較結果信号 res0、res1 に対応する論理値をリフレッシュブロックから読み出されるべきデータの論理値とし、出力データ DOUT[15:0]として出力する。具体的には、res0 の優先度が res1 より高い場合、リフレッシュブロックに対応する出力データ DOUT のビットは、“論理 0”に設定
25 される。res1 の優先度が res0 より高い場合、リフレッシュブロックに対応する出力データ DOUT のビットは、“論理 1”に設定される。

さらに、エラー復号回路 36 は、データブロック MBD0-15 のいずれかに 1 ビットエラーが存在すると判定した場合、このエラーを訂正する。エラーのあるデータブロック MBD0-15（1 ビットエラーを発生しているエラーアドレス）は、次式

(6)、(7)にエラー訂正符号 $\text{recc}[0:7]$ を代入して求められる。例えば、 $(L1, L0) = "01"$ 、 $(C1, C0) = "11"$ の場合、図 7 に示したように、エラーブロックは、データブロック MBD7 である。このとき、エラー復号回路 36 は、データブロック MBD7 から読み出されたデータの論理値を反転してデータ出力バッファ 14 に出力する。

5 なお、この実施形態では、エラーを訂正したデータは、ブロックに書き込まれない。これは、1 ビットエラーおよび ECC エラーは、そのメモリセルのデータリテンション時間の不足あるいは物理的な不良と考えられるためである。この種のエラーに対してメモリセルに正しいデータを書き込んでも、エラーは再び発生するため、処理時間の無駄になる。

10 $(L1, L0) = (\text{lp}[3] + \text{LP}[3], \text{lp}[1] + \text{LP}[1]) \cdots \cdots (6)$

$(C1, C0) = (\text{cp}[3] + \text{CP}[3], \text{cp}[1] + \text{CP}[1]) \cdots \cdots (7)$

図 10 は、第 1 の実施形態の読み出し動作の一例（シミュレーション結果）を示している。

この例では、データブロック MBD10 がリフレッシュブロックとして動作する（図 10 (a)）。擬似 SRAM は、読み出しコマンド RDZ とともに、16 進数の "0" から "F" のロウアドレス MBAD を読み出しアドレスとして順次受け、読み出し動作を連続して実行する（図 10 (b)）。

読み出しアドレス "0"、"1"、"2"、...、"D"、"E"、"F" が示すデータブロック MBD0-15 のメモリセルには、予め 16 進数のデータ "0000"、"1111"、
20 "2222"、...、"DDDD"、"EEEE"、"FFFF" がそれぞれ書き込まれている。読み出しアドレス "0"、"1"、"2"、...、"D"、"E"、"F" が示す符号ブロック MBE0-7 のメモリセルには、予め 16 進数のデータ "00" が書き込まれている。しかし、この例では、符号ブロック MBE6 に 1 ビットエラー（ECC エラー）が発生するため、読み出しデータ $\text{recc}[7:0]$ は、16 進数の "40" になる（図 10 (c)）。なお、
25 実際の異なるアドレスの読み出し動作において、符号ブロック MBE6 に連続して 1 ビットエラーが発生することはまれである。

図 1 に示したリフレッシュ制御回路 20 は、読み出しアドレス "2" の読み出し動作中に、リフレッシュ要求信号 $\text{refd}[10]$ を出力する（図 10 (d)）。メモリコア 24 は、リフレッシュ要求信号 $\text{refd}[10]$ に応答して、データブロック MBD10

のリフレッシュ動作を実行する。

リフレッシュブロックとして動作するデータブロック MBD10 からデータは読み出されないため、読み出しデータ rdat[15:0]の第10ビットは、例えば“論理0”として読み出される。このため、第10ビットの期待値が“論理1”のとき（例えば、“4444”、“5555”など）、リフレッシュブロックの読み出しエラーが発生する。
5 このとき、エラー復号回路36は、2ビット以上のエラーを示す比較結果信号 res0 (“0001”) と、ECC エラーを示す比較結果信号 res1 (“0100”) を受け、比較結果信号 res1 の優先度が高いと判定し、ECC エラーの発生を検出する(図10(e、f))。

10 また、エラー復号回路36は、比較結果信号 res1 の優先度が高いため、リフレッシュブロックから読み出されるべきデータの論理値を“論理1”と判定し、アシユームフラグ ASMFLG を“論理1”にセットする(図10(g、h))。アシユームフラグ ASMFLG は、エラー復号回路36内で使用される内部信号である。アシユームフラグ ASMFLG のセットにより、出力データ[15:0]の第10ビットは、“論理1”
15 に設定される。すなわち、読み出し動作を実行しないリフレッシュブロックからのデータが再生される。

一方、読み出しデータ rdat[15:0]の第10ビットの期待値が“論理0”のとき（例えば、“0000”、“1111”、“2222”、“3333”など）、リフレッシュブロックの読み出しエラーは発生しない。このとき、エラー復号回路36は、ECC エラーを示す比較結果信号 res0 (“0100”) と、2ビット以上のエラーを示す比較結果信号 res1 (“0001”) を受け、比較結果信号 res0 の優先度が高いと判定し、ECC エラーの発生を検出する(図10(i、j、k))。
20

また、エラー復号回路36は、比較結果信号 res0 の優先度が高いため、リフレッシュブロックから読み出されるべきデータの論理値を“論理0”と判定し、アシユームフラグ ASMFLG を“論理0”にセットする(図10(l、m、n))。アシユームフラグ ASMFLG にセットにより、出力データ[15:0]の第10ビットは、“論理0”に設定される。すなわち、読み出し動作を実行しないリフレッシュブロックからのデータが再生される。
25

図11は、第1の実施形態の読み出し動作の別の例(シミュレーション結果)

を示している。図 10 と同じ動作については、詳細な説明を省略する。

この例では、図 10 と同様に、擬似 SRAM は、読み出しコマンド RDZ とともに、16 進数の“0”から“F”のロウアドレス MBAD を読み出しアドレスとして順次受け、読み出し動作を連続して実行する（図 11 (a)）。リフレッシュブロックは、連続する読み出し動作の途中で、データブロック MBD10 からデータブロック MBD11 に切り替わる（図 11 (b)）。データブロック MBD0-15、符号ブロック MBE0-7 に予め書き込まれているデータは、図 10 と同じである。この例では、データブロック MBD12 に 1 ビットエラーが発生し、読み出しデータ rdat[15:0]の第 12 ビットは、読み出し動作毎に“0”および“1”に変化する。なお、異なるアドレスの読み出し動作において、符号ブロック MBD12 に連続して 1 ビットエラーが発生することはまれである。

リフレッシュブロックとして動作するデータブロック MBD10 からデータは読み出されないため、読み出しデータ rdat[15:0]の第 10 ビットは、例えば“論理 0”として読み出される。同様に、リフレッシュブロックとして動作するデータブロック MBD11 からデータは読み出されないため、読み出しデータ rdat[15:0]の第 11 ビットは、例えば“論理 1”として読み出される。

上述した図 10 と同様に、データ制御回路 12 のエラー判定回路 32、34 は、読み出しデータ rdat[15:0]およびエラー訂正符号 recc[7:0]から比較結果信号 res0、res1 を生成する。比較結果信号 res0、res1 は、2 ビット以上のエラー (“0001”) またはデータブロックの 1 ビットエラー (“0010”) を示す。エラー復号回路 36 は、比較結果信号 res0、res1 に基づいて、リフレッシュ動作の実行により読み出せないデータを再生する。また、エラー復号回路 36 は、上述の式(6)、(7)に示したように、エラー訂正符号 recc[7:0]から 1 ビットエラーが発生したブロックアドレスを求め、そのブロックアドレスから読み出されたデータを反転する。すなわち、データブロック MBD12 で発生した 1 ビットエラーが訂正される。

以上、本実施形態では、リフレッシュブロックから読み出されるべき予想データ（論理 1、論理 0）を、他のブロックから読み出されるデータにそれぞれ付加して 2 つの読み出しデータ列 rdat0、rdat1 を生成するデータ付加回路 26 と、読

み出しデータ列 rdat0、rdat1 についてそれぞれエラーを検出し、これ等のエラー検出結果から最も確からしい一つを真とするエラー訂正回路 12c とを擬似 SRAM に構成した。このため、擬似 SRAM をアクセスするシステムに対してリフレッシュ動作を隠すことができ、同時にリフレッシュブロック以外で発生した 1 ビットエラーを訂正できる。これは、最小ハミング距離が“4”の水平垂直パリティ符号を使用することで可能になる。

リフレッシュ動作を読み出し要求と競合することなく隠すことができるため、読み出しサイクル時間が長くなることを防止できる。また、リフレッシュ要求間隔を延ばすことができ、スタンバイ中の消費電力を削減できる。したがって、A C 特性および D C 特性の両方を SRAM と同等以下にできる擬似 SRAM を構成できる。換言すれば、SRAM と完全に互換性を有する擬似 SRAM をユーザに提供できる。ユーザは、システム上に搭載される SRAM を本発明の擬似 SRAM に置き換えるだけで、コストを削減できる。

リフレッシュ制御回路 20 が出力するリフレッシュポイント信号 refpointd[15:0]、refpointd[7:0]により、データブロック MBD0-15 および符号ブロック MBE0-7 のいずれか 1 つのみをリフレッシュブロックに設定した。リフレッシュ動作により読み出せないデータを常に 1 ビットにできるため、リフレッシュ動作により読み出せないデータを、簡易なデータ付加回路 26 およびエラー訂正回路 12c によって復号できる。

アクセス制御回路 18 は、リフレッシュブロックに対して読み出し要求信号 rdd、rde を出力しないため、リフレッシュブロックは、リフレッシュ動作の実行中、非実行中にかかわらず読み出し動作を実行しない。このため、ブロック制御回路 BCNT の読み出し制御を容易にできる。

ブロック制御回路 BCNT は、リフレッシュ要求信号 refd（または refe）と書き込み要求信号 wrd（または wre）が競合するときに、リフレッシュ動作を実行した後、書き込み動作を遅れて実行する。このため、書き込みデータ wdat（または wecc）をリフレッシュブロックに対して確実に書き込むことができる。

書き込みコマンドの最小供給間隔である外部書き込みサイクル時間 Tcyce を、内部書き込みサイクル時間 Tcycei より長く設定した。具体的には、8 回の外部書

き込みサイクル時間 T_{cyce} の間に、1 回のリフレッシュ動作と 8 回の書き込み動作を実行可能にした。このため、リフレッシュ動作の優先により発生した書き込み動作の遅れを徐々に取り戻すことができる。すなわち、書き込みデータをリフレッシュブロックに確実に書き込むことができる。

- 5 また、リフレッシュブロックの最後のリフレッシュ要求信号から次のリフレッシュブロックに切り替わるまでの切り替え期間に、8 回の外部書き込みサイクル時間 T_{cyce} を挿入可能とした。このため、あるブロックにおいてリフレッシュ要求により遅れた書き込み動作は、そのブロックがリフレッシュブロックの間に完了する。したがって、読み出し動作が、リフレッシュブロック中の書き込み動作
- 10 の遅れの影響を受けて遅れることを防止でき、リフレッシュ動作と書き込み動作の競合に伴い読み出しアクセス時間が増加することを防止できる。

- エラー訂正回路 12c は、4 種類のエラー項目を検出し（エラー無し、データブロックのいずれかの 1 ビットエラー、符号ブロックのいずれかの 1 ビットエラーである符号エラーおよび 2 ビット以上のエラー）、優先度の高いエラー項目を
- 15 真とする。このため、簡易な論理でエラー訂正回路 12c を構成できる。

データブロック MBD0-15 を、データ端子 DQ0-15 にそれぞれ対応して形成することで、擬似 SRAM に一度に入出力されるデータ単位でエラーを訂正できる。このため、符号データの生成および読み出しデータのエラー訂正の論理を簡易にでき、読み出しサイクル時間および書き込みサイクル時間が増加することを防止できる。

- 20 図 12 は、本発明の半導体メモリの第 2 の実施形態を示している。第 1 の実施形態で説明した回路・信号と同一の回路・信号については、同一の符号を付し、これ等については、詳細な説明を省略する。

- この実施形態の半導体メモリは、シリコン基板上に CMOS プロセスを使用してクロック非同期式の擬似 SRAM として形成されている。すなわち、擬似 SRAM は、
- 25 クロック信号を受けるクロック端子を持っていない。また、この実施形態では、後述する図 13、図 14 に示すように、拡大ハミング符号がエラー訂正符号として使用される。この拡大ハミング符号の最小ハミング距離は、“4”であり、1 ビットのエラー訂正と、2 ビットのエラー検出とを同時にできる。

擬似 SRAM は、第 1 の実施形態のデータ制御回路 12、アクセス制御回路 18、

リフレッシュ制御回路 20 およびメモリコア 24 の代わりに、データ制御回路 12 A、アクセス制御回路 18 A、リフレッシュ制御回路 20 A およびメモリコア 24 A を有している。また、擬似 SRAM は、クロック信号 OSC を生成する発振器 40 A を新たに有している。

- 5 メモリコア 24 A は、データ端子 DQ15-0 にそれぞれ対応する 16 個のデータブロック MBD0-15、6 ビットの誤り訂正符号の各ビットをそれぞれ保持する 6 個の符号ブロック MBE0-5 およびブロック制御回路 BCNT を有している。データブロック MBD0-15 および符号ブロック MBE0-5 の構造、メモリ容量は、第 1 の実施形態と同じである。すなわち、各ブロック MBD0-15、MBE0-5 は、ダイナミックメモリセルと、これ等ダイナミックメモリセルに接続されたワード線、ビット線とを有している。

- 15 拡大ハミング符号は、符号化する情報データが 16 ビット（本実施形態）、32 ビット、64 ビット、128 ビット、256 ビットするとき、それぞれ 6 ビット、7 ビット、8 ビット、9 ビット、10 ビット必要になる。すなわち、情報ビットが 2 の n 乗するとき、符号データは $n + 2$ ビット必要になる。このため、拡大ハミング符号は、上述した水平垂直パリティ符号より符号化率が高く、符号ブロック MBE の数を減らすことができる。また、符号ブロック MBE に配線される信号線を減らすことができる。この結果、擬似 SRAM のチップサイズを削減できる。

- 20 発振器 40 A は、パワーオン後、所定の周期のクロック信号 OSC を自己発振する。この実施形態では、クロック信号 OSC の周波数は、第 1 の実施形態のクロック信号 CLK の周波数と同じに設定されている。

- 25 リフレッシュ制御回路 20 A は、リフレッシュポイント信号 $\text{refpointd}[15:0]$ 、 $\text{refpointe}[7:0]$ およびリフレッシュ要求信号 $\text{refd}[15:0]$ 、 $\text{refe}[7:0]$ の代わりに、リフレッシュポイント信号 $\text{refpoint}[21:0]$ およびリフレッシュ要求信号 $\text{ref}[21:0]$ を出力することを除き、第 1 の実施形態のリフレッシュ制御回路 20 と同じである。すなわち、リフレッシュ制御回路 20 A は、クロック信号 OSC に同期して動作するタイマ、カウンタ、シフトレジスタおよびリフレッシュ要求生成回路を有している。シフトレジスタは、縦続接続された 22 個のラッチを有しており、最終段のラッチの出力は初段のラッチの入力に帰還されている。シフトレ

ジスタは、カウンタのオーバフロー毎にシフト動作し、リフレッシュ動作を実行するブロック MBD0-15、MBE0-5 を示すためのリフレッシュポイント信号 refpoint[21:0]をラッチからそれぞれ出力する。リフレッシュポイント信号 refpoint[21:0]により選択されたブロック MBD0-15、MBE0-5 の一つは、リフレッシュ動作を実行するリフレッシュブロックとして動作する。リフレッシュ要求生成回路は、リフレッシュポイント信号 refpoint[21:0]の活性化期間に、周期信号に同期して対応するリフレッシュ要求信号 ref[21:0]を出力する。

リフレッシュ制御回路 20 Aの動作は、信号名が異なることを除き、図 3 および図 4 と同じである。すなわち、各リフレッシュポイント信号 refpoint[21:0]の活性化間隔 T1（各メモリセルのリフレッシュ間隔に対応）は、ワーストのメモリセルのデータリテンション時間より長く設定されている。ワーストのメモリセルは、リフレッシュ動作を実行するときに既にデータを消失する。しかし、失われたデータは、データ制御回路 12 Aのエラー訂正機能により復号できる。この結果、第 1 の実施形態と同様にスタンバイ期間の消費電力を削減できる。

アクセス制御回路 18 Aは、コマンド信号 CMD を解釈し、解釈結果に応じて読み出し動作を実行する読み出し要求信号 rd[21:0]または書き込み要求信号 wr[21:0]を出力する。但し、アクセス制御回路 18 Aは、リフレッシュポイント信号 refpoint[21:0]のいずれかの活性化を受けたとき、対応する読み出し要求信号 rd[21:0]のいずれかの出力をマスクする。また、アクセス制御回路 18 Aは、リフレッシュ要求信号 ref[21:0]のいずれかを受けたとき、対応する書き込み要求信号 wr[21:0]のいずれかを、リフレッシュ動作が挿入される期間だけ順次遅らせて出力する。

データ制御回路 12 Aは、書き込み動作中に、入力データ DIN[15:0]を書き込みデータ wdat[21:0]としてメモリコア 24 に出力する。書き込みデータ wdat[21:0]の上位 6 ビット[21:16]は、入力データ DIN[15:0]から生成されるエラー訂正符号（書き込み符号データ）である。また、データ制御回路 12 は、読み出し動作中に、メモリコア 24 から読み出しデータ rdat[21:0]を受け、リフレッシュ動作により読み出させないデータおよびデータ保持特性が悪くデータが失われたメモリセルからの読み出しデータをエラー訂正し、出力データ DOUT[15:0]

として出力する。読み出しデータ rdat[21:0] の上位 6 ビット[21:16]は、符号ブロック MBE21-16 から読み出されるエラー訂正符号(読み出し符号データ)である。また、データ制御回路 1 2 A は、読み出し動作中に訂正が不可能な 2 ビット以上のエラーを検出したときに、ECC エラー信号 ECCERR をエラー端子に出力する。

- 5 なお、書き込み要求信号 wr[21:0]、読み出し要求信号 rd[21:0]、リフレッシュポイント信号 refpoint[21:0]、リフレッシュ要求信号 ref[21:0]、書き込みデータ wdat[21:0]、読み出しデータ rdat[21:0] のビットは、データブロック MBD15-0 および符号ブロック MBE21-16 にそれぞれ対応する。入力データ DIN[15:0] および出力データ DOUT[15:0] のビットは、データブロック MBD15-0 のブロック番号にそれぞれ対応する。
- 10

図 1 3 は、図 1 2 に示したデータ制御回路 1 2 A の詳細を示している。

- データ制御回路 1 2 A は、メモリア 2 4 A からの読み出しデータを訂正するデータ訂正部 1 2 d およびメモリア 2 4 A への書き込みデータからエラー訂正符号を生成するデータ符号化部 1 2 e を有している。データ訂正部 1 2 d は、データ付加回路 4 2 およびエラー訂正回路 1 2 f を有している。エラー訂正回路 1 2 f は、シンδροーム生成回路 4 4、4 6、エラー判定回路 4 8、5 0 およびエラー復号回路 5 2 を有している。データ符号化部 1 2 e は、符号化回路 5 4 A を有している。
- 15

- データ付加回路 4 2 は、読み出し動作を実行できないリフレッシュブロックとして動作するブロック (MBD0-15、MBE0-5 のいずれか) からの読み出しデータを“論理 1”および“論理 0”と仮定して、読み出しデータ列 code0[21:0]、code1[21:0] をそれぞれ生成する。データ付加回路 4 2 は、リフレッシュポイント信号 refpoint[21:0] によりリフレッシュブロックを判定する。すなわち、データ付加回路 4 2 は、リフレッシュブロックから読み出されるべきデータを“論理 1”または“論理 0”の予想データとして、他のブロックからの読み出しデータに付加し、読み出しデータ列 code0[21:0]、code1[21:0] を生成する。
- 20
- 25

シンδροーム生成回路 4 4 は、読み出しデータ code0[21:0] からシンδροーム syn0[5:0] を生成する。また、シンδροーム生成回路 4 4 は、生成したシンδροーム syn0[5:0] からエラーの種別を判定し、比較結果信号 res0 として出力する。シ

シンドローム生成回路 46 は、読み出しデータ `code1[21:0]` からシンドローム `syn1[5:0]` を生成する。また、シンドローム生成回路 46 は、生成したシンドローム `syn1[5:0]` からエラーの種別を判定し、比較結果信号 `res1` として出力する。比較結果信号 `res0`、`res1` の仕様は、第 1 の実施形態とほぼ同じである。但し、本実施形態では、「ECC エラー」はない。このため、エラー判定の優先度は、高い順からエラー無し > 1 ビットエラー > 2 ビット以上のエラーになる。

エラー判定回路 48 は、シンドローム `syn0[5:0]` からエラーの種別およびエラーブロックを判定し、読み出しデータ `code0[21:0]` のエラーを訂正する。訂正された 22 ビットの読み出しデータは、訂正コード `ccode0[21:0]` として出力される。

10 訂正された 22 ビットの読み出しデータのうちメモリブロック MBD15-0 に対応する下位の 16 ビットは、出力データ `dout0[15:0]` として出力される。また、エラー判定回路 48 は、2 ビット以上のエラーを検出したときに、エラーフラグ信号 `errflg_2b0` を出力する。

エラー判定回路 50 は、エラー判定回路 48 と同じ回路である。エラー判定回路 50 は、シンドローム `syn1[5:0]` からエラーの種別およびエラーブロックを判定し、読み出しデータ `code1[21:0]` のエラーを訂正し、訂正コード `ccode1[21:0]`、出力データ `dout1[15:0]`、エラーフラグ信号 `errflg_2b1` を出力する。

エラー復号回路 52 は、比較結果信号 `res0`、`res1` が示すエラー内容に応じて、エラー判定回路 48、50 のどちらが真かを判定し、判定したエラー判定回路 48 (または 50) から出力される出力データ `dout0[15:0]` (または `dout1[15:0]`)、エラーフラグ信号 `errflg_2b0` (または `errflg_2b1`) および訂正コード `ccode0[21:0]` (または `ccode1[21:0]`) を、出力データ `DOUT[15:0]`、ECC エラー信号 `ECCERR` および訂正コード `ccode[21:0]` として出力する。出力データ `DOUT[15:0]` および ECC エラー信号 `ECCERR` は、データ端子 `DQ15-0` およびエラー端子 `ECCERR` を介して擬似 SRAM の外部に出力される。訂正コード `ccode[21:0]` は、この実施形態では使用されないが、エラーの発生したメモリセルのデータを書き換える場合に使用される書き換えデータである。

データ符号化部 12e の符号化回路 54A は、入力データ `DIN[15:0]` (書き込みデータ) からエラー訂正符号 `wdat[21:16]` を生成し、書き込みデータ `wdat[15:0]`

とともにメモリコア 24 に出力する。

図 1 4 ~ 図 1 7 は、図 1 3 に示したデータ制御回路 1 2 A の一部の回路ブロックの RTL 記述および論理合成結果を示している。記述中の "syndrome" は、図 1 3 に示した "syn" に対応する。

5 図 1 4 は、図 1 3 に示したエラー判定回路 4 8、5 0 の RTL 記述を示している。図 1 5 は、図 1 4 の RTL 記述から論理合成した結果を示している。図 1 6 は、図 1 3 に示したシンドローム生成回路 4 4、4 6 のシンドローム生成部の RTL 記述を示している。図 1 7 は、図 1 3 に示したデータ符号化部 1 2 e の符号化回路 5 4 の RTL 記述を示している。

10 図 1 8 は、図 1 2 に示したデータ制御回路 1 2 A のエラー訂正動作を示している。

図中のブロック D0-D15 およびブロック P0-P5 は、図 1 2 に示したデータブロック MBD0-15 および符号ブロック MBE16-21 にそれぞれ対応する。ブロック D0-D15、P0-P5 のビットアドレス E1-E22 は、一般的なハミング符号と同様に、シンドロームのビットがエラーアドレスを示すように割り当てられている。

この例では、データブロック MBD10 (ブロック D10) がリフレッシュブロックとして動作している。このため、読み出しデータ rdat のうちブロック D10 に対応するビットは、網掛けの太枠で囲ったように不定である。図 1 3 に示したデータ制御回路 1 2 A のデータ付加回路 4 2 は、ブロック D10 に対応するビットに "論理 0" を付加した読み出しデータ code0 と、ブロック D10 に対応するビットに "論理 1" を付加した読み出しデータ code1 とを生成する。

シンドローム生成回路 4 4 は、読み出しデータ code0 からシンドローム syn0[5:0] を生成する。シンドローム生成回路 4 4 は、シンドローム syn0[5:0] の全ビット S5-0 が "論理 0" のため、エラー無しを示す比較結果信号 res0 ("1000") を出力する。シンドローム生成回路 4 6 は、読み出しデータ code1 からシンドローム syn1[5:0] を生成する。シンドローム生成回路 4 6 は、シンドローム syn1[5:0] の最上位ビット S5 が "論理 1" のため、1 ビットエラーを示す比較結果信号 res1 ("0010") を出力する。

エラー判定回路 4 8 は、シンドローム生成回路 4 4 からのシンドローム

syn0[5:0]がエラー無しを示すため、読み出しデータ code0[21:0]のうちデータブロック MBD15-0 に対応するビットを出力データ dout0[15:0]として出力する。エラー判定回路 5 0 は、シンドローム生成回路 4 4 からのシンドローム syn1[5:0]が 1 ビットエラーを示すため、シンドローム syn1[5:0]の下位 5 ビット (“01111”
 5 = 1 0 進の “1 5”) をエラーアドレス (“E15”=MBD10) とする。エラー判定回路 5 0 は、読み出しデータ code1[21:0]のうちデータブロック MBD10 から読み出されたデータの論理を反転し、データブロック MBD15-0 に対応するビットを出力データ dout1[15:0]として出力する。

エラー復号回路 5 2 は、シンドローム生成回路 4 4、4 6 からの比較結果信号
 10 res0 (エラー無し)、res1 (1 ビットエラー) より、エラー判定回路 4 8 の出力が正しいと判定する。エラー復号回路 5 2 は、エラー判定回路 4 8 から出力される出力データ dout0[15:0] およびエラーフラグ信号 errflg_2b0 (エラー無しを示す) を出力データ DOUT[15:0]および ECC エラー信号 ECCERR (エラー無しを示す) として出力する。

15 図 1 9 は、図 1 2 に示したデータ制御回路 1 2 A の別のエラー訂正動作を示している。図 1 8 と同じ記述および動作については詳細な説明を省略する。

この例では、図 1 8 と同様に、データブロック MBD10 (ブロック D10) がリフレッシュブロックとして動作している。また、図中に太枠で示すように、データブロック MBD1 (ブロック D1) に 1 ビットエラーが発生する。

20 データ制御回路 1 2 A のデータ付加回路 4 2 は、ブロック D10 に対応するビットに “論理 0” を付加した読み出しデータ code0 と、ブロック D10 に対応するビットに “論理 1” を付加した読み出しデータ code1 とを生成する。

シンドローム生成回路 4 4、4 6 は、読み出しデータ code0、code1 からシンドローム syn0[5:0]、syn1[5:0]をそれぞれ生成する。シンドローム生成回路 4 4 は、
 25 シンドローム syn0[5:0]の最上位ビット S5 が “論理 1” のため、1 ビットエラーを示す比較結果信号 res0 (“0010”) を出力する。シンドローム生成回路 4 6 は、シンドローム syn1[5:0]の最上位ビット S5 が “論理 0” かつビット S4-0 の少なくともいずれかが “論理 1” のため、2 ビット以上のエラーを示す比較結果信号 res1 (“0001”) を出力する。

エラー判定回路 4 8 は、シンドローム生成回路 4 4 からのシンドローム syn0[5:0]が 1 ビットエラーを示すため、シンドローム syn0[5:0]の下位 5 ビット ("00101" = 1 0 進の "5") をエラーアドレス ("E5" = MBD1) とする。エラー判定回路 4 8 は、データブロック MBD1 から読み出された "論理 1" を反転し、読み出し
5 データ code0[21:0]のうちデータブロック MBD15-0 に対応するビットを出力データ dout0[15:0]として出力する。エラー判定回路 5 0 は、シンドローム生成回路 4 4 からのシンドローム syn1[5:0]が 2 ビット以上のエラーを示すため、エラー訂正を行わず、読み出しデータ code1[21:0]のうちデータブロック MBD15-0 に対応するビットをそのまま出力データ dout1[15:0]として出力する。また、エラー
10 判定回路 5 0 は、エラーフラグ信号 errflg_2b1 を高レベルに活性化する。

エラー復号回路 5 2 は、シンドローム生成回路 4 4、4 6 からの比較結果信号 res0 (1 ビットエラー)、res1 (2 ビット以上のエラー) より、エラー判定回路 4 8 の出力が正しいと判定する。エラー復号回路 5 2 は、エラー判定回路 4 8 から出力される出力データ dout0[15:0]およびエラーフラグ信号 errflg_2b0 (エラー無しを示す) を出力データ DOUT[15:0]および ECC エラー信号 ECCERR (エラー無しを示す) として出力する。
15

この実施形態においても、上述した第 1 の実施形態と同様の効果を得ることができる。さらに、この実施形態では、符号データとして拡大ハミング符号を使用することで、符号データのビット数を減らすことができる。この結果、符号ブロック MBE の数を第 1 の実施形態に比べ 2 つ減らすことができ、擬似 SRAM のチップ
20 サイズを削減できる。

また、エラー訂正回路 1 2 f は、訂正不可能な「2 ビット以上のエラー」を検出したときに、ECC エラー信号 ECCERR を出力する。ECC エラー信号 ECCERR により、擬似 SRAM をアクセスする外部のシステムは、読み出し動作のリトライおよびデータの廃棄等、エラーを回避する動作を実行できる。この結果、システムの信頼性を向上できる。
25

図 2 0 は、本発明の半導体メモリの第 3 の実施形態を示している。第 1 および第 2 の実施形態で説明した回路・信号と同一の回路・信号については、同一の符号を付し、これ等については、詳細な説明を省略する。

この実施形態の半導体メモリは、シリコン基板上に CMOS プロセスを使用してクロック非同期式の擬似 SRAM として形成されている。また、この実施形態では、拡大ハミング符号がエラー訂正符号として使用される。この拡大ハミング符号の最小ハミング距離は、“4”であり、1 ビットのエラー訂正と、2 ビットのエラー検出とを同時にできる。

擬似 SRAM は、第 2 の実施形態のメモリコア 24 A および発振器 40 A の代わりに、メモリコア 24 B および発振器 40 B を有している。その他の構成は、第 2 の実施形態とほぼ同一である。

メモリコア 24 B は、ワード線単位で不良を救済するためのロウ冗長回路およびビット線単位で不良を救済するためのコラム冗長回路が形成されていないことを除き、第 2 の実施形態のメモリコア 24 A と同じである。

発振器 40 B は、第 2 の実施形態より周期の短いクロック信号 OSC を自己発振する。このため、リフレッシュ制御回路は 20 A が出力する各リフレッシュポイント信号 `refpoint[21:0]` の活性化期間およびリフレッシュ要求信号 `ref[21:0]` の発生間隔は、第 2 の実施形態より短くなる。具体的には、各リフレッシュポイント信号 `refpoint[21:0]` の活性化間隔 T1 (各メモリセルのリフレッシュ間隔に対応) は、ワーストのメモリセルのデータリテンション時間より短く設定されている。このため、ワーストのメモリセルを含む全てのメモリセルは、リフレッシュ動作の実行により常にデータを保持できる。データ制御回路 12 A のエラー訂正機能は、製造工程で発生したメモリコアの不良を救済するために使用される。すなわち、符号ブロック MBE0-5 は、ロウ冗長回路およびコラム冗長回路の代わりに機能する。

この実施形態においても、上述した第 1 および第 2 の実施形態と同様の効果を得ることができる。さらに、この実施形態では、符号ブロック MBE0-5 を、ロウ冗長回路およびコラム冗長回路の代わりに形成することで、製造工程で不良になったメモリセルから読み出される誤ったデータを、エラー訂正回路により訂正できる。冗長回路の形成が不要になるため、チップサイズを削減できる。

図 21 は、本発明の半導体メモリの第 4 の実施形態を示している。第 1 および第 2 の実施形態で説明した回路・信号と同一の回路・信号については、同一の符

号を付し、これ等については、詳細な説明を省略する。

この実施形態の半導体メモリは、シリコン基板上に CMOS プロセスを使用してクロック非同期式の擬似 SRAM として形成されている。また、この実施形態では、拡大ハミング符号がエラー訂正符号として使用される。この拡大ハミング符号の最小ハミング距離は、“4”であり、1 ビットのエラー訂正と、2 ビットのエラー検出とを同時にできる。

擬似 SRAM は、第 2 の実施形態のデータ制御回路 1 2 A、アクセス制御回路 1 8 A および発振器 4 0 A の代わりに、データ制御回路 1 2 C、アクセス制御回路 1 8 C および第 3 の実施形態と同じ発振器 4 0 B を有している。

10 データ制御回路 1 2 C は、読み出し動作中に、訂正可能なエラー（1 ビットエラー）を検出したときに、訂正したデータをメモリコア 2 4 A に書き戻す機能を有している。アクセス制御回路 1 8 C は、データ制御回路 1 2 C が訂正したデータをメモリコア 2 4 A に書き戻すときに、書き込み要求信号 wr[21:0] を出力する。

図 2 2 は、図 2 1 に示したデータ制御回路 1 2 C の詳細を示している。

15 データ制御回路 1 2 C は、第 2 の実施形態のデータ符号化部 1 2 e の代わりにデータ符号化部 1 2 g を有している。データ訂正部 1 2 d の構成は、第 2 の実施形態と同じである。データ符号化部 1 2 g は、符号化回路 5 4 A およびセクタ 5 6 B を有している。セクタ 5 6 B は、書き込み動作中に、入力データ DIN[15:0]（書き込みデータ）を符号化回路 5 4 A で符号化したデータを書き込み
20 データ wdat[21:0] として出力する。また、セクタ 5 6 B は、読み出し動作中に 1 ビットエラーが発生したときに、データ訂正部 1 2 d のエラー復号回路 5 2 から出力される訂正コード ccode[21:0] を書き込みデータ wdat[21:0] として出力する。すなわち、セクタ 5 6 B は、読み出し動作中に、エラー訂正回路 1 2 f が訂正したデータをデータブロック MBD および符号ブロック MBE に書き戻す書き戻し制御回路として機能する。
25

この実施形態においても、上述した第 1 および第 2 の実施形態と同様の効果を得ることができる。さらに、この実施形態では、エラー訂正回路 1 2 f が訂正したデータをセクタ 5 6 B によってデータブロック MBD および符号ブロック MBE に書き戻すことができる。このため、電源ノイズ、ソフトエラー等により破壊さ

れたメモリセル内のデータが読み出されたとき、このメモリセルに正しいデータを書き込むことができ、擬似 SRAM の信頼性を向上できる。

図 2 3 は、本発明の半導体メモリの第 5 の実施形態を示している。第 1 および第 2 の実施形態で説明した回路・信号と同一の回路・信号については、同一の符号を付し、これ等については、詳細な説明を省略する。

この実施形態の半導体メモリは、シリコン基板上に CMOS プロセスを使用してクロック非同期式の擬似 SRAM として形成されている。また、この実施形態では、拡大ハミング符号がエラー訂正符号として使用される。この拡大ハミング符号の最小ハミング距離は、“4”であり、1 ビットのエラー訂正と、2 ビットのエラー検出とを同時にできる。

擬似 SRAM は、第 2 の実施形態のデータ制御回路 1 2 A、アクセス制御回路 1 8 A、リフレッシュ制御回路 2 0 A、発振器 4 0 A およびメモリコア 2 4 A の代わりに、データ制御回路 1 2 D、アクセス制御回路 1 8 D、リフレッシュ制御回路 2 0 D、発振器 4 0 D およびメモリコア 2 4 D を有している。

メモリコア 2 4 D は、6 4 個のデータブロック MBD[63:0]と、8 個の符号ブロック MBE[7:0]を有している。この実施形態では、データブロック MBD[63:0]の数は、データ端子 DQ[15:0]のビット数の 4 倍に設計されている。このため、1 回の書き込み動作でデータ端子[15:0]に供給される書き込みデータだけでは、符号データを生成できない。したがって、データ制御回路 1 2 D は、後述するように、書き込み動作の最初に、データブロック MBD[63:0]および符号ブロック MBE[7:0]からデータを読み出す必要がある。しかし、符号データを生成するための情報データのビット数を増やすことで、符号ブロック MBE の数を相対的に減らすことができる。特に、拡大ハミング符号の場合、情報ビットが 2 の n 乗のとき、符号データは $n + 2$ ビットだけでよい。この結果、符号ブロック MBE の数を相対的に大幅に減らすことができ、擬似 SRAM のチップサイズが増加することを防止できる。

データ制御回路 1 2 D は、読み出し動作中に、読み出しデータ rdat[71:0]に訂正可能なエラー（1 ビットエラー）がある場合、エラーを訂正する。そして、データ制御回路 1 2 D は、アドレス端子 AD に供給されるロウアドレスの下位ビットに応じてデータブロック MBD0-63 のうちの 1 6 個に対応する読み出しデータを選

択し、選択したデータを出力データ DOUT[15:0]として出力する。データ制御回路 1 2 Dは、読み出し動作中に、読み出しデータ rdat[71:0]に訂正不可能なエラー（2 ビット以上のエラー）がある場合、ECC エラー信号 ECCERR をエラー端子に出力する。

- 5 データ制御回路 1 2 Dは、書き込み動作中に、データブロック MBD0-63 および符号ブロック MBE0-7 からデータを読み出し、エラーがある場合に情報データを復号する。次に、データ制御回路 1 2 Dは、復号した情報データのうち書き込みデータ DIN[15:0]に対応するビットを、この書き込みデータ DIN[15:0]に入れ替えて書き換えデータ列を生成する。そして、データ制御回路 1 2 Dは、書き換えデータ列から新たな符号データを求め、書き換えデータ列および新たな符号データをデータブロック MBD0-63 および符号ブロック MBE0-7 にそれぞれ書き込む。このように、本実施形態では、擬似 SRAM は、書き込み動作の最初にメモリコア 2 4 Aからデータを読み出す読み出し動作が実行される。

- 15 アクセス制御回路 1 8 Dは、データ制御回路 1 2 Dの動作に合わせて、読み出し要求信号 rd[71:0]または書き込み要求信号 wr[71:0]を出力する。具体的には、書き込み動作中に、読み出し要求信号 rd[71:0]および書き込み要求信号 wr[71:0]が順次出力される。

- 20 リフレッシュ制御回路 2 0 Dは、リフレッシュポイント信号 refpoint[71:0]およびリフレッシュ要求信号 ref[71:0]を順次出力する。リフレッシュ制御回路 2 0 Dの動作は、出力する信号のビット数が増えたことを除き、第 2 の実施形態のリフレッシュ制御回路 2 0 Aと同じである。発振器 4 0 Dが出力するクロック信号 OSC の周波数は、リフレッシュポイント信号 refpoint[71:0]のビット数に対応して、第 2 の実施形態に比べて高く設定されている。

図 2 4 は、図 2 3 に示した擬似 SRAM の書き込み動作を示している。

- 25 書き込み動作では、まず、図示しないワード線が活性化され、符号ブロック MBE およびデータブロック MBD のそれぞれにおいて、メモリセルから相補のビット線 BL、/BL のいずれかにデータが読み出される（図 2 4（a））。この後、センスアンプが活性化され、ビット線 BL、/BL の電圧差が増幅される（図 2 4（b））。次に、コラムスイッチ信号 CSW が活性化され、増幅されたデータは、読み出しデ

ータ rdat[63:0]として、データ制御回路 1 2 D に転送される (図 2 4 (c))。データ制御回路 1 2 D は、読み出しデータ rdat[63:0] のエラーを訂正する。次に、データ制御回路 1 2 D は、データブロック MBD から読み出されたデータのうち 8 ビットを、データ端子 DQ7-0 に供給された書き込みデータ DIN[7:0] と置き換える (図 2 4 (d))。

次に、データ制御回路 1 2 D は、置き換えた書き込みデータを用いて新たな符号データを生成する (図 2 4 (e))。そして、置き換えた書き込みデータおよび新たな符号データは、書き込みデータ wdat[63:0] として符号ブロック MBE およびデータブロック MBD に順次書き込まれる (図 2 4 (f、g))。この後、ワード線が非活性化され、ビット線 BL、/BL がプリチャージされて、書き込み動作が完了する (図 2 4 (h、i))。

この例では、データの書き込み時のコラムスイッチ信号 CSW のタイミング、ワード線の非活性化タイミングおよびビット線 BL、/BL のプリチャージ開始タイミングは、符号ブロック MBE およびデータブロック MBD で相違している。しかし、これ等のタイミングは、符号ブロック MBE およびデータブロック MBD で同じにしてもよい。

この実施形態においても、上述した第 1 および第 2 の実施形態と同様の効果を得ることができる。さらに、この実施形態では、情報データを記憶するデータブロックの数を増やすことで、情報データのビット数に対する符号データのビット数を相対的に減らすことができる。この結果、符号ブロックの数を、データブロックの数に対して相対的に減らすことができ、擬似 SRAM のチップサイズを削減できる。

なお、上述した実施形態では、本発明を擬似 SRAM チップに適用した例について述べた。本発明はかかる実施形態に限定されるものではない。例えば、本発明をシステム L S I に搭載される擬似 SRAM コアに適用してもよい。

以上、本発明について詳細に説明してきたが、上記の実施形態およびその変形例は発明の一例に過ぎず、本発明はこれに限定されるものではない。本発明を逸脱しない範囲で変形可能であることは明らかである。

産業上の利用の可能性

本発明の半導体メモリでは、読み出しサイクル時間が長くすることなくリフレッシュ動作を外部のシステムに対して隠すことができ、同時に読み出しデータ列のエラーを訂正できる。エラー訂正機能を、データリテンション時間が短いメモリセルの読み出し不良の救済に利用することで、リフレッシュ要求間隔を延ばすことができ、スタンバイ中の消費電力を削減できる。したがって、本発明を、例えば、擬似 SRAM に適用した場合、AC 特性および DC 特性の両方を SRAM と同等以下にできる。換言すれば、SRAM と完全に互換性を有する擬似 SRAM をユーザに提供できる。ユーザは、システム上に搭載される SRAM を本発明が適用された擬似 SRAM に置き換えるだけで、コストを削減できる。

本発明の半導体メモリでは、リフレッシュ要求に応答するリフレッシュ動作を 1 つのブロックでのみ実行することで、リフレッシュ動作により読み出せないデータを、容易に復元できる。

本発明の半導体メモリでは、リフレッシュ要求と読み出し要求が競合することを防止でき、読み出しアクセス時間が増加することを防止できる。また、アクセス制御回路による読み出し動作の制御を容易にできる。

本発明の半導体メモリでは、リフレッシュ要求信号と書き込み要求信号が競合するときに、リフレッシュ動作を実行した後に書き込み動作を遅れて実行することで、書き込みデータをリフレッシュブロックに対して確実に書き込むことができる。

本発明の半導体メモリでは、連続して供給される書き込み要求信号の間にリフレッシュ要求が発生した場合にも、書き込み動作の遅れを徐々に取り戻すことができる。

本発明の半導体メモリでは、リフレッシュブロック以外のブロックにおいて、リフレッシュ動作と書き込み動作の競合に伴って読み出し動作が遅れることを防止でき、読み出しアクセス時間が長くなることを防止できる。

本発明の半導体メモリでは、符号ブロックを冗長回路の代わりに形成することで、冗長回路を不要にでき、チップサイズを削減できる。

本発明の半導体メモリでは、読み出しデータ列のうちエラーの少ない読み出し

データ列を真として選択することで、エラー訂正回路のエラー判定論理を簡易に構成できる。

- 本発明の半導体メモリでは、エラー信号により、半導体メモリをアクセスする外部のシステムは、読み出し動作のリトライ要求およびデータの廃棄等、エラーを回避する動作を実行できる。この結果、システムの信頼性を向上できる。

本発明の半導体メモリでは、メモリセル内のデータが電源ノイズ、ソフトエラー等により破壊されたときに、書き戻し制御回路により正しいデータを書き込むことで、半導体メモリの信頼性を向上できる。

- 本発明の半導体メモリでは、データブロックを複数のデータ端子にそれぞれ対応して形成することで、半導体メモリに一度に入出力されるデータ単位でエラーを訂正でき、符号データの生成および読み出しデータのエラー訂正の論理を簡易にできる。この結果、読み出しサイクル時間および書き込みサイクル時間が増加することを防止できる。

- 本発明の半導体メモリでは、情報データのビット数を増やすことで、情報データのビット数に対する符号データのビット数を相対的に減らすことができる。この結果、符号ブロックの数を、データブロックの数に対して相対的に減らすことができ、チップサイズを削減できる。

請求の範囲

(1) 情報データを記憶するメモリセルを有する複数のデータブロックと、

前記データブロックに記憶されたデータを復号するための符号データを記憶するメモリセルを有する複数の符号ブロックと、

前記データブロックおよび前記符号ブロックにリフレッシュ動作を順次実行させるためにリフレッシュ要求信号を出力するリフレッシュ制御回路と、

前記データブロックおよび前記符号ブロックのうちリフレッシュ動作によりデータを読み出せないブロックであるリフレッシュブロックから読み出されるべき複数種の予想データを、他のブロックから読み出されるデータにそれぞれ付加して複数の読み出しデータ列を生成するデータ付加回路と、

前記読み出しデータ列についてそれぞれエラーを検出し、これ等のエラー検出結果から最も確からしい一つを真とし、真としたエラー検出結果に対応する読み出しデータ列のエラーを訂正するエラー訂正回路とを備えていることを特徴とする半導体メモリ。

(2) 請求の範囲 1 の半導体メモリにおいて、

前記リフレッシュ制御回路は、前記データブロックおよび前記符号ブロックのいずれか 1 つにリフレッシュ動作を実行させるために、前記データブロックおよび前記符号ブロックにリフレッシュ要求信号を順次出力することを特徴とする半導体メモリ。

(3) 請求の範囲 1 の半導体メモリにおいて、

読み出しコマンドに応答してリフレッシュブロックを除く前記データブロックおよび前記符号ブロックに読み出し要求信号を出力するアクセス制御回路を備えていることを特徴とする半導体メモリ。

(4) 請求の範囲 1 の半導体メモリにおいて、

書き込みコマンドに応答して前記データブロックおよび前記符号ブロックに書き込み要求信号を出力するアクセス制御回路を備え、

前記データブロックおよび前記符号ブロックは、前記リフレッシュ要求信号と前記書き込み要求信号が競合するときに、リフレッシュ動作を実行した後書き込

み動作を順次遅れて実行するブロック制御回路をそれぞれ備えていることを特徴とする半導体メモリ。

(5) 請求の範囲4の半導体メモリにおいて、

- 前記書き込みコマンドの最小供給間隔である外部書き込みサイクル時間は、前記各ブロックの書き込み動作時間である内部書き込みサイクル時間より長く設定されていることを特徴とする半導体メモリ。

(6) 請求の範囲5の半導体メモリにおいて、

- あるリフレッシュブロックの最後のリフレッシュ要求信号から次のリフレッシュブロックに切り替わるまでの切り替え期間に、 n 回の前記外部書き込みサイクル時間が挿入可能であり、かつ1回のリフレッシュ動作と n 回の書き込み動作を実行可能であることを特徴とする半導体メモリ。

(7) 請求の範囲6の半導体メモリにおいて、

- 前記切り替え期間は、外部書き込みサイクル時間 T_{cyc} と内部書き込みサイクル時間との差を δ とするときに、少なくとも $(T_{cyc}/\delta) \times T_{cyc}$ に設定されることを特徴とする半導体メモリ。

(8) 請求の範囲1の半導体メモリにおいて、

前記符号ブロックは、不良を救済する冗長回路の代わりに形成されることを特徴とする半導体メモリ。

(9) 請求の範囲8の半導体メモリにおいて、

- 前記符号ブロックは、ワード線単位で不良を救済するロウ冗長回路の代わりに形成されることを特徴とする半導体メモリ。

(10) 請求の範囲8の半導体メモリにおいて、

前記符号ブロックは、ビット線単位で不良を救済するコラム冗長回路の代わりに形成されることを特徴とする半導体メモリ。

- (11) 請求の範囲1の半導体メモリにおいて、

前記データ付加回路は、リフレッシュブロックからの読み出しデータの論理を論理1および論理0と仮定して2つの前記読み出しデータ列を生成することを特徴とする半導体メモリ。

(12) 請求の範囲11の半導体メモリにおいて、

前記エラー訂正回路は、前記読み出しデータ列のうちエラーの少ない読み出しデータ列を真として選択することを特徴とする半導体メモリ。

(13) 請求の範囲12の半導体メモリにおいて、

5 前記エラー訂正回路は、「エラー無し」、「データブロックのいずれかの1ビットエラー」、「符号ブロックのいずれかの1ビットエラーである符号エラー」および「2ビット以上のエラー」のいずれかを検出し、エラー判定の優先度を、エラー無し>符号エラー>1ビットエラー>2ビット以上のエラーとし、優先度の高い項目を真とすることを特徴とする半導体メモリ。

(14) 請求の範囲13の半導体メモリにおいて、

10 前記エラー訂正回路が「2ビット以上のエラー」を検出したときに、エラー信号を出力するエラー端子を備えていることを特徴とする半導体メモリ。

(15) 請求の範囲12の半導体メモリにおいて、

15 前記エラー訂正回路は、「エラー無し」、「データブロックまたは符号ブロックのいずれかの1ビットエラー」、および「2ビット以上のエラー」のいずれかを検出し、エラー判定の優先度を、エラー無し>1ビットエラー>2ビット以上のエラーとし、優先度の高い項目を真とすることを特徴とする半導体メモリ。

(16) 請求の範囲1の半導体メモリにおいて、

前記エラー訂正回路が訂正不可能なエラーを検出したときに、エラー信号を出力するエラー端子を備えていることを特徴とする半導体メモリ。

20 (17) 請求の範囲1の半導体メモリにおいて、

前記エラー訂正回路が訂正したデータを前記データブロックおよび前記符号ブロックに書き戻す書き戻し制御回路を備えていることを特徴とする半導体メモリ。

(18) 請求の範囲1の半導体メモリにおいて、

前記情報データを入出力する複数のデータ端子を備え、

25 前記データブロックは、前記データ端子にそれぞれ対応して形成されていることを特徴とする半導体メモリ。

(19) 請求の範囲1の半導体メモリにおいて、

前記情報データを入出力する複数のデータ端子を備え、

前記データ端子に一度に供給される書き込みデータは、前記データブロックの

一部に書き込まれ、

前記エラー訂正回路は、書き込み動作中に、前記データブロックおよび前記符号ブロックからのデータにより前記情報データを復号し、復号した情報データの一部を前記書き込みデータに入れ替えて書き換えデータ列を生成し、前記書き換えデータ列から新たな符号データを求め、前記書き換えデータ列および新たな符号データを前記データブロックおよび前記符号ブロックにそれぞれ書き込むことを特徴とする半導体メモリ。

(20) 請求の範囲1の半導体メモリにおいて、

前記各メモリセルのリフレッシュ間隔は、ワーストのメモリセルのデータリテンション時間より長く設定されていることを特徴とする半導体メモリ。

(21) 請求の範囲1の半導体メモリにおいて、

前記エラー訂正回路は、

論理0の予想データを付加した情報データ列から第1符号データを生成する第1符号化回路と、

論理1の予想データを付加した情報データ列から第2符号データを生成する第2符号化回路と、

前記第1符号データを前記符号ブロックから読み出される符号データと比較し、エラーを判定する第1エラー判定回路と、

前記第2符号データを前記符号ブロックから読み出される符号データと比較し、エラーを判定する第2エラー判定回路と、

前記第1および第2エラー判定回路の判定結果のうち、最も確からしい判定結果を真とし、真とした判定結果に基づいてエラーを訂正するエラー復号回路とを備えていることを特徴とする半導体メモリ。

(22) 請求の範囲1の半導体メモリにおいて、

前記エラー訂正回路は、

前記データブロックから読み出される情報データ列から符号データを生成する符号化回路と、

生成した符号データを、論理0の予想データを付加した符号データ列と比較し、エラーを判定する第1エラー判定回路と、

生成した符号データを、論理 1 の予想データを付加した符号データ列と比較し、エラーを判定する第 2 エラー判定回路と、

前記第 1 および第 2 エラー判定回路の判定結果のうち、最も確からしい判定結果を真とし、真とした判定結果に基づいてエラーを訂正するエラー復号回路とを
5 備えていることを特徴とする半導体メモリ。

(2 3) 請求の範囲 1 の半導体メモリにおいて、

前記エラー訂正回路は、

論理 0 の予想データを付加した読み出しデータ列から第 1 シンドロームを生成する第 1 シンドローム生成回路と、

10 論理 1 の予想データを付加した読み出しデータ列から第 2 シンドロームを生成する第 2 シンドローム生成回路と、

前記第 1 シンドロームからエラーの有無およびエラーアドレスを検出する第 1 エラー判定回路と、

前記第 2 シンドロームからエラーの有無およびエラーアドレスを検出する第 2
15 エラー判定回路と、

前記第 1 および第 2 エラー判定回路の判定結果のうち、最も確からしい判定結果を真とし、真とした判定結果に基づいてエラーを訂正するエラー復号回路とを備えていることを特徴とする半導体メモリ。

(2 4) 請求の範囲 1 の半導体メモリにおいて、

20 前記符号データの最小ハミング距離は、4 以上であることを特徴とする半導体メモリ。

(2 5) 請求の範囲 2 4 の半導体メモリにおいて、

前記符号データは、水平垂直パリティ符号であることを特徴とする半導体メモリ。

25 (2 6) 請求の範囲 2 4 の半導体メモリにおいて、

前記符号データは、拡大ハミング符号であることを特徴とする半導体メモリ。

(2 7) 情報データを記憶するメモリセルを有する複数のデータブロックと、前記データブロックに記憶されたデータを復号するための符号データを記憶するメモリセルを有する複数の符号ブロックと、前記データブロックおよび前記符号ブ

ロックにリフレッシュ動作を順次実行させるためにリフレッシュ要求信号を出力するリフレッシュ制御回路とを備えた半導体メモリの動作方法であって、

- 前記データブロックおよび前記符号ブロックのうちリフレッシュ動作によりデータを読み出せないブロックであるリフレッシュブロックから読み出されるべき
- 5 複数種の予想データを、他のブロックから読み出されるデータにそれぞれ付加して複数の読み出しデータ列を生成し、

前記読み出しデータ列についてそれぞれエラーを検出し、これ等のエラー検出結果から最も確からしい一つを真とし、

- 真としたエラー検出結果に対応する読み出しデータ列のエラーを訂正すること
- 10 を特徴とする半導体メモリの動作方法。

(28) 請求の範囲27の半導体メモリの動作方法において、
前記データブロックおよび前記符号ブロックのいずれか1つにリフレッシュ動作を実行させるために、前記データブロックおよび前記符号ブロックにリフレッシュ要求信号を順次出力することを特徴とする半導体メモリの動作方法。

- 15 (29) 請求の範囲27の半導体メモリの動作方法において、
読み出しコマンドに応答してリフレッシュブロックを除く前記データブロックおよび前記符号ブロックに読み出し要求信号を出力することを特徴とする半導体メモリの動作方法。

- (30) 請求の範囲27の半導体メモリの動作方法において、
- 20 書き込みコマンドに応答して前記データブロックおよび前記符号ブロックに書き込み要求信号を出力し、

前記データブロックおよび前記符号ブロックは、前記リフレッシュ要求信号と前記書き込み要求信号が競合するときに、リフレッシュ動作を実行した後書き込み動作を順次遅れて実行することを特徴とする半導体メモリの動作方法。

- 25 (31) 請求の範囲30の半導体メモリの動作方法において、
前記書き込みコマンドの最小供給間隔である外部書き込みサイクル時間は、前記各ブロックの書き込み動作時間である内部書き込みサイクル時間より長く設定されていることを特徴とする半導体メモリの動作方法。

(32) 請求の範囲31の半導体メモリの動作方法において、

あるリフレッシュブロックの最後のリフレッシュ要求信号から次のリフレッシュブロックに切り替わるまでの切り替え期間に、 n 回の前記外部書き込みサイクル時間が挿入可能であり、かつ1回のリフレッシュ動作と n 回の書き込み動作を実行可能であることを特徴とする半導体メモリの動作方法。

5 (33) 請求の範囲32の半導体メモリの動作方法において、

前記切り替え期間は、外部書き込みサイクル時間 T_{cyc} と内部書き込みサイクル時間との差を δ とするときに、少なくとも $(T_{cyc}/\delta) \times T_{cyc}$ に設定されることを特徴とする半導体メモリの動作方法。

(34) 請求の範囲27の半導体メモリの動作方法において、

10 リフレッシュブロックからの読み出しデータの論理を論理1および論理0と仮定して2つの前記読み出しデータ列を生成することを特徴とする半導体メモリの動作方法。

(35) 請求の範囲34の半導体メモリの動作方法において、

15 前記読み出しデータ列のうちエラーの少ない読み出しデータ列を真として選択することを特徴とする半導体メモリの動作方法。

(36) 請求の範囲35の半導体メモリの動作方法において、

「エラー無し」、「データブロックのいずれかの1ビットエラー」、「符号ブロックのいずれかの1ビットエラーである符号エラー」および「2ビット以上のエラー」のいずれかを検出し、

20 エラー判定の優先度を、エラー無し>符号エラー>1ビットエラー>2ビット以上のエラーとし、

優先度の高い項目を真とすることを特徴とする半導体メモリの動作方法。

(37) 請求の範囲36の半導体メモリの動作方法において、

25 「2ビット以上のエラー」を検出したときに、エラー信号をエラー端子に出力することを特徴とする半導体メモリの動作方法。

(38) 請求の範囲35の半導体メモリの動作方法において、

「エラー無し」、「データブロックおよび符号ブロックのいずれかの1ビットエラー」、および「2ビット以上のエラー」のいずれかを検出し、

エラー判定の優先度を、エラー無し>1ビットエラー>2ビット以上のエラー

とし、

優先度の高い項目を真とすることを特徴とする半導体メモリの動作方法。

(39) 請求の範囲27の半導体メモリの動作方法において、

- 訂正不可能なエラーを検出したときに、エラー信号をエラー端子に出力すること
5 を特徴とする半導体メモリの動作方法。

(40) 請求の範囲27の半導体メモリの動作方法において、

訂正したデータを前記データブロックおよび前記符号ブロックに書き戻すこと
を特徴とする半導体メモリの動作方法。

(41) 請求の範囲27の半導体メモリの動作方法において、

- 10 書き込み動作中に、

前記データブロックおよび前記符号ブロックからのデータにより前記情報デー
タを復号し、

復号した情報データの一部を前記書き込みデータに入れ替えて書き換えデー
タ列を生成し、

- 15 前記書き換えデータ列から新たな符号データを求め、

前記書き換えデータ列および新たな符号データを前記データブロックおよび前
記符号ブロックにそれぞれ書き込むことで、前記情報データを入出力するデー
タ端子に一度に供給される書き込みデータを前記データブロックの一部に書き込む
ことを特徴とする半導体メモリの動作方法。

- 20 (42) 請求の範囲27の半導体メモリの動作方法において、

前記各メモリセルのリフレッシュ間隔は、ワーストのメモリセルのデータリテ
ンション時間より長く設定されていることを特徴とする半導体メモリの動作方法。

(43) 請求の範囲27の半導体メモリの動作方法において、

論理0の予想データを付加した情報データ列から第1符号データを生成し、

- 25 論理1の予想データを付加した情報データ列から第2符号データを生成し、

前記第1符号データを前記符号ブロックから読み出される符号データと比較し、
エラーを判定し、

前記第2符号データを前記符号ブロックから読み出される符号データと比較し、
エラーを判定し、

前記第 1 および第 2 エラー判定回路の判定結果のうち、最も確からしい判定結果を真とし、真とした判定結果に基づいてエラーを訂正することを特徴とする半導体メモリの動作方法。

(4 4) 請求の範囲 2 7 の半導体メモリの動作方法において、

- 5 前記データブロックから読み出される情報データ列から符号データを生成し、生成した符号データを、論理 0 の予想データを付加した符号データ列と比較し、エラーを判定し、

生成した符号データを、論理 1 の予想データを付加した符号データ列と比較し、エラーを判定し、

- 10 前記第 1 および第 2 エラー判定回路の判定結果のうち、最も確からしい判定結果を真とし、真とした判定結果に基づいてエラーを訂正することを特徴とする半導体メモリの動作方法。

(4 5) 請求の範囲 2 7 の半導体メモリの動作方法において、

- 15 論理 0 の予想データを付加した読み出しデータ列から第 1 シンドロームを生成しと、

論理 1 の予想データを付加した読み出しデータ列から第 2 シンドロームを生成し、

前記第 1 シンドロームからエラーの有無およびエラーアドレスを検出し、

前記第 2 シンドロームからエラーの有無およびエラーアドレスを検出し、

- 20 前記第 1 および第 2 エラー判定回路の判定結果のうち、最も確からしい判定結果を真とし、真とした判定結果に基づいてエラーを訂正することを特徴とする半導体メモリの動作方法。

(4 6) 請求の範囲 2 7 の半導体メモリの動作方法において、

- 25 前記符号データの最小ハミング距離は、4 以上であることを特徴とする半導体メモリの動作方法。

(4 7) 請求の範囲 4 6 の半導体メモリの動作方法において、

前記符号データは、水平垂直パリティ符号であることを特徴とする半導体メモリの動作方法。

(4 8) 請求の範囲 4 6 の半導体メモリの動作方法において、

前記符号データは、拡大ハミング符号であることを特徴とする半導体メモリの動作方法。

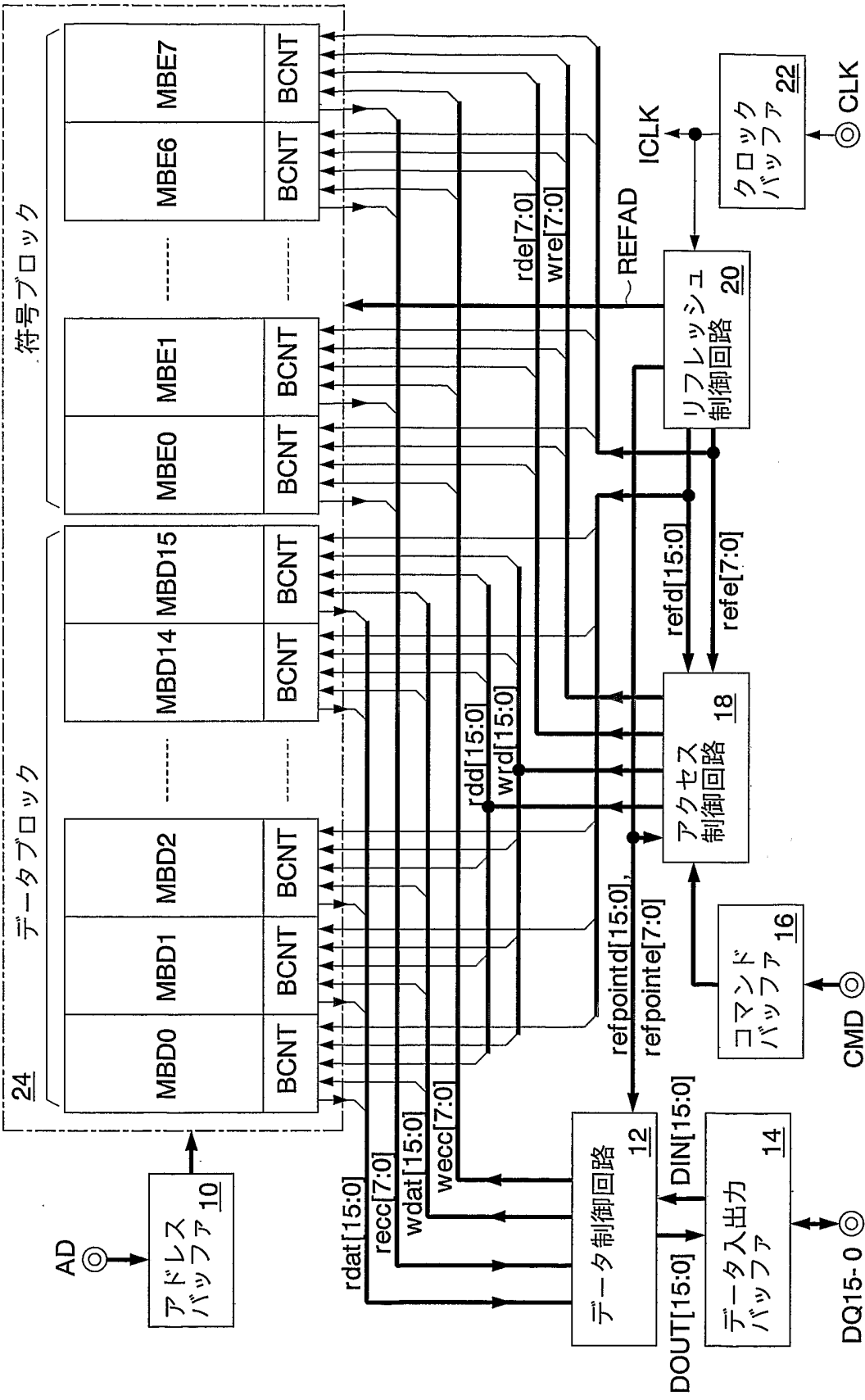


図 1

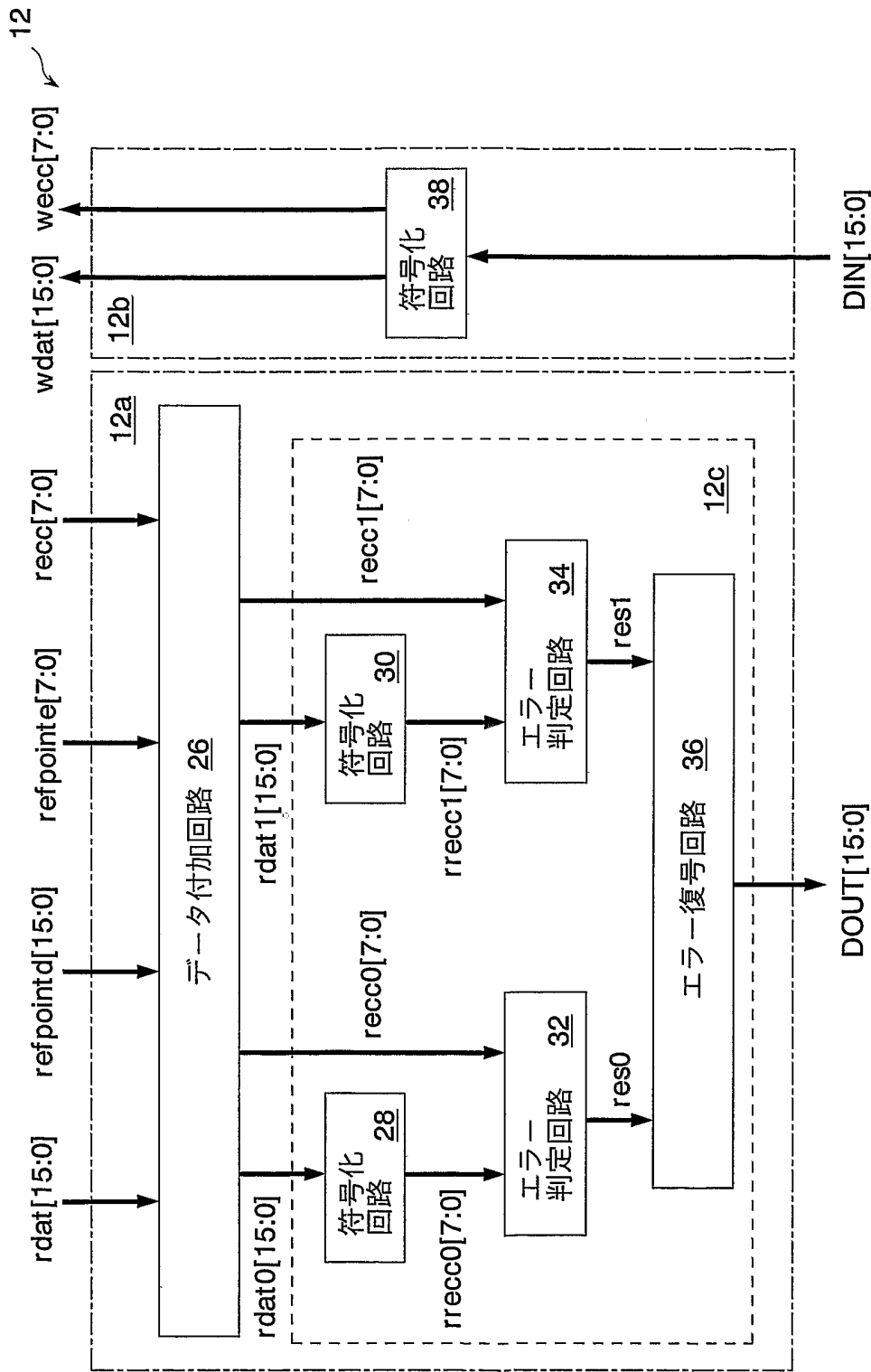
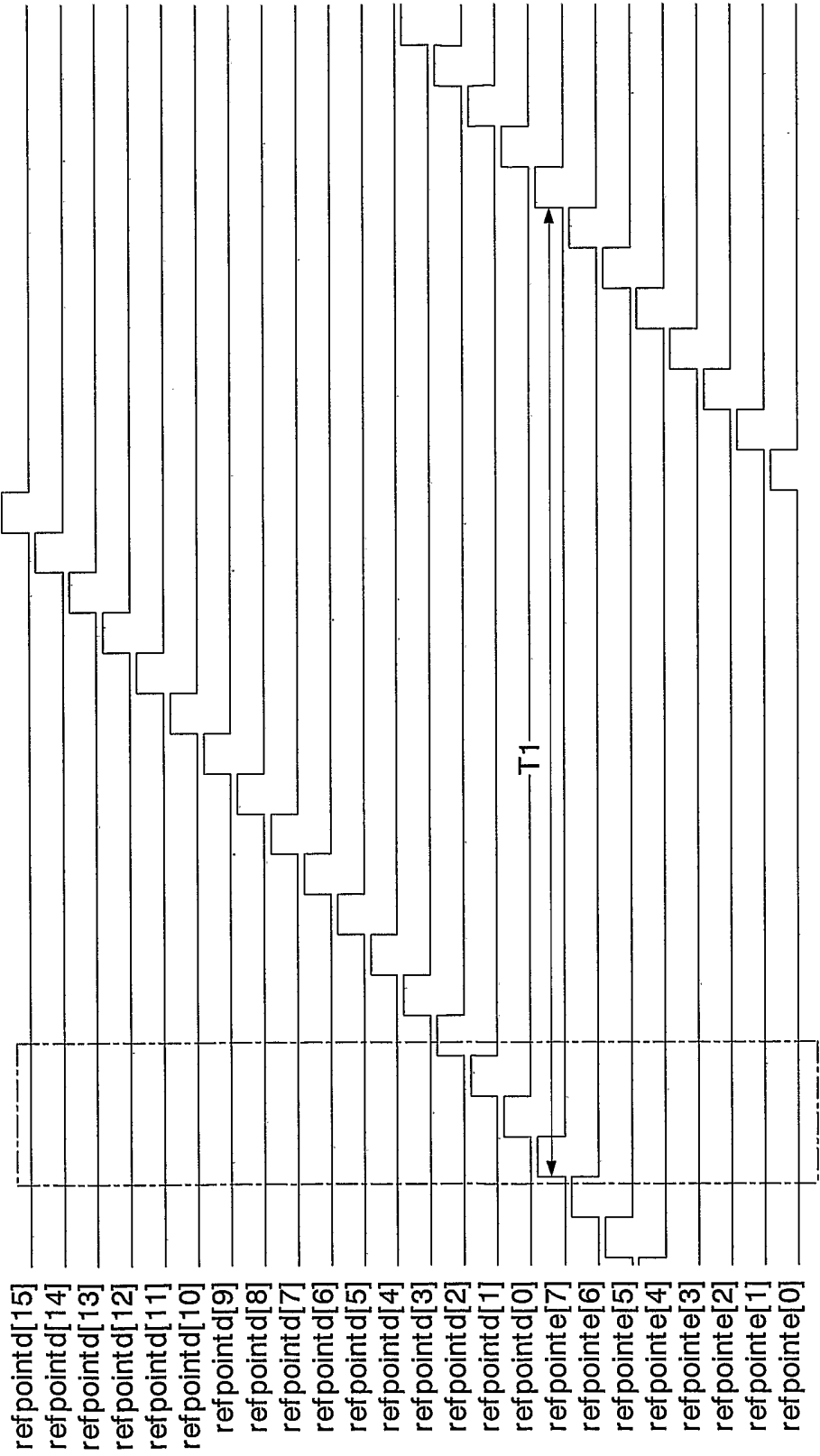


図 2



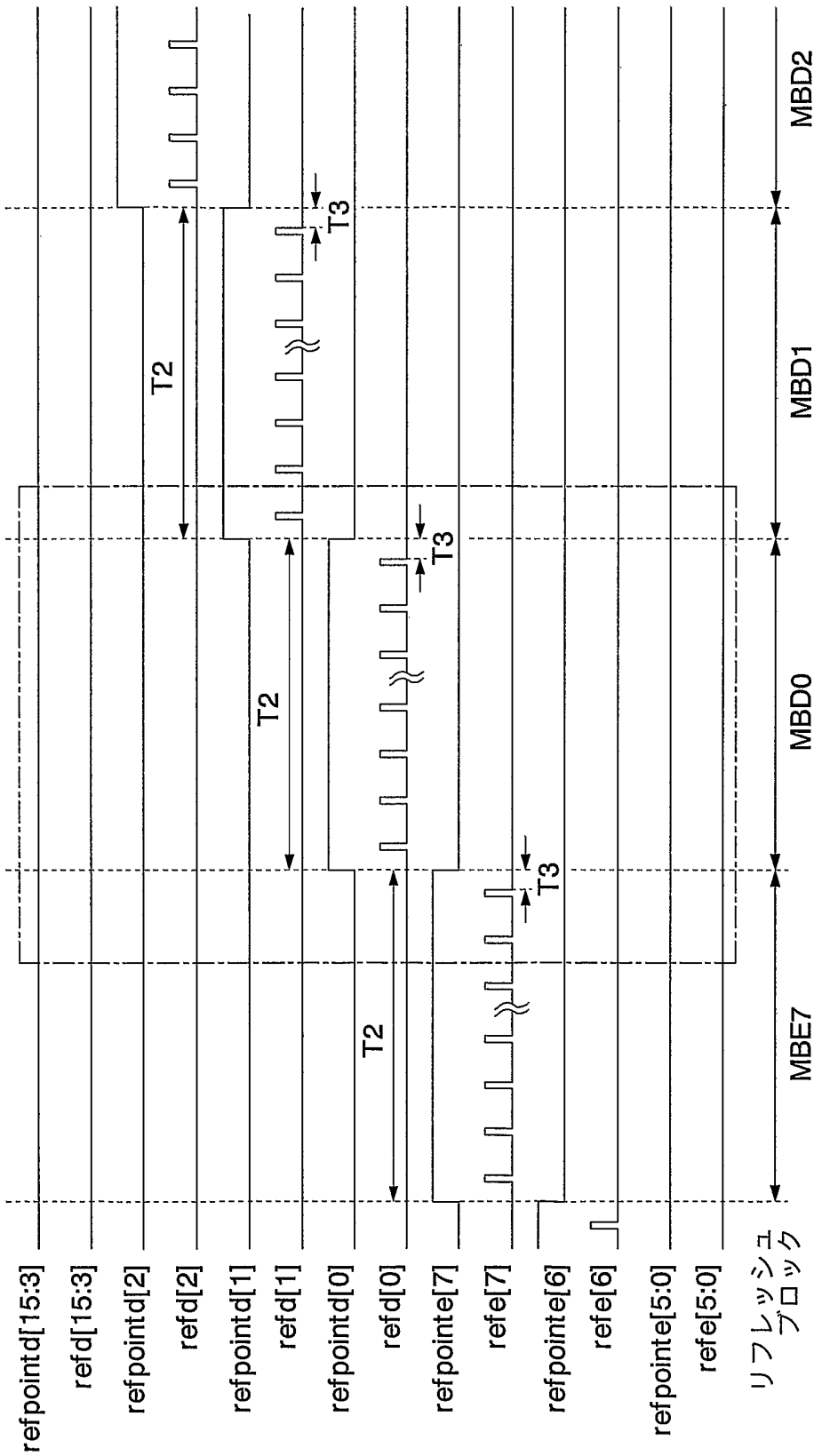
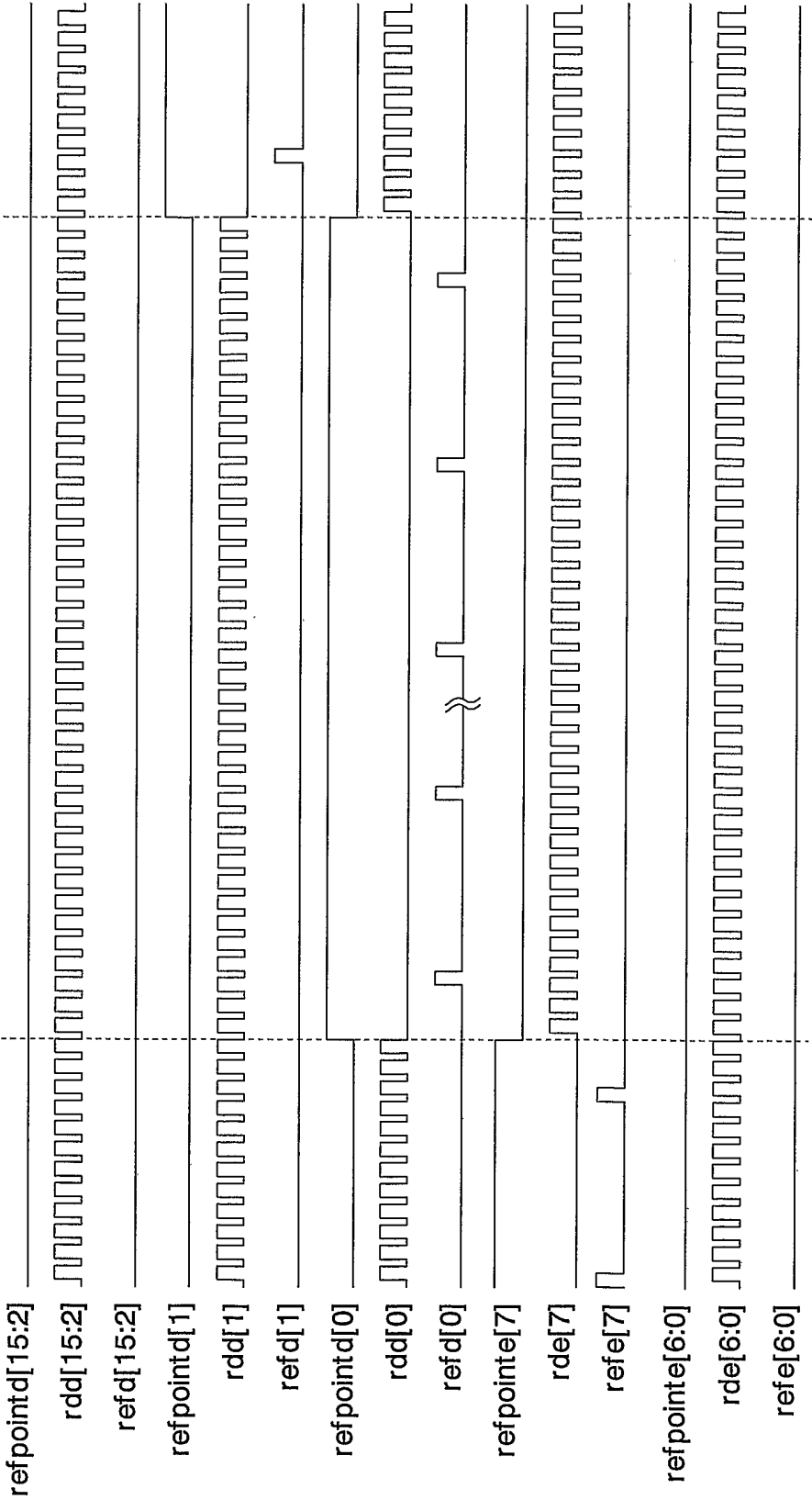


図 4



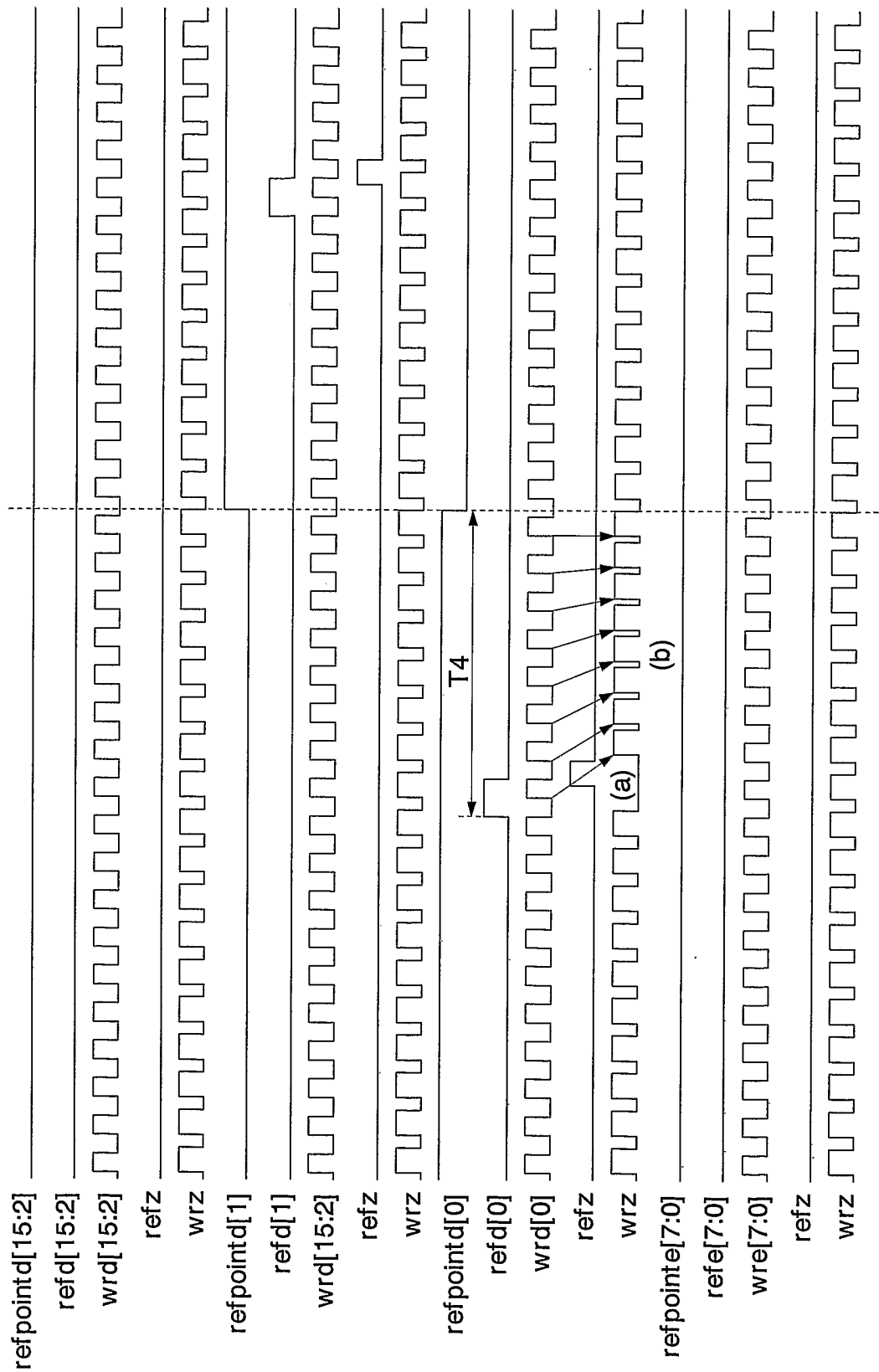


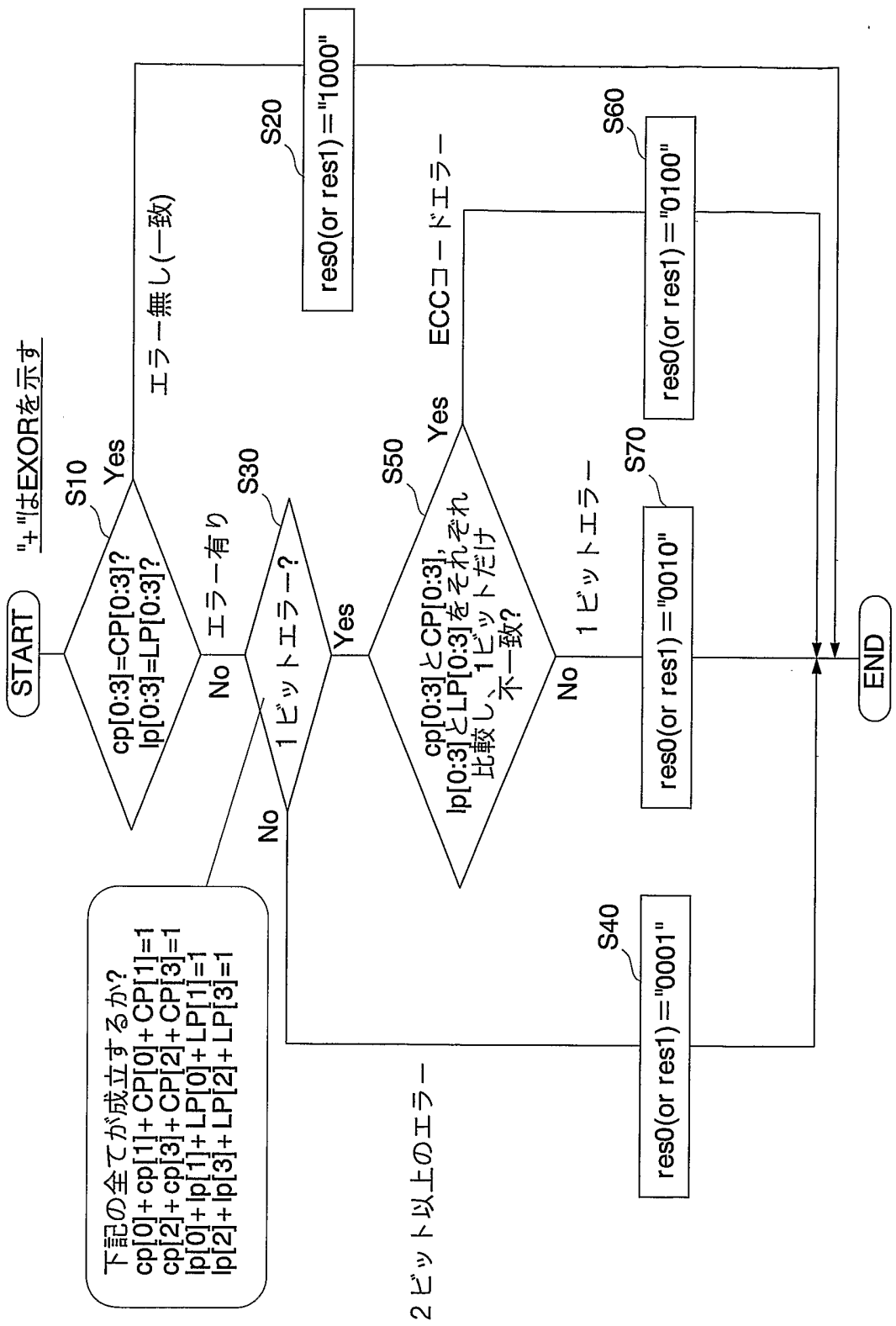
図 6

	C1		C0		L1		L0
	0	1	0	1	0	1	
MBD0							
MBD1							
MBD2							
MBD3							
MBD4							
MBD5							
MBD6							
MBD7							
MBD8							
MBD9							
MBD10							
MBD11							
MBD12							
MBD13							
MBD14							
MBD15							

↓ パリティを生成

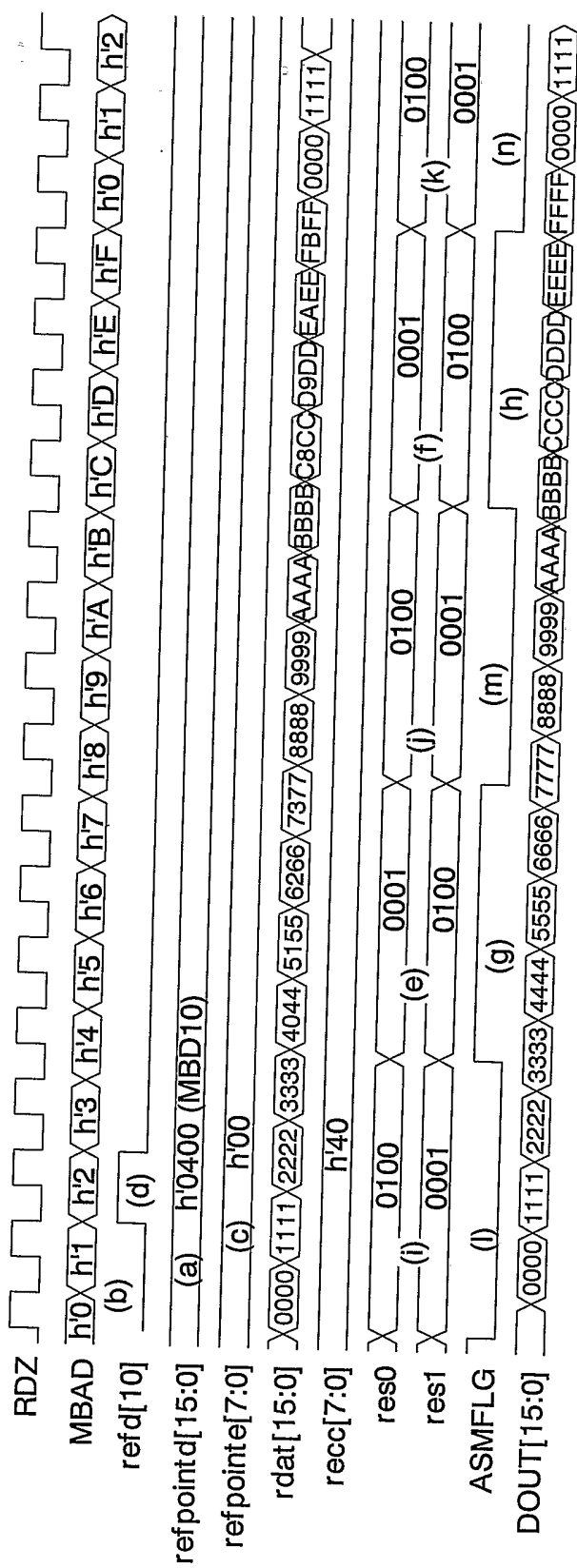
"+"はEXORを示す

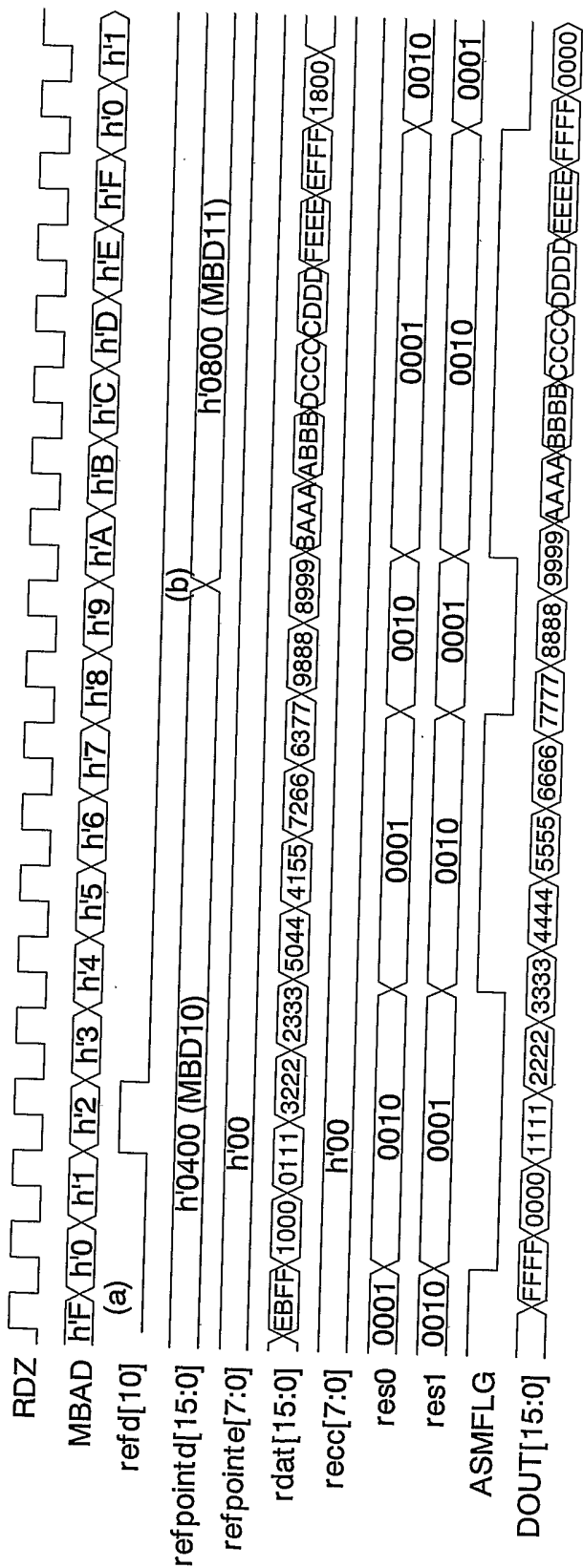
LP0(MBE0)=MBD0+MBD1+MBD2+MBD3+MBD8+MBD9+MBD10+MBD11
LP1(MBE1)=MBD4+MBD5+MBD6+MBD7+MBD12+MBD13+MBD14+MBD15
LP2(MBE2)=MBD0+MBD1+MBD2+MBD3+MBD4+MBD5+MBD6+MBD7
LP3(MBE3)=MBD8+MBD9+MBD10+MBD11+MBD12+MBD13+MBD14+MBD15
CP0(MBE4)=MBD0+MBD4+MBD8+MBD12+MBD2+MBD6+MBD10+MBD14
CP1(MBE5)=MBD1+MBD5+MBD9+MBD13+MBD3+MBD7+MBD11+MBD15
CP2(MBE6)=MBD0+MBD4+MBD8+MBD12+MBD1+MBD5+MBD9+MBD13
CP3(MBE7)=MBD2+MBD6+MBD10+MBD14+MBD3+MBD7+MBD11+MBD15



リフレッシュ中の ブロック	リフレッシュ中のブ ロックから読み出さ れるべきデータ		エラー無し: b'1000, ECCエラー: b'0100, 1ビットエラー: b'0010, 2ビット以上エラー: b'0001			
	res0	res1	(a)	(b)	(c)	ECCエラー
データブロック	0	res1	エラー無し 1ビットエラー	1ビットエラー 2ビット以上エラー	2ビット以上エラー	2ビット以上エラー
	1	res0	1ビットエラー	2ビット以上エラー	2ビット以上エラー	2ビット以上エラー
符号ブロック	0	res1	エラー無し	1ビットエラー	(f)	ECCエラー
		res0	エラー無し	1ビットエラー	(i)	ECCエラー
	1	res1	ECCエラー	2ビット以上エラー	2ビット以上エラー	2ビット以上エラー
		res0	ECCエラー	2ビット以上エラー	(l)	2ビット以上エラー
エラー判定			エラー無し	1ビットエラー		ECCエラー
			エラー無し	データブロックに 1ビットエラー		ECCブロックに 1ビットエラー

エラー判定の優先度：一致>ECCエラー>1ビットエラー>2ビット以上エラー





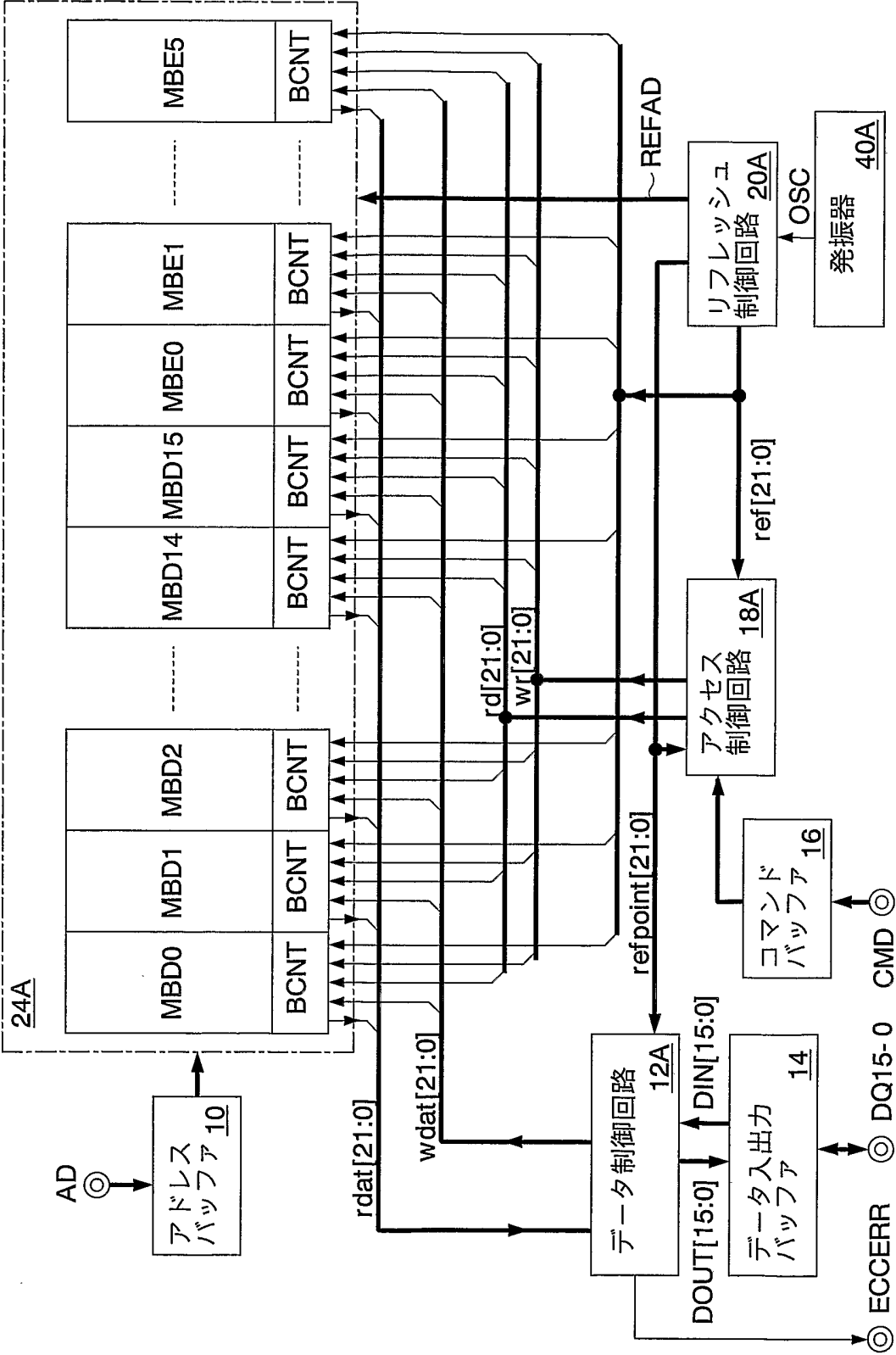


図 1 2

1 3 / 2 3

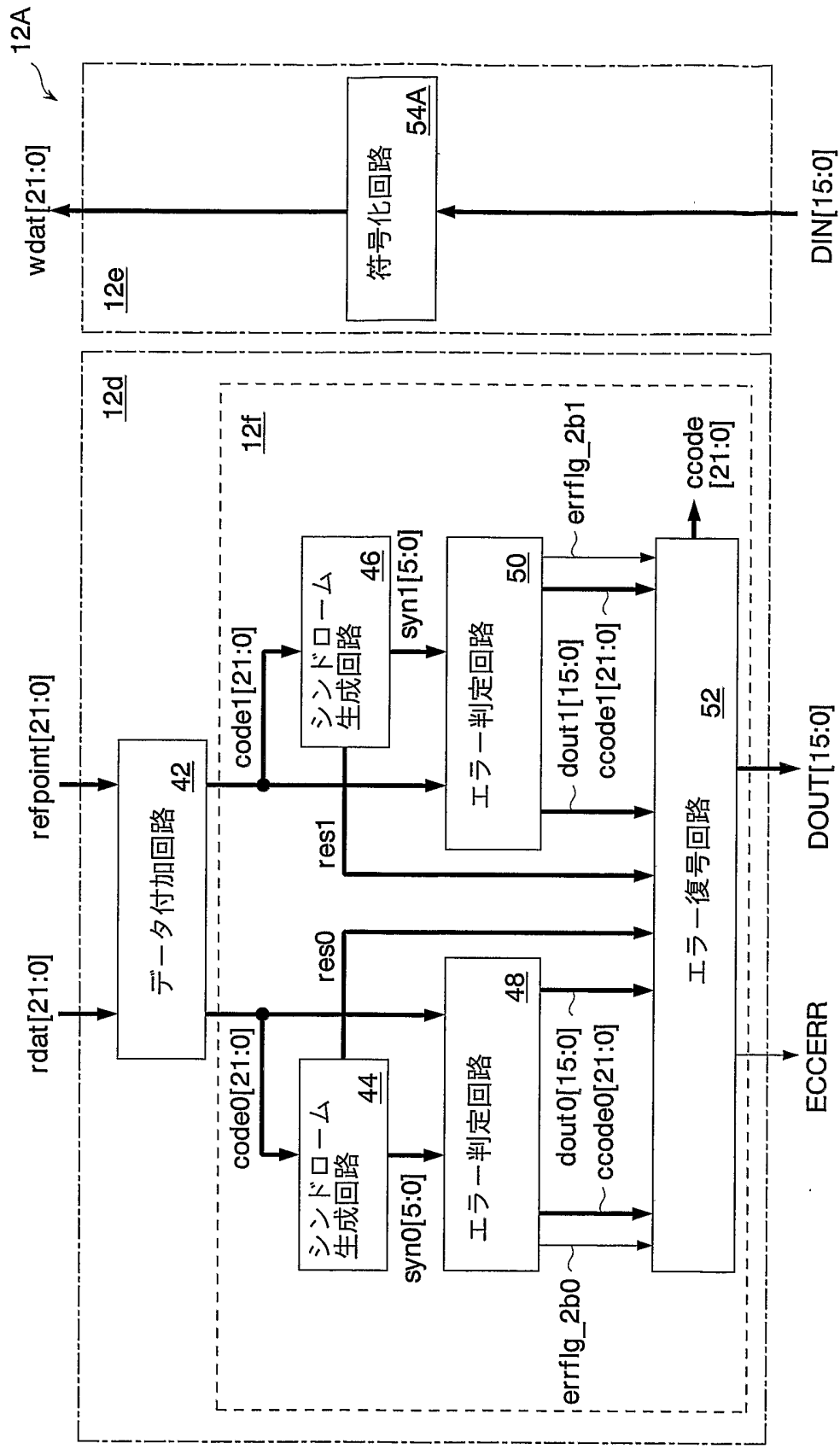


図 1 3

14/23

```

module ECC(syndrome,code,ccode,dout,errflg_2b)
    input[5:0] syndrome
    input[21:0] code
    output[21:0] ccode
    output[15:0] dout
    output errflg_2b
    wire[21:0] errpoint

    ; // 入力シンドローム
    ; // 入力符号
    ; // 訂正済み符号
    ; // 出力情報データ
    ; // 2ビットエラー検出フラグ
    ; // エラー位置ポインタ

    assign errflg_2b = |syndrome[4:0] & ~syndrome[5] ; // 2ビットエラーフラグ論理
    assign errpoint = dec_erradd(syndrome) ; // シンドロームからエラー位置決め
    assign ccode = code ^ errpoint ; // エラー訂正
    assign dout = {ccode[20:16],ccode[14:8],ccode[6:4],ccode[2]} ; // 符号から情報データ抽出

    function[21:0] dec_erradd
    input [5:0] syndrome
    begin
    case(syndrome)
        6'b100001 : dec_erradd = 22'b0000000000000000000001
        6'b100010 : dec_erradd = 22'b0000000000000000000010
        6'b100011 : dec_erradd = 22'b00000000000000000000100
        6'b100100 : dec_erradd = 22'b000000000000000000001000
        6'b100101 : dec_erradd = 22'b0000000000000000000010000
        6'b100110 : dec_erradd = 22'b00000000000000000000100000
        6'b100111 : dec_erradd = 22'b000000000000000000001000000
        6'b101000 : dec_erradd = 22'b0000000000000000000010000000
        6'b101001 : dec_erradd = 22'b00000000000000000000100000000
        6'b101010 : dec_erradd = 22'b000000000000000000001000000000
        6'b101011 : dec_erradd = 22'b0000000000000000000010000000000
        6'b101100 : dec_erradd = 22'b00000000000000000000100000000000
        6'b101101 : dec_erradd = 22'b000000000000000000001000000000000
        6'b101110 : dec_erradd = 22'b0000000000000000000010000000000000
        6'b101111 : dec_erradd = 22'b00000000000000000000100000000000000
        6'b110000 : dec_erradd = 22'b000000000000000000000000000000000
        6'b110001 : dec_erradd = 22'b0000000000000000000000000000000000
        6'b110010 : dec_erradd = 22'b00000000000000000000000000000000000
        6'b110011 : dec_erradd = 22'b000000000000000000000000000000000000
        6'b110100 : dec_erradd = 22'b0000000000000000000000000000000000000
        6'b110101 : dec_erradd = 22'b00000000000000000000000000000000000000
        6'b110110 : dec_erradd = 22'b000000000000000000000000000000000000000
        6'b110111 : dec_erradd = 22'b0000000000000000000000000000000000000000
        default : dec_erradd = 22'b0000000000000000000000000000000000000000 ; //Case @ 0 or 2 bit Error
    endcase
    end
endfunction
endmodule

```

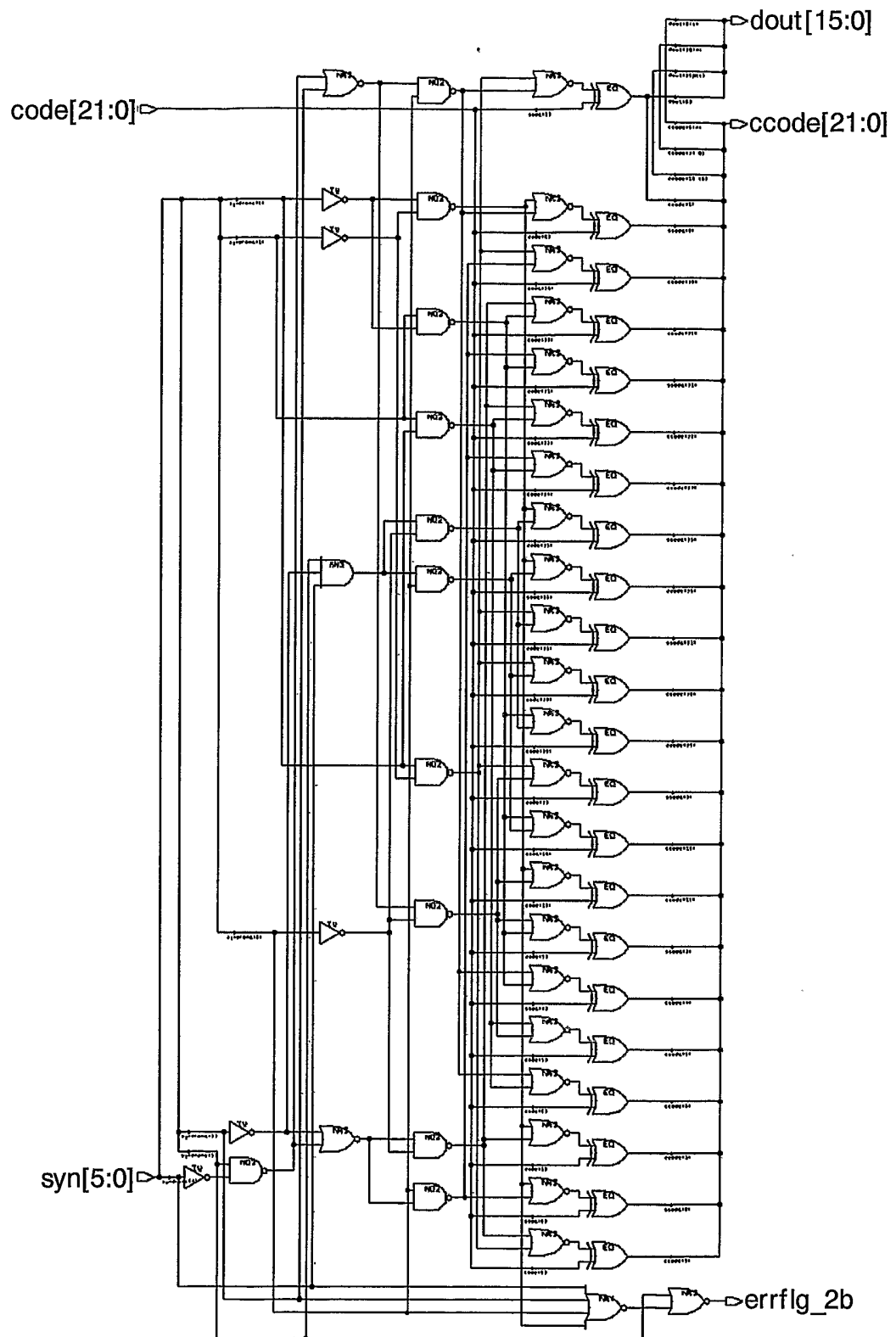
$1\ 5 / 2\ 3$ 

图 15

16 / 23

```

module synd(code,syndrome)
    input[21:0]    code           ; //読み出された符号
    output[5:0]    syndrome       ; //シンドローム
    wire[21:0]     c              ;

    assign c       = code         ;

    // シンドローム演算

    assign syndrome[0] = ^{c[0],c[2],c[4],c[6],c[8],c[10],c[12],c[14],c[16],c[18],c[20]} ;
    assign syndrome[1] = ^{c[1],c[2],c[5],c[6],c[9],c[10],c[13],c[14],c[17],c[18]} ;
    assign syndrome[2] = ^{c[3],c[4],c[5],c[6],c[11],c[12],c[13],c[14],c[19],c[20]} ;
    assign syndrome[3] = ^c[14:7] ;
    assign syndrome[4] = ^c[20:15] ;
    assign syndrome[5] = ^c ;

endmodule

```

図 16

```

module Hamming(data,code)
    input[15:0]    data           ; // 情報ビット
    output[21:0]   code           ; // 拡大ハミングコード
    wire[15:0]     d              ;

    assign d       = data         ;
    assign code[2]  = d[0]         ;
    assign code[6:4] = d[3:1]      ;
    assign code[14:8] = d[10:4]    ;
    assign code[20:16] = d[15:11] ;

    // <<<< 拡大ハミングコード生成ロジック >>>>

    assign code[0]  = ^{d[0],d[1],d[3],d[4],d[6],d[8],d[10],d[11],d[13],d[15]} ;
    assign code[1]  = ^{d[0],d[2],d[3],d[5],d[6],d[9],d[10],d[12],d[13]} ;
    assign code[3]  = ^{d[1],d[2],d[3],d[7],d[8],d[9],d[10],d[14],d[15]} ;
    assign code[7]  = ^d[10:4] ;
    assign code[15] = ^d[15:11] ;
    assign code[21] = ^code[20:0] ;

endmodule

```

図 17

書き込みデータ (期待値)

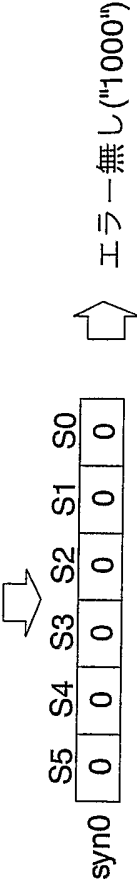
ブロック アドレス	P0 E1	P1 E2	D0 E3	P2 E4	D1 E5	P3 E6	D2 E7	P4 E8	D3 E9	P5 E10	D4 E11	P6 E12	D5 E13	P7 E14	D6 E15	P8 E16	D7 E17	P9 E18	D8 E19	P10 E20	D9 E21	P11 E22
wdat	0	1	0	0	0	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	1

読み出しデータ

ブロック アドレス	P0 E1	P1 E2	D0 E3	P2 E4	D1 E5	P3 E6	D2 E7	P4 E8	D3 E9	P5 E10	D4 E11	P6 E12	D5 E13	P7 E14	D6 E15	P8 E16	D7 E17	P9 E18	D8 E19	P10 E20	D9 E21	P11 E22
rdat	0	1	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	1

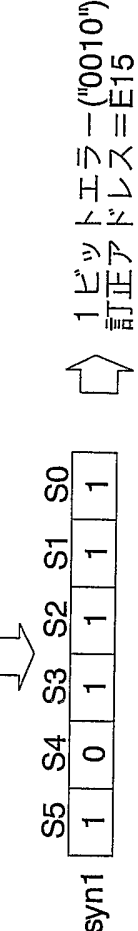
リフレッシュブロックからのデータを"0"としてエラー訂正

ブロック アドレス	P0 E1	P1 E2	D0 E3	P2 E4	D1 E5	P3 E6	D2 E7	P4 E8	D3 E9	P5 E10	D4 E11	P6 E12	D5 E13	P7 E14	D6 E15	P8 E16	D7 E17	P9 E18	D8 E19	P10 E20	D9 E21	P11 E22
code0	0	1	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	1



リフレッシュブロックからのデータを"1"としてエラー訂正

ブロック アドレス	P0 E1	P1 E2	D0 E3	P2 E4	D1 E5	P3 E6	D2 E7	P4 E8	D3 E9	P5 E10	D4 E11	P6 E12	D5 E13	P7 E14	D6 E15	P8 E16	D7 E17	P9 E18	D8 E19	P10 E20	D9 E21	P11 E22
code1	0	1	0	0	0	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	1	0



書き込みデータ (期待値)

ブロック アドレス	P0	P1	D0	P2	D1	D2	D3	P3	D4	D5	D6	D7	D8	D9	D10	P4	D11	S12	D13	D14	D15	P5
wdat	0	1	0	0	0	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	1	0

読み出しデータ

ブロック アドレス	P0	P1	D0	P2	D1	D2	D3	P3	D4	D5	D6	D7	D8	D9	D10	P4	D11	S12	D13	D14	D15	P5
rdat	0	1	0	0	1	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	1	0

リフレッシュブロックからのデータを"0"としてエラー訂正

ブロック アドレス	P0	P1	D0	P2	D1	D2	D3	P3	D4	D5	D6	D7	D8	D9	D10	P4	D11	S12	D13	D14	D15	P5
code0	0	1	0	0	1	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	1	0



	S5	S4	S3	S2	S1	S0
syn0	1	0	0	1	0	1

1ビットエラー ("0010")
訂正アドレス=E5

リフレッシュブロックからのデータを"1"としてエラー訂正

ブロック アドレス	P0	P1	D0	P2	D1	D2	D3	P3	D4	D5	D6	D7	D8	D9	D10	P4	D11	S12	D13	D14	D15	P5
code1	0	1	0	0	1	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	1	0



S5	S4	S3	S2	S1	S0	
syn1	0	0	1	0	1	0

2ビット以上エラー ("0001")

図 19

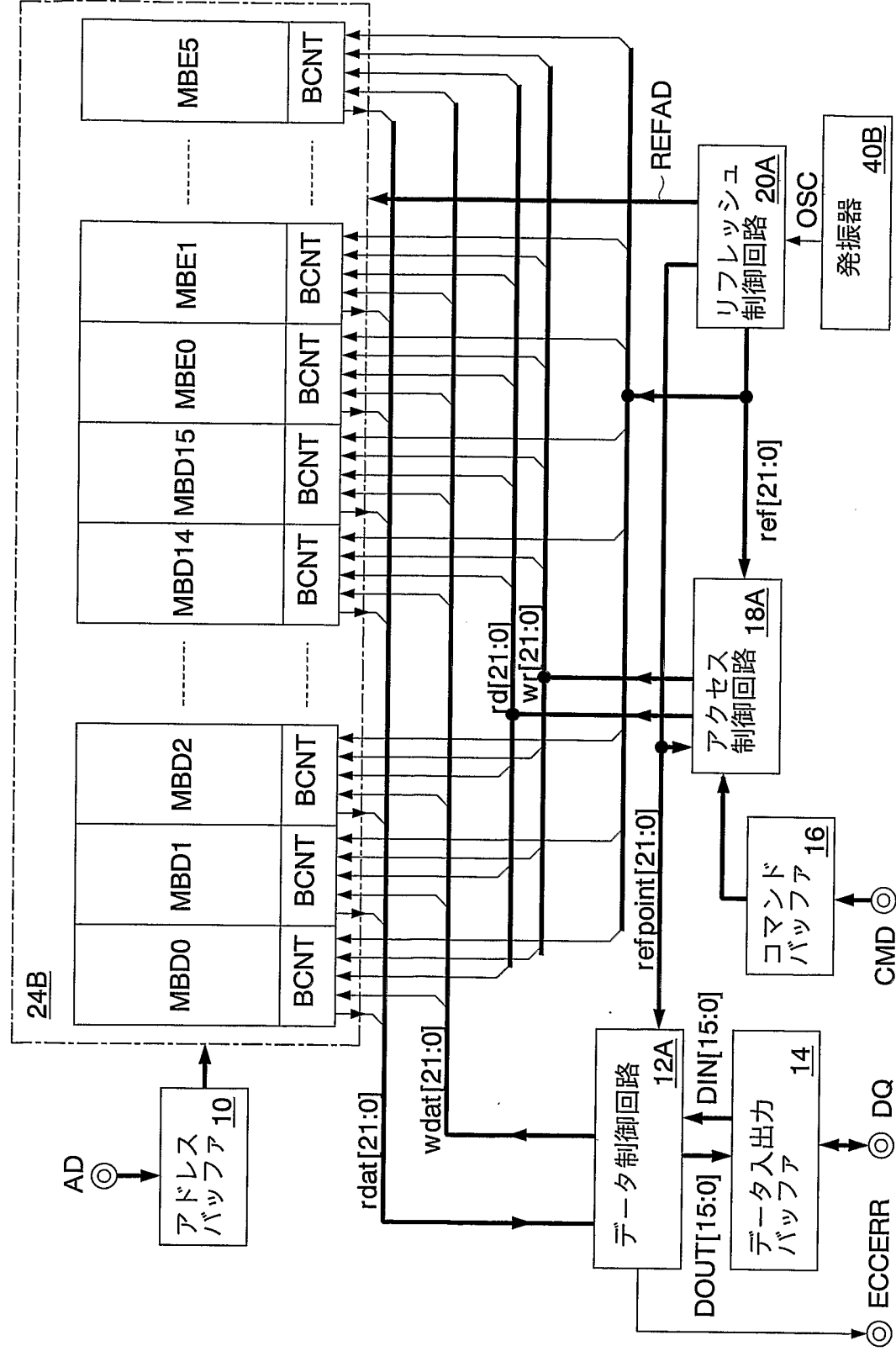


図 20

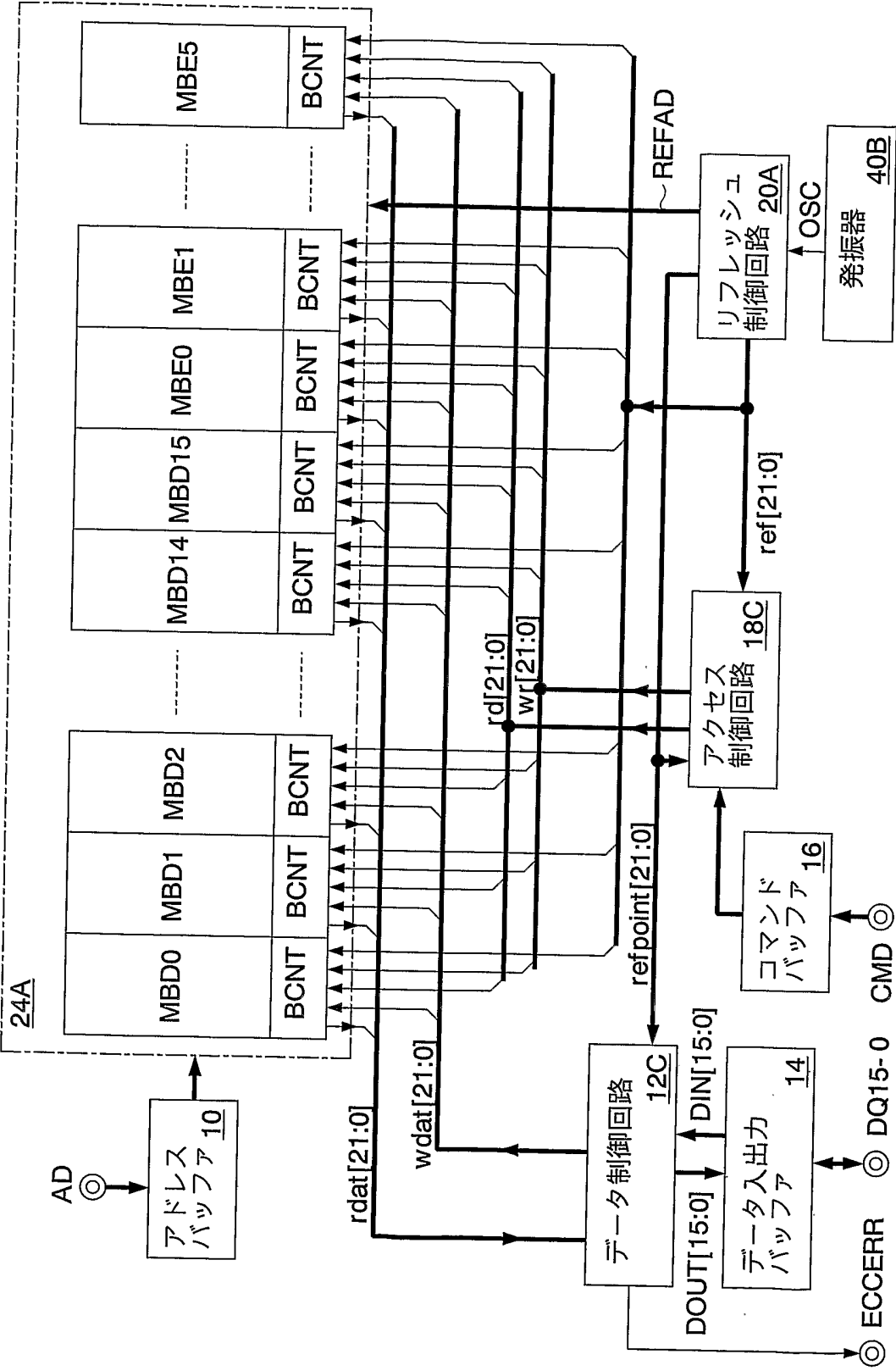


図 21

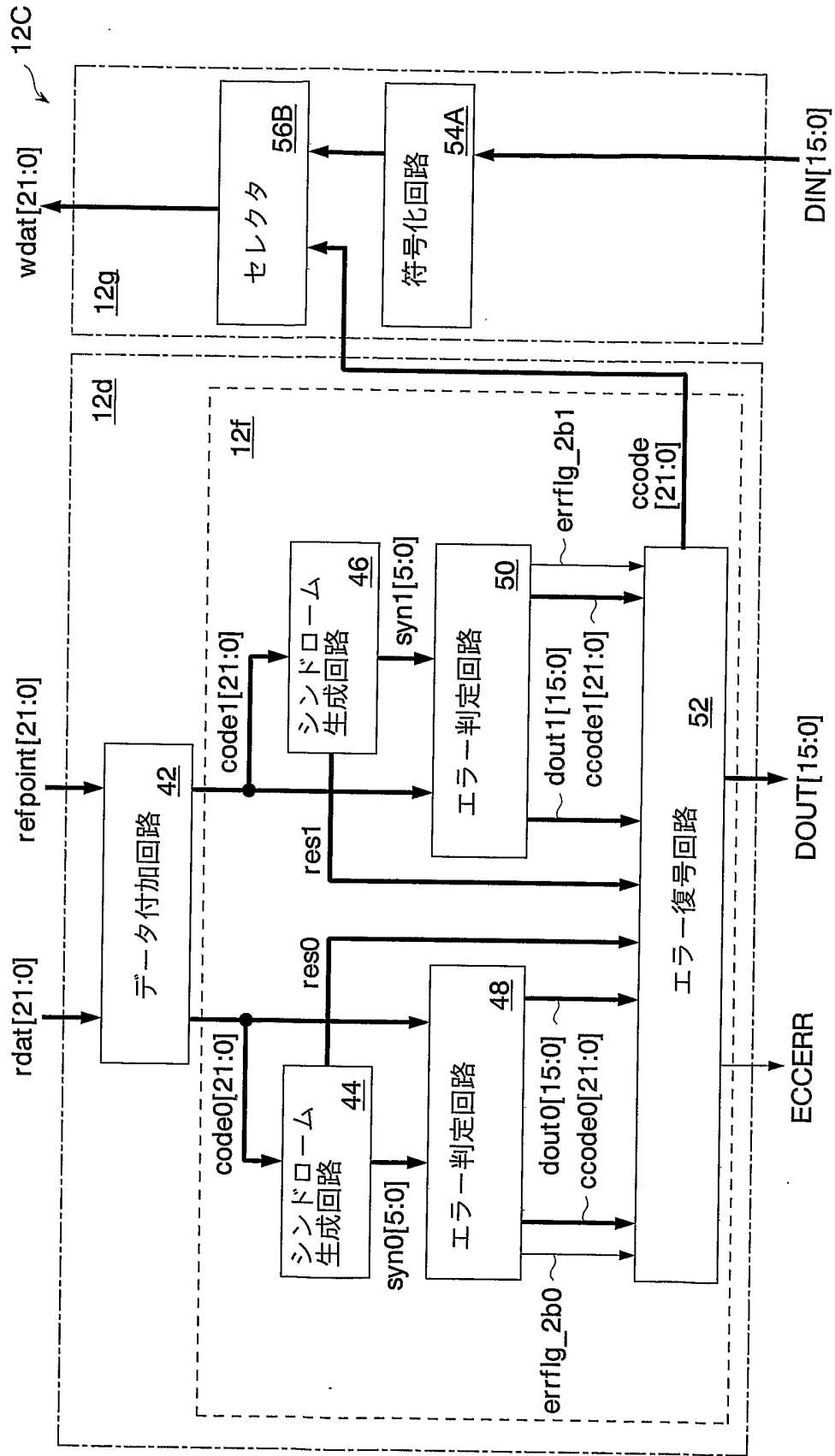


図 22

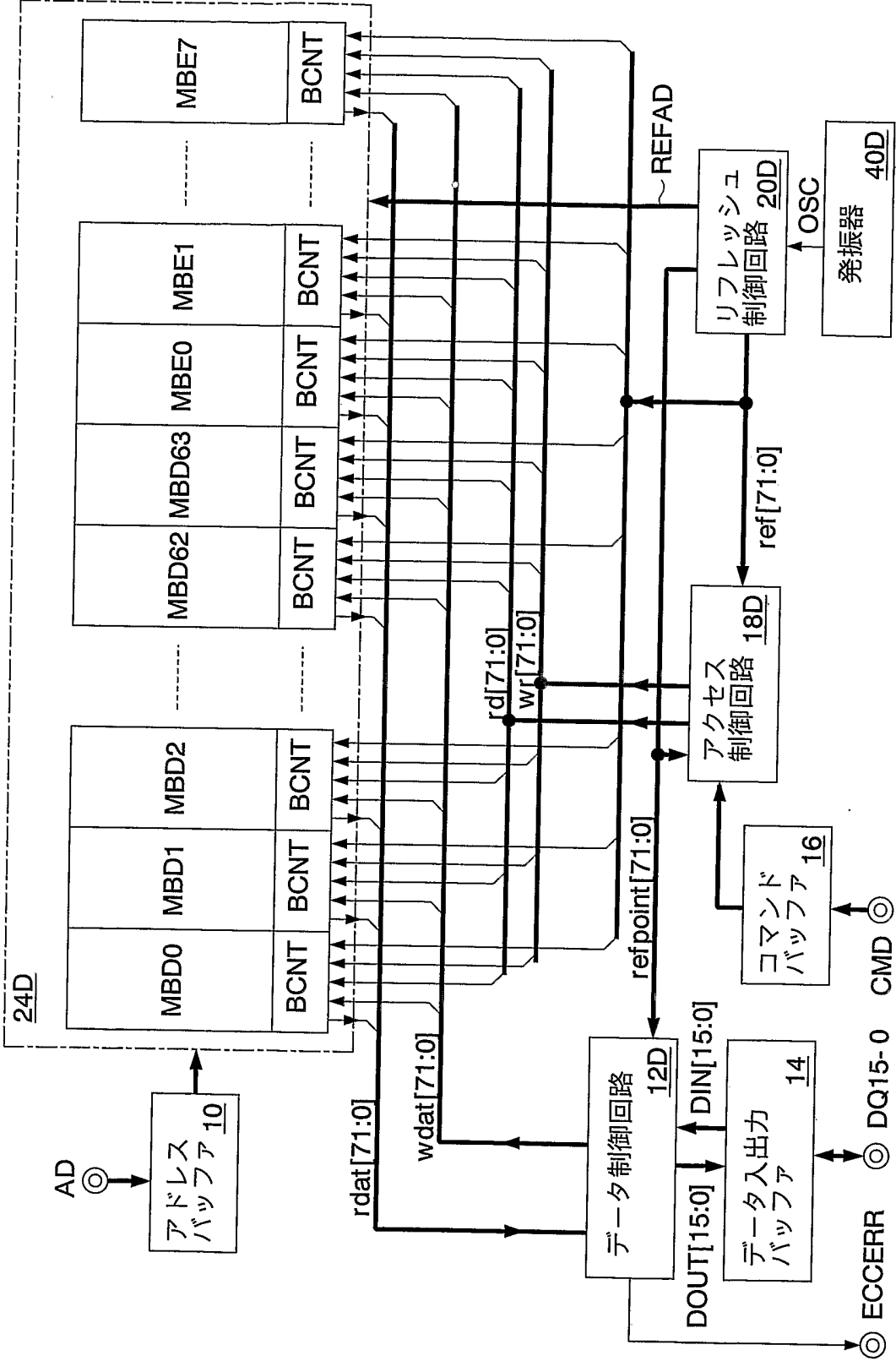


図 23

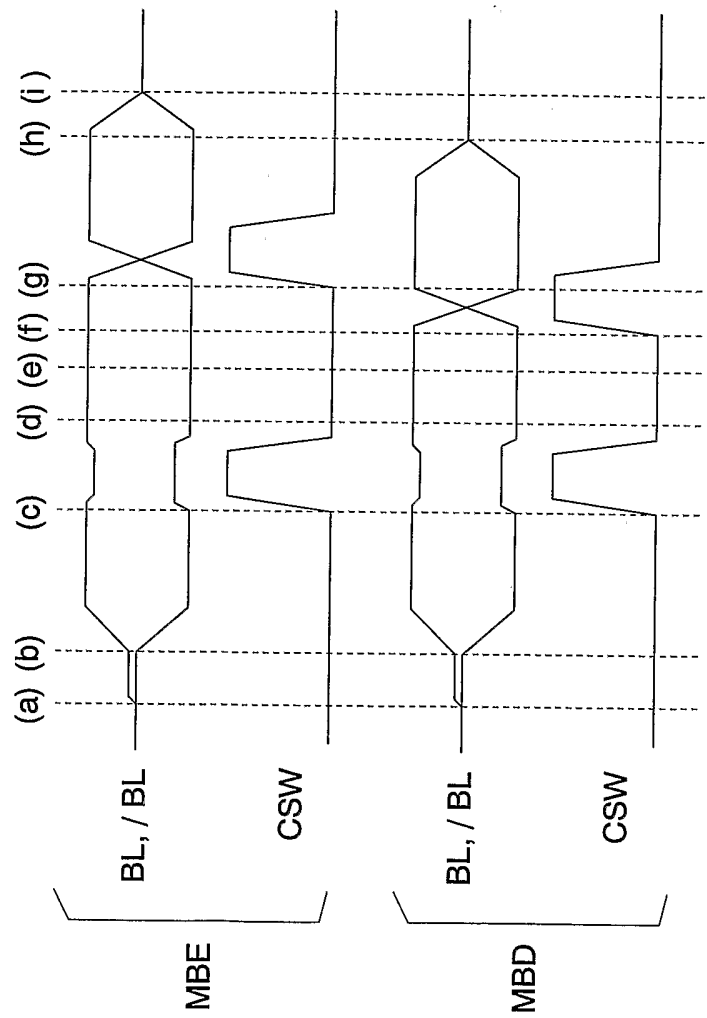


図 24

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP03/10412

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl⁷ G11C29/00, 11/401

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl⁷ G11C29/00, 11/401

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2004
Kokai Jitsuyo Shinan Koho 1971-2004 Toroku Jitsuyo Shinan Koho 1994-2004

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 4-132093 A (Toshiba Corp.), 06 May, 1992 (06.05.92), Full text; Figs. 5 to 6	1, 2, 11, 12, 16, 21-28, 34, 35, 39, 43-48
Y	Full text; Figs. 5 to 6 (Family: none)	3-10, 17-20, 29-33, 40-42
Y	JP 2003-173676 A (Fujitsu Ltd.), 20 June, 2003 (20.06.03), Full text; all drawings & US 2003/0026161 A1 & JP 2003-51186 A & EP 1288961 A2 & CN 1402255 A & KR 2003012812 A & TW 550569 A	3, 18, 29

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document but published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 18 May, 2004 (18.05.04)	Date of mailing of the international search report 01 June, 2004 (01.06.04)
--	--

Name and mailing address of the ISA/ Japanese Patent Office	Authorized officer
Facsimile No.	Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/10412

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2002-170384 A (Fujitsu Ltd.), 14 June, 2002 (14.06.02), Full text; all drawings & TW 512343 A & US 2002/0067649 A1 & EP 1215678 A2 & CN 1357891 A & US 6421292 B1 & KR 2002044045 A	4-7, 30-33
Y	JP 2003-51186 A (Fujitsu Ltd.), 21 February, 2003 (21.02.03), Full text; all drawings & US 2003/0026161 A1 & EP 1288961 A2 & CN 1402255 A & JP 2003-173676 A & KR 2003173676 A & TW 550569 A	5-7, 31-33
Y	JP 4-40697 A (Matsushita Electric Industrial Co., Ltd.), 12 February, 1992 (12.02.92), Full text; all drawings (Family: none)	8-10
Y	JP 62-117198 A (Fujitsu Ltd.), 28 May, 1987 (28.05.87), Full text; all drawings (Family: none)	17, 40
Y	JP 2003-59290 A (Toshiba Corp.), 28 February, 2003 (28.02.03), & US 2002/0184592 A1 & CN 1389872 A	19, 41
Y	JP 2002-56671 A (Hitachi, Ltd.), 22 February, 2002 (22.02.02), Full text; all drawings & US 2002/0018389 A1 & KR 2002013798 A & TW 548648 A & US 6697992 B2	20, 42
A	JP 2002-313077 A (Fujitsu Ltd.), 25 October, 2002 (25.10.02), Full text; all drawings & EP 1251522 A2 & US 2002/0156967 A1 & CN 1381847 A & KR 2003010465 A & TW 546657 A	1-48
A	JP 2003-22694 A (Fujitsu Ltd.), 24 January, 2003 (24.01.03), Full text; all drawings & US 2003/0007410 A1 & CN 1396599 A & US 6535452 B2 & KR 2003006933 A & TW 550595 A	1-48
A	JP 2002-304882 A (Fujitsu Ltd.), 18 October, 2002 (18.10.02), Full text; all drawings & US 2002/0145934 A1 & EP 1251518 A2 & CN 1380698 A & KR 2003009031 A & TW 517382 A & US 6628562 B2	1-48

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/10412

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 62-222497 A (Fujitsu Ltd.), 30 September, 1987 (30.09.87), Full text; all drawings & DE 3781294 A & EP 238417 A2 & US 4766573 A & KR 9102501 B & DE 3781294 T	1-48
E,A	JP 2004-46969 A (Elpida Memoty, Inc.), 12 February, 2004 (12.02.04), Full text; all drawings & US 2004/0008562 A1	20,42

A. 発明の属する分野の分類 (国際特許分類 (IPC))		
Int. Cl ⁷ G11C 29/00, 11/401		
B. 調査を行った分野		
調査を行った最小限資料 (国際特許分類 (IPC))		
Int. Cl ⁷ G11C 29/00, 11/401		
最小限資料以外の資料で調査を行った分野に含まれるもの		
日本国実用新案公報 1922-1996年 日本国公開実用新案公報 1971-2004年 日本国実用新案登録公報 1996-2004年 日本国登録実用新案公報 1994-2004年		
国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)		
C. 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X	JP 4-132093 A (株式会社東芝) 1992. 05. 06 全文, 第5-6図	1, 2, 11, 12, 16, 21-28, 34, 35, 39, 43-48
Y	全文, 第5-6図 (ファミリーなし)	3-10, 17-20, 29-33, 40-42
☒ C欄の続きにも文献が列挙されている。 ☐ パテントファミリーに関する別紙を参照。		
* 引用文献のカテゴリー 「A」 特に関連のある文献ではなく、一般的技術水準を示すもの 「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの 「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す) 「O」 口頭による開示、使用、展示等に言及する文献 「P」 国際出願日前で、かつ優先権の主張の基礎となる出願日の後に公表された文献 「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの 「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの 「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの 「&」 同一パテントファミリー文献		
国際調査を完了した日	18. 05. 2004	国際調査報告の発送日 01. 6. 2004
国際調査機関の名称及びあて先	特許庁審査官 (権限のある職員)	5 N 8840
日本国特許庁 (ISA/JP)	堀田 和義	
郵便番号100-8915	電話番号 03-3581-1101	内線 6840
東京都千代田区霞が関三丁目4番3号		

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
Y	JP 2003-173676 A (富士通株式会社) 2003. 06. 20, 全文, 全図 & US 2003/0026161 A1 & JP 2003-51186 A & EP 1288961 A2 & CN 1402255 A & KR 2003012812 A & TW 550569 A	3, 18, 29
Y	JP 2002-170384 A (富士通株式会社) 2002. 06. 14, 全文, 全図 & TW 512343 A & US 2002/0067649 A1 & EP 1215678 A2 & CN 1357891 A & US 6421292 B1 & KR 2002044045 A	4-7, 30-33
Y	JP 2003-51186 A (富士通株式会社) 2003. 02. 21, 全文, 全図 & US 2003/0026161 A1 & EP 1288961 A2 & CN 1402255 A & JP 2003-173676 A & KR 2003173676 A & TW 550569 A	5-7, 31-33
Y	JP 4-40697 A (松下電器産業株式会社) 1992. 02. 12, 全文, 全図 (ファミリーなし)	8-10
Y	JP 62-117198 A (富士通株式会社) 1987. 05. 28, 全文, 全図 (ファミリーなし)	17, 40
Y	JP 2003-59290 A (株式会社東芝) 2003. 02. 28, 全文, 全図 & US 2002/0184592 A1 & CN 1389872 A	19, 41
Y	JP 2002-56671 A (株式会社日立製作所) 2002. 02. 22, 全文, 全図 & US 2002/0018389 A1 & KR 2002013798 A & TW 548648 A & US 6697992 B2	20, 42

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	JP 2002-313077 A (富士通株式会社) 2002. 10. 25, 全文, 全図 & EP 1251522 A2 & US 2002/0156967 A1 & CN 1381847 A & KR 2003010465 A & TW 546657 A	1-48
A	JP 2003-22694 A (富士通株式会社) 2003. 01. 24, 全文, 全図 & US 2003/0007410 A1 & CN 1396599 A & US 6535452 B2 & KR 2003006933 A & TW 550595 A	1-48
A	JP 2002-304882 A (富士通株式会社) 2002. 10. 18, 全文, 全図 & US 2002/0145934 A1 & EP 1251518 A2 & CN 1380698 A & KR 2003009031 A & TW 517382 A & US 6628562 B2	1-48
A	JP 62-222497 A (富士通株式会社) 1987. 09. 30, 全文, 全図 & DE 3781294 A & EP 238417 A2 & US 4766573 A & KR 9102501 B & DE 3781294 T	1-48
E, A	JP 2004-46969 A (エルピーダメモリ株式会社) 2004. 02. 12, 全文, 全図 & US 2004/0008562 A1	20, 42