

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5082580号
(P5082580)

(45) 発行日 平成24年11月28日(2012.11.28)

(24) 登録日 平成24年9月14日(2012.9.14)

(51) Int.Cl. F I
G06F 12/16 (2006.01)
 G06F 12/16 330D
 G06F 12/16 320F
 G06F 12/16 330C

請求項の数 8 (全 20 頁)

(21) 出願番号	特願2007-129510 (P2007-129510)	(73) 特許権者	000005223
(22) 出願日	平成19年5月15日(2007.5.15)		富士通株式会社
(65) 公開番号	特開2008-287351 (P2008-287351A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43) 公開日	平成20年11月27日(2008.11.27)	(74) 代理人	100108187
審査請求日	平成22年1月19日(2010.1.19)		弁理士 横山 淳一
		(72) 発明者	小口 孝
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	鈴木 賢司
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		審査官	岩間 直純

最終頁に続く

(54) 【発明の名称】 メモリシステム、メモリコントローラ、制御方法及び制御プログラム

(57) 【特許請求の範囲】

【請求項1】

メモリに対して誤り訂正符号が付加されたデータの書き込み又は読み出しを行うメモリコントローラと、該メモリコントローラに命令を送信するシステムコントローラとを備えるメモリシステムにおいて、

該メモリコントローラは、

該メモリに書き込むデータと読み出されたデータに対して、該誤り訂正符号に基づいて検出された第1のエラーの情報を格納するエラー情報格納手段と、

データを該メモリに書き込む際、該エラー情報格納手段に第1のエラーの情報が格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加する付加手段と、

読み出されたデータの該パリティ符号に基づいて第2のエラーを検出するエラー検出手段と、

データが該メモリに書き込まれてから、該エラー検出手段が第2のエラーを検出するまでに、該データが該メモリから読み出された回数を計測する計測手段と、

該エラー検出手段が第2のエラーを検出した場合、第2のエラーを検出した時間情報と、該データに付加された時間情報を取得する時間情報取得手段と、

を有することを特徴とするメモリシステム。

【請求項2】

該エラー情報格納手段が格納する第1のエラーの情報は、該誤り訂正符号に基づいてエラーが検出されたデータのアドレス情報を含み、

該付加手段は、該メモリに書き込むデータのアドレスが該エラー情報格納手段に格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加することを特徴とする請求項1記載のメモリシステム。

【請求項3】

メモリに対して誤り訂正符号が付加されたデータの書き込み又は読み出しを行うメモリコントローラにおいて、

該メモリに書き込むデータと読み出されたデータに対して、該誤り訂正符号に基づいて検出された第1のエラーの情報を格納するエラー情報格納手段と、

データを該メモリに書き込む際、該エラー情報格納手段に第1のエラーの情報が格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加する付加手段と、

読み出されたデータの該パリティ符号に基づいて第2のエラーを検出するエラー検出手段と、

データが該メモリに書き込まれてから、該エラー検出手段が第2のエラーを検出するまでに、該データが該メモリから読み出された回数を計測する計測手段と、

該エラー検出手段が第2のエラーを検出した場合、第2のエラーを検出した時間情報と、該データに付加された時間情報を取得する時間情報取得手段と、

を有することを特徴とするメモリコントローラ。

【請求項4】

該エラー情報格納手段が格納する第1のエラーの情報は、該誤り訂正符号に基づいてエラーが検出されたデータのアドレス情報を含み、

該付加手段は、該メモリに書き込むデータのアドレスが該エラー情報格納手段に格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加することを特徴とする請求項3記載のメモリコントローラ。

【請求項5】

メモリに対して誤り訂正符号が付加されたデータの書き込み又は読み出しを行うメモリコントローラを制御する制御方法において、

該メモリに書き込むデータと読み出されたデータに対して、該誤り訂正符号に基づいて検出された第1のエラーの情報をエラー情報格納手段に格納するエラー情報格納工程と、

データを該メモリに書き込む際、該エラー情報格納手段に第1のエラーの情報が格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加する付加工程と、

読み出されたデータの該パリティ符号に基づいて第2のエラーを検出するエラー検出工程と、

データが該メモリに書き込まれてから、該エラー検出工程によって第2のエラーが検出されるまでに、該データが該メモリから読み出された回数を計測する計測工程と、

該エラー検出工程によって第2のエラーが検出された場合、第2のエラーが検出された時間情報と、該データに付加された時間情報を取得する時間情報取得工程と、

を有することを特徴とする制御方法。

【請求項6】

該エラー情報格納手段が格納する第1のエラーの情報は、該誤り訂正符号に基づいてエラーが検出されたデータのアドレス情報を含み、

該付加工程は、該メモリに書き込むデータのアドレスが該エラー情報格納手段に格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ

10

20

30

40

50

ティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加することを特徴とする請求項 5 記載の制御方法。

【請求項 7】

メモリに対して誤り訂正符号が付加されたデータの書き込み又は読み出しを行うメモリコントローラを制御する制御プログラムであって、該メモリコントローラを、

該メモリに書き込むデータと読み出されたデータに対して、該誤り訂正符号に基づいて検出された第 1 のエラーの情報を格納するエラー情報格納手段と、

データを該メモリに書き込む際、該エラー情報格納手段に第 1 のエラーの情報が格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加する付加手段と、

読み出されたデータの該パリティ符号に基づいて第 2 のエラーを検出するエラー検出手段と、

データが該メモリに書き込まれてから、該エラー検出手段が第 2 のエラーを検出するまでに、該データが該メモリから読み出された回数を計測する計測手段と、

該エラー検出手段が第 2 のエラーを検出した場合、第 2 のエラーを検出した時間情報と、該データに付加された時間情報を取得する時間情報取得手段として機能させることを特徴とする制御プログラム。

【請求項 8】

該エラー情報格納手段が格納する第 1 のエラーの情報は、該誤り訂正符号に基づいてエラーが検出されたデータのアドレス情報を含み、

該付加手段は、該メモリに書き込むデータのアドレスが該エラー情報格納手段に格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加することを特徴とする請求項 7 記載の制御プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、メモリシステムの不具合の原因の推定に関する。

【背景技術】

【0002】

中央処理装置（CPU：Central Processing Unit）、システムコントローラ、メモリコントローラ及びメモリモジュール等の複数のデバイスから構成されるシステムがある。このようなシステムを構成するにあたり、まずは各デバイスが正常に動作するか否かを確認する試験を行う。その後、試験を通過したデバイスのみからシステムを構成する。

【0003】

しかし、試験を通過したデバイスのみからシステムを構成しても、各デバイスの試験で検出されなかった異常が原因でシステムに不具合が発生してしまうことがある。ここで、不具合の原因がデバイスまたはシステムのどちらにあるかを特定するのは困難であり、多

【0004】

先行技術文献としては下記のものがある。

【特許文献 1】特開平 11 - 120715 号公報

【特許文献 2】特開 2006 - 31764 号公報

【発明の開示】

【発明が解決しようとする課題】

【0005】

本発明の課題は、複数のデバイスから構成されるシステムにおいて、不具合の原因を推定するために活用可能な情報を提供することである。

10

20

30

40

50

【課題を解決するための手段】

【0006】

上記の課題を解決するため、メモリに対して誤り訂正符号が付加されたデータの書き込み又は読み出しを行うメモリコントローラと、該メモリコントローラに命令を送信するシステムコントローラとを備えるメモリシステムにおいて、該メモリコントローラは、該メモリに書き込むデータと読み出されたデータに対して、該誤り訂正符号に基づいて検出された第1のエラーの情報を格納するエラー情報格納手段と、データを該メモリに書き込む際、該エラー情報格納手段に第1のエラーの情報が格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加する付加手段と、読み出されたデータの該パリティ符号に基づいて第2のエラーを検出するエラー検出手段と、データが該メモリに書き込まれてから、該エラー検出手段が第2のエラーを検出するまでに、該データが該メモリから読み出された回数を計測する計測手段と、該エラー検出手段が第2のエラーを検出した場合、第2のエラーを検出した時間情報と、該データに付加された時間情報を取得する時間情報取得手段とを有する。

10

【0007】

また、該エラー情報格納手段が格納する第1のエラーの情報は、該誤り訂正符号に基づいてエラーが検出されたデータのアドレス情報を含み、該付加手段は、該メモリに書き込むデータのアドレスが該エラー情報格納手段に格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加する。

20

【0008】

また、メモリに対して誤り訂正符号が付加されたデータの書き込み又は読み出しを行うメモリコントローラにおいて、該メモリに書き込むデータと読み出されたデータに対して、該誤り訂正符号に基づいて検出された第1のエラーの情報を格納するエラー情報格納手段と、データを該メモリに書き込む際、該エラー情報格納手段に第1のエラーの情報が格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加する付加手段と、読み出されたデータの該パリティ符号に基づいて第2のエラーを検出するエラー検出手段と、データが該メモリに書き込まれてから、該エラー検出手段が第2のエラーを検出するまでに、該データが該メモリから読み出された回数を計測する計測手段と、該エラー検出手段が第2のエラーを検出した場合、第2のエラーを検出した時間情報と、該データに付加された時間情報を取得する時間情報取得手段とを有する。

30

【0009】

また、該エラー情報格納手段が格納する第1のエラーの情報は、該誤り訂正符号に基づいてエラーが検出されたデータのアドレス情報を含み、該付加手段は、該メモリに書き込むデータのアドレスが該エラー情報格納手段に格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加する。

【0010】

また、メモリに対して誤り訂正符号が付加されたデータの書き込み又は読み出しを行うメモリコントローラを制御する制御方法において、該メモリに書き込むデータと読み出されたデータに対して、該誤り訂正符号に基づいて検出された第1のエラーの情報をエラー情報格納手段に格納するエラー情報格納工程と、データを該メモリに書き込む際、該エラー情報格納手段に第1のエラーの情報が格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加する付加工程と、読み出されたデータの該パリティ符号に基づいて第2のエラーを検出するエラー検出工程と、データが該メモリに書き込まれてから、該エラー検出工程によって第2のエラーが検出されるまでに、該データが該メモリから読み出された回数を計測する計測工程と、該エラー検出工程によって第2のエラーが

40

50

検出された場合、第2のエラーが検出された時間情報と、該データに付加された時間情報を取得する時間情報取得工程とを有する。

【0011】

また、該エラー情報格納手段が格納する第1のエラーの情報、該誤り訂正符号に基づいてエラーが検出されたデータのアドレス情報を含み、該付加工程は、該メモリに書き込むデータのアドレスが該エラー情報格納手段に格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加する。

【0012】

また、メモリに対して誤り訂正符号が付加されたデータの書き込み又は読み出しを行うメモリコントローラを制御する制御プログラムであって、該メモリコントローラを、

該メモリに書き込むデータと読み出されたデータに対して、該誤り訂正符号に基づいて検出された第1のエラーの情報を格納するエラー情報格納手段と、データを該メモリに書き込む際、該エラー情報格納手段に第1のエラーの情報が格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加する付加手段と、読み出されたデータの該パリティ符号に基づいて第2のエラーを検出するエラー検出手段と、データが該メモリに書き込まれてから、該エラー検出手段が第2のエラーを検出するまでに、該データが該メモリから読み出された回数を計測する計測手段と、該エラー検出手段が第2のエラーを検出した場合、第2のエラーを検出した時間情報と、該データに付加された時間情報を取得する時間情報取得手段として機能させる。

【0013】

また、該エラー情報格納手段が格納する第1のエラーの情報、該誤り訂正符号に基づいてエラーが検出されたデータのアドレス情報を含み、該付加手段は、該メモリに書き込むデータのアドレスが該エラー情報格納手段に格納されている場合、該データに付加された誤り訂正符号を該データに対応した時間情報とパリティ符号に置き換えるために該時間情報と該パリティ符号を該データに付加する。

【発明の効果】

【0026】

本発明によれば、データがメモリに書き込まれてから読み出しエラーが発生するまでの経過時間と当該データの読み出し回数とを取得することができる。そのため、システムの不具合の原因を推定するために活用可能な情報を提供できるという効果を有する。

【発明を実施するための最良の形態】

【0027】

以下に図面を用いて本発明の実施の形態について説明する。

【0028】

図1は、メモリシステムの全体構成図である。メモリシステム500は、CPU400、メモリモジュール200、メモリコントローラ100及びシステムコントローラ300から構成される。さらに、メモリコントローラ100は、1つ以上のメモリモジュール200とメモリデータバス201を介して接続される。システムコントローラ300は、1

【0029】

システムコントローラ300は、CPU400及びメモリコントローラ100を制御する。システムコントローラ300は、メモリコントローラ100に、同期制御とメモリアクセスを命令する。メモリコントローラ100は、システムコントローラ300からの命令に基づいてメモリアクセスを行う。

【0030】

図2は、実施形態の例を示すものである。図1のシステム全体構成図のうち、特にメモリコントローラ100、メモリモジュール200を表している。メモリモジュール200へのデータライト時には、メモリコントローラ100は、システムコントローラ300か

10

20

30

40

50

らライトするデータをライトデータキュー102に受信する。そして、メモリコントローラ100は、メモリモジュール200に対して、ライトを実行する。メモリコントローラ100が1回のライトを実行する際に、メモリモジュール200に送信するデータ形式を図3に表す。図3に表すように、データは、例えば、 $n(n-1) \times 8 \text{ Byte}$ から構成される。そして、各8Byteデータには、例えば、8bitの誤り訂正符号(ECC: Error Correcting code)が付加されている。メモリコントローラ100は、8Byteデータと当該データに付加された誤り訂正符号を n 回($n-1$)でメモリモジュール200に送信する。このようにして、メモリコントローラ100は、データをメモリモジュール200にライトする。

【0031】

メモリモジュール200からのデータリード時には、メモリコントローラ100は、メモリモジュール200に対して、必要な回数リードを実行する。メモリコントローラ100が1回のリードを実行する際に、メモリモジュール200から受信するデータ形式は、ライトを実行する際のデータ形式と同様であり、図3に表したとおりである。

【0032】

メモリコントローラ100は、メモリモジュール200からのデータのリード後、リードしたデータ及び当該データに付加された誤り訂正符号に基づいて、誤り検出及び訂正を行う。誤り検出及び訂正によりデータの誤り検出及び訂正を行った後、メモリコントローラ100はデータをリードデータキュー104に受信し、リードデータをシステムコントローラ300へ送信する。誤り検出及び訂正により、1ビットエラーは自動訂正され、システムの継続運転が可能となっている。また、2ビットエラーについては、確実に問題を検出し、安全にシステムを停止することができる。メモリコントローラ100は、誤り検出及び訂正により検出したエラー情報をエラーレジスタに格納する。ここで、エラーレジスタとは専用の経路を用いてファームウェアにより読み出し可能なレジスタであり、必要に応じてエラー情報が読み出される。

【0033】

タイムスタンプ書き込み部108は、ライトデータの誤り訂正符号をパリティビットに置き換える。また、メモリシステムを駆動するためのシステムクロック(図示せず)から取得した時間情報に基づいてタイムスタンプを生成する手段を備える。タイムスタンプとは、一般には、データがいつ作成され、その後現在まで修正が行われていないかを客観的に証明するための技術である。タイムスタンプ書き込み部108はライトデータの形式を図4に表したように誤り訂正符号をライト時のタイムスタンプとデータのリードエラーを検出するためのパリティビットに置き換える。図4に表したように、データ1からデータ n に付加されていた誤り訂正符号 n 個分のデータ、即ち、 $n \times 8 \text{ bit}$ のデータをタイムスタンプと、パリティビットに置き換える。ここでは、1bitをパリティビットを付加するために使用し、残りの($n \times 8 - 1$)ビットをタイムスタンプを付加するために使用する。これによれば、データにタイムスタンプ及びパリティビットを付加しても、データ量が大きくなることはない。

【0034】

データチェック部110は、リードデータの誤り検出及び訂正を行う。さらに、タイムスタンプが付加されたデータについてはパリティチェックを行うとともに、タイムスタンプ及びパリティビットが付加されたデータを、図3に表したように誤り訂正符号が付加されたデータに戻す。

【0035】

タイムスタンプ読み出し部112は、タイムスタンプが付加されたリードデータからデータライト時のタイムスタンプを取得する。さらに、タイムスタンプ読み出し部112は、データリード時のタイムスタンプを取得する。

【0036】

エラーアドレスカウント部114は、通常動作中にデータチェック部110においてリードエラーが検出されたデータのメモリアクセスアドレス(エラー発生アドレス)を格納

10

20

30

40

50

する。さらに、エラーアドレスカウンタ部 114 は、エラー発生アドレスに対するデータリード回数をカウントするカウンタを備える。カウンタは、エラー発生アドレスに対して、データリードが行われる毎にカウントアップする。また、カウンタは、エラー発生アドレスに対してデータライトが行われるとカウントアップした値をクリアする。

【0037】

通常動作中において、エラーアドレスカウンタ部 114 は、データリード時にリードエラーを検出した場合は、エラーが発生したメモリアクセスアドレスを格納する。エラー発生後は、エラー発生アドレスに対しては、タイムスタンプ書き込み部 108 が、データにタイムスタンプを付加してメモリモジュール 200 にライトする。タイムスタンプが付加されたデータがデータライトされた後、エラー発生アドレスに対するデータリード時は、

10

データチェック部 110 は、パリティチェックによるデータチェックを行う。

【0038】

メモリモジュール 200 のエラー発生アドレスに格納されたデータにおいて、パリティチェックによってリードエラーが検出された場合は、エラー発生アドレス、データのリード回数、当該データのデータライト時のタイムスタンプ及びリードエラー検出時のタイムスタンプの各種情報がエラーレジスタ 116 に格納される。そして、タイムスタンプが付加されたデータがパリティチェックによって、リードエラーが検出された時点で、データの訂正が不可能なためシステムは停止する。そのため、ファームウェアにより専用の読み出し経路を用いてエラーレジスタ 116 から各種情報を読み出す。

【0039】

20

図 5 は、本発明の実施の形態におけるメモリコントローラの実施例を表す。図 5 において、実線はデータまたはアドレスの流れを表す。点線は制御信号の流れを表す。

【0040】

図 2 において説明したタイムスタンプ書き込み部 108 は、タイムスタンプ生成回路 118、PG (Parity Generator) 120、セクタ 602 から構成される。

【0041】

ライトデータキュー 102 はライトデータバス 302 を介して、システムコントローラ 300 からライトデータを受信する。ライトデータキュー 102 は、ライトデータの順序を保つために、システムコントローラ 300 から送信されてきた順序でライトデータをメモリモジュール 200 に送信する必要がある。そのため、ライトデータキュー 102 は、受信したデータを FIFO (First In First Out) で制御する。具体的には、ライトデータキュー 102 は、メモリモジュール 200 がライトデータの受け付けが可能であるか否かを判定する。ライトデータの受け付けが可能であった場合は、ライトデータキュー 102 は、ライトデータをメモリモジュール 200 に書き込むために、PG 120 またはセクタ 602 に送信する。

30

【0042】

PG 120 は、ライトデータキュー 102 からライトデータを受信する。そして、ライトデータに付加されている誤り訂正符号をパリティビットに置き換える。図 4 に表したように、 $n \times 8 \text{ byte}$ 、即ち、 $n \times 64 \text{ bit}$ のデータの、「1」の数の合計が偶数であれば「1」のデータを、「1」の数の合計が奇数であれば「0」のデータをパリティビットとして付加する。1 bit エラーが発生すると「1」の数の合計が変化するので、エラーを検出することができる。

40

【0043】

タイムスタンプ生成回路 118 は、システムを駆動するためのシステムクロックからクロック信号を受信する。タイムスタンプ生成回路 118 は、システムクロックが一定数経過する毎にカウントアップすることで、システム起動時からの経過時間を示す情報であるタイムスタンプを生成する。また、カウントするシステムクロックの数量は予め任意の値に設定可能であり、タイムスタンプの単位 (タイムスケール) を任意に設定可能である。タイムスタンプ生成回路 118 は、生成したタイムスタンプをライトデータに付加するた

50

めに P G 1 2 0 に送信する。

【 0 0 4 4 】

P G 1 2 0 は、タイムスタンプ生成回路 1 1 8 からタイムスタンプを受信する。そして、タイムスタンプをライトデータに付加する。さらに、P G 1 2 0 は、生成したパリティビットをライトデータに付加する。ライトするデータ形式を図 4 に表す。このように、P G 1 2 0 は、ライトデータに付加された誤り訂正符号を、タイムスタンプ 2 0 6 及びパリティビット 2 0 7 に置き換える。ここで、図 4 のタイムスタンプフィールドに書き込まれるタイムスタンプとは、例えば、システムクロックに基づいて、タイムスタンプ生成回路 1 1 8 がカウントアップした値である。

【 0 0 4 5 】

セクタ 6 0 2 は、ライトデータキュー 1 0 2 から送信されてきたデータと、P G 1 2 0 から送信されてきたデータのいずれかをメモリモジュール 2 0 0 に送信するかを選択する。コンパレータ 1 0 3 が、エラーアドレスレジスタ 1 2 8 に格納されているエラー発生アドレスとアドレスバス 3 0 6 を介して、アドレスキュー 1 0 6 が受信したアドレスとが一致するか否かを判定することで、セクタ 6 0 2 は、いずれかのデータをメモリモジュール 2 0 0 に送信するかを選択できるようになる。エラー発生アドレスと、アドレスキュー 1 0 6 が受信したアドレスとが一致しない場合、セクタ 6 0 2 は送信経路を切り替えることで、ライトデータキュー 1 0 2 は、ライトデータをセクタ 6 0 2 に送信できるようになる。一方、エラー発生アドレスと、アドレスキュー 1 0 6 が受信したアドレスとが一致する場合、セクタ 6 0 2 は送信経路を切り替えることで、ライトデータキュー 1 0 2 は、ライトデータを P G 1 2 0 に送信できるようになる。なお、コンパレータ 1 0 3 については、後述する。

【 0 0 4 6 】

図 2 において説明したデータチェック部 1 1 0 は、P C (P a r i t y C h e c k e r) 1 2 6、E C C C H K (E C C C H E C K E R) 1 2 2、E C C G E N (E C C G e n e r a t o r) 1 2 4、セクタ 6 0 4 から構成される。P C 1 2 6 は、リードデータにパリティビットが付加されている状態の時に、リードデータのパリティチェックを行う。E C C C H K 1 2 2 は、リードデータに誤り訂正符号が付加されている状態の時に、リードデータの誤り検出及び訂正を行う。E C C G E N 1 2 4 は、リードデータにパリティビットが付加されている状態の時に、誤り訂正符号を生成する。

【 0 0 4 7 】

P C 1 2 6 は、メモリデータバス 2 0 8 を介してメモリモジュール 2 0 0 からリードデータを受信する。P C 1 2 6 は、当該リードデータを受信した時点でのパリティチェックを行う。パリティチェックは、リードデータ中の「 1 」の数の合計が偶数または奇数のいずれかであるかを判定し、判定結果とパリティビットとに基づいて受信したリードデータのエラーを検出する。リードデータ中の「 1 」の数の合計が偶数であれば、パリティビットは「 1 」となっており、リードデータ中の「 1 」の数の合計が奇数であれば、パリティビットは「 0 」となっている。P C 1 2 6 は、パリティビットとリードデータ中の「 1 」の数の合計の奇偶性に基づいてリードデータのエラーを検出することができる。P C 1 2 6 は、リードデータ及びパリティチェックの結果をタイムスタンプ解析回路 1 3 2 に送信する。

【 0 0 4 8 】

E C C C H K 1 2 2 は、メモリデータバス 2 0 8 を介して、メモリモジュール 2 0 0 からリードデータを受信する。E C C C H K 1 2 2 は、受信したリードデータの誤り検出及び訂正を行う。本実施例では、8 b y t e のデータに誤り訂正符号が付加されている。当該誤り訂正符号は、8 b y t e のデータの並びを 8 b i t で定義したものである。E C C C H K 1 2 2 は、メモリモジュール 2 0 0 からメモリデータバス 2 0 8 を介して受信したリードデータに基づいて、当該リードデータを受信した時点での誤り訂正符号を生成する。次に、生成した誤り訂正符号とリードデータに付加された誤り訂正符号とを比較する。比較の結果、誤り訂正符号の値が同じであれば、データにエラーがないと判定することが

10

20

30

40

50

できる。一方、比較の結果、誤り訂正符号の値が異なると、データにエラーが発生していると判定することができる。誤り検出及び訂正を行った結果、リードエラーを検出しなかった場合は、E C C C H K 1 2 2 は、リードデータをセクタ 6 0 4 に送信する。一方、リードエラーを検出した場合は、E C C C H K 1 2 2 は、誤り訂正符号から正しいデータを算出して、データの訂正を行う。そして、リードデータが格納されていたメモリモジュール 2 0 0 のアドレスをエラーアドレスレジスタ 1 2 8 に格納するために、制御信号をエラーアドレスレジスタ 1 2 8 に送信する。エラーアドレスレジスタ 1 2 8 は、当該制御信号を受信すると、アドレスキューからリードデータが格納されていたメモリモジュール 2 0 0 のアドレスを取得する。なお、エラーアドレスレジスタ 1 2 8 については後述する。

【 0 0 4 9 】

E C C G E N 1 2 4 は、メモリデータバス 2 0 8 を介してメモリモジュール 2 0 0 からリードデータを受信する。E C C G E N 1 2 4 は、リードデータから誤り訂正符号を生成する。そして、リードデータに付加されたタイムスタンプ及びパリティビットを誤り訂正符号に置き換える。E C C G E N 1 2 4 は、誤り訂正符号を付加したリードデータをセクタ 6 0 4 に送信する。

【 0 0 5 0 】

セクタ 6 0 4 は、E C C C H K 1 2 2 から送信されてきたデータと、E C C G E N 1 2 4 から送信されてきたデータのいずれかをリードデータキュー 1 0 4 に送信するかを選択する。コンパレータ 1 0 3 が、エラーアドレスレジスタ 1 2 8 に格納されているエラー発生アドレスとアドレスバス 3 0 6 を介して、アドレスキュー 1 0 6 が受信したアドレスとが一致するか否かを判定することで、セクタ 6 0 4 は、いずれかのデータをリードデータキュー 1 0 4 に送信するかを選択できるようになる。エラー発生アドレスと、アドレスキュー 1 0 6 が受信したアドレスとが一致しない場合、セクタ 6 0 4 は、リードデータの送信経路を E C C C H K 1 2 2 側に切り替える。これによって、リードデータキュー 1 0 4 は、E C C C H K 1 2 2 が送信したリードデータを受信することができるようになる。ゆえに、リードデータに誤り訂正符号が付加されている場合は、E C C C H K 1 2 2 によって、誤り検出及び誤り訂正が行われたリードデータをリードデータキュー 1 0 4 に送信することができる。一方、エラー発生アドレスと、アドレスキュー 1 0 6 が受信したアドレスとが一致する場合、セクタ 6 0 4 は、リードデータの送信経路を E C C G E N 1 2 4 側に切り替える。これによって、リードデータキュー 1 0 4 は、E C C G E N 1 2 4 が送信したリードデータを受信することができるようになる。ゆえに、リードデータにパリティチェックビットが付加されている場合は、E C C G E N 1 2 4 によって、誤り訂正符号が付加されたリードデータをリードデータキュー 1 0 4 に送信することができる。なお、コンパレータ 1 0 3 については、後述する。

【 0 0 5 1 】

リードデータキュー 1 0 4 は、リードデータの順序を保つために、E C C C H K 1 2 2 または E C C G E N 1 2 4 から送信されてきた順序でリードデータをシステムコントローラ 3 0 0 に送信する必要がある。そのため、リードデータキュー 1 0 4 は、受信したリードデータを F I F O で制御する。具体的には、リードデータキュー 1 0 4 は、システムコントローラ 3 0 0 がリードデータの受け付けが可能であるか否かを判定する。リードデータの受け付けが可能であった場合は、リードデータキュー 1 0 4 は、リードデータをリードデータバス 3 0 4 を介してシステムコントローラ 3 0 0 に送信する。

【 0 0 5 2 】

図 2 において説明したエラーアドレスカウンタ部 1 1 4 は、エラーアドレスレジスタ 1 2 8、リードカウンタ 1 3 0、コンパレータ 1 0 3 から構成される。

【 0 0 5 3 】

アドレスキュー 1 0 6 は、システムコントローラ 3 0 0 からアドレスバス 3 0 6 を介してアドレスを受信する。そして、受信したアドレスをメモリアドレスバス 2 1 0 を介してメモリモジュール 2 0 0 に送信する。また、アドレスキュー 1 0 6 は、アドレスバス 3 0 6 から送信されてきたアドレスをコンパレータ 1 0 3 及びエラーアドレスレジスタ 1 2 8

10

20

30

40

50

に送信する。

【 0 0 5 4 】

エラーアドレスレジスタ 1 2 8 は、上述した E C C C H K 1 2 2 から制御信号を受信するとアドレスキューからリードエラーが発生したリードデータが格納されていたメモリモジュール 2 0 0 のアドレスを取得する。そして、コンパレータ 1 0 3 にエラーアドレスレジスタ 1 2 8 に格納されているエラー発生アドレスと、アドレスキュー 1 0 6 が受信したアドレスとが一致するか否かを判定させるために、エラー発生アドレスを送信する。

【 0 0 5 5 】

コンパレータ 1 0 3 は、アドレスキュー 1 0 6 から送信されてきたアドレスと、エラーレジスタ 1 2 8 から送信されてきたエラー発生アドレスを受信する。そして、両者が一致するか否かを判定する。アドレスキュー 1 0 6 から送信されてきたアドレスと、エラーアドレスとが一致した場合は、リードカウンタ 1 3 0 をカウントアップさせるために、制御信号をリードカウンタ 1 3 0 に送信する。また、タイムスタンプ解析回路 1 3 2 に、リードデータに付加されたタイムスタンプを取得させるための制御信号を送信する。なお、タイムスタンプ解析回路については後述する。

【 0 0 5 6 】

リードカウンタ 1 3 0 は、コンパレータ 1 0 3 から制御信号を受信する。制御信号を受信すると、カウントアップする。リードカウンタ 1 3 0 は、エラー発生アドレスに新たにタイムスタンプが付加されたデータがライトされてからリードエラーとなるまでのデータリード回数をカウントする。リードカウンタ 1 3 0 は、エラー発生アドレスに対してデータリードされる毎にカウントアップし、当該アドレスに対して新たにライトデータがライトされる毎にクリアする。リードカウンタ 1 3 0 は、エラー発生アドレスに新たにタイムスタンプが付加されたデータがライトされた後、当該データがリードエラーとなった場合、エラーレジスタ 1 1 6 にカウントアップしたリード回数に関するデータを送信する。

【 0 0 5 7 】

図 2 において説明したタイムスタンプ読み出し部 1 1 2 は、タイムスタンプ生成回路 1 1 8、タイムスタンプ解析回路 1 3 2 から構成される。

【 0 0 5 8 】

タイムスタンプ解析回路 1 3 2 は、P C 1 2 6 からリードデータ及びパリティチェックの結果を受信する。また、コンパレータ 1 0 3 から制御信号を受信する。そして、コンパレータ 1 0 3 から制御信号を受信し、かつ、P C 1 2 6 からリードデータのエラーを検出したことを受信した場合は、リードデータに付加されているタイムスタンプを取得する。コンパレータ 1 0 3 から制御信号を受信する場合とは、当該リードデータが格納されていたメモリモジュール 2 0 0 のアドレスと、エラー発生アドレスとが一致している場合である。これによって、タイムスタンプ生成回路 1 3 2 は、P C 1 2 6 が、エラー発生アドレスに格納されていたリードデータのエラーを検出したか否かを判定することができる。そして、タイムスタンプ解析回路 1 3 2 は、リードデータのエラーが検出された時点のタイムスタンプをタイムスタンプ生成回路 1 1 8 から取得する。さらに、リードデータから取得したタイムスタンプと、タイムスタンプ生成回路 1 1 8 から取得したタイムスタンプとをエラーレジスタ 1 1 6 に送信する。

【 0 0 5 9 】

ここで、P C 1 2 6 は、リードデータに誤り訂正符号が付加されている場合も、当該リードデータの誤り検出を行う。この場合は、リードデータにパリティチェックビットが付加されていないので、P C 1 2 6 はリードデータの誤りを検出し、タイムスタンプ解析回路 1 3 2 にはリードデータとともにエラーを検出したことを送信する。しかし、リードデータに誤り訂正符号が付加されているということは、当該リードデータはメモリモジュール 2 0 0 のエラー発生アドレスには格納されていなかったことになる。ゆえに、コンパレータ 1 0 3 から、制御信号を受信することはないため、タイムスタンプ解析回路は P C 1 2 6 がエラーを検出したことを無視することができる。

【 0 0 6 0 】

エラーレジスタ 116 は、リードカウンタ 130 から送信されてきたリード回数に関するデータを受信する。また、タイムスタンプ解析回路 132 から送信されてきたタイムスタンプを受信する。

【0061】

以下に図 6 を用いて、メモリモジュールへのライト動作の制御処理について説明する。

【0062】

ステップ S001 において、ライトデータキュー 102 は、システムコントローラ 300 からライトデータバス 302 を介してライトデータが送信されてきたか否かを判定する。ライトデータが送信されてきた場合は、処理はステップ S002 へ移行する。

10

【0063】

ステップ S002 において、エラーアドレスレジスタ 128 は、エラー発生アドレスを格納しているか否かを判定される。エラーアドレスレジスタ 128 にエラー発生アドレスが記憶されている場合は、処理はステップ S003 へ移行する。一方、エラーアドレスレジスタ 128 にエラー発生アドレスが記憶されていない場合は、処理はステップ S006 へ移行し、メモリコントローラ 100 はライトデータをメモリモジュール 200 にライトする。なお、エラー発生アドレスについては、メモリモジュール 200 のリード動作の制御処理において詳細に説明する。

【0064】

ステップ S003 において、コンパレータ 103 は、データライトアドレスとエラー発生アドレスとが一致するか否かを判定する。具体的には、エラーアドレスレジスタ 128 に格納されているエラー発生アドレスと、アドレスバス 306 を経由してアドレスキュー 106 に送信されてきたメモリアドレスとが一致するか否かを判定する。データライトアドレスとエラー発生アドレスとが一致した場合は、処理はステップ S004 へ移行する。一方、データライトアドレスとエラー発生アドレスとが一致しない場合は、処理はステップ S006 へ移行し、メモリコントローラ 100 はライトデータをメモリモジュール 200 にライトする。

20

【0065】

ステップ S004 において、PG120 は、ライトデータに基づいてパリティビットを生成する。そして、ライトデータに付加されている誤り訂正符号をパリティビットに置き換える。これによれば、後述する PC126 が、パリティチェックを行い、エラー発生アドレスにライトされたデータを正常にリードできたか否かを判定することが可能になる。処理はステップ S005 へ移行する。

30

【0066】

ステップ S005 において、タイムスタンプ生成回路 118 は、システムクロックから取得した時間情報に基づいて生成したライト時のタイムスタンプを取得する。そして、当該タイムスタンプを PG120 に送信する。PG120 は、ライトデータに付加されている誤り訂正符号をタイムスタンプに置き換える。処理はステップ S006 へ移行する。

【0067】

ステップ S006 において、セクタ 602 は、PG から送信されてきたライトデータを選択することで、ステップ S004 において生成したパリティビットと、ステップ S005 においてタイムスタンプ生成回路 118 から取得したタイムスタンプとを付加したライトデータをメモリデータバス 208 を介してメモリモジュール 200 に送信する。そして、データにタイムスタンプ及びパリティビットが付加されたデータがメモリモジュール 200 にライトされる。これによれば、後述するタイムスタンプ解析回路 132 が、エラー発生アドレスにデータが書き込まれたタイムスタンプを取得することが可能になる。さらに、タイムスタンプに基づいて、データがメモリモジュール 200 にライトされてからリードエラーが発生するまでの経過時間を取得することができる。処理は終了する。

40

【0068】

以下に図 7 を用いて、メモリモジュールへのリード動作の制御処理について説明する。

50

【 0 0 6 9 】

ステップ S 1 0 1 において、メモリコントローラ 1 0 0 は、データリードを行う。処理はステップ S 1 0 2 へ移行する。

【 0 0 7 0 】

ステップ S 1 0 2 において、エラーアドレスレジスタ 1 2 8 は、エラー発生アドレスを格納しているか否かを判定する。エラーアドレスレジスタ 1 2 8 にエラー発生アドレスが記憶されている場合は、処理はステップ S 1 0 6 へ移行する。一方、エラーアドレスレジスタ 1 2 8 にエラー発生アドレスが記憶されていない場合は、処理はステップ S 1 0 3 へ移行する。

【 0 0 7 1 】

ステップ S 1 0 3 において、E C C C H K 1 2 2 は、リードデータに付加されている誤り訂正符号に基づいてリードデータの誤り検出及び誤り訂正を行う。処理はステップ S 1 0 4 へ移行する。

【 0 0 7 2 】

ステップ S 1 0 4 において、E C C C H K 1 2 2 は、リードエラーを検出したか否かを判定する。E C C C H K 1 2 2 がリードエラーを検出した場合は、処理はステップ S 1 0 5 へ移行する。一方、E C C C H K 1 2 2 がリードエラーを検出しなかった場合は、処理は終了する。

【 0 0 7 3 】

ステップ S 1 0 5 において、E C C C H K 1 2 2 は、エラー発生アドレスをエラーアドレスレジスタ 1 2 8 に格納するために、制御信号をエラーアドレスレジスタ 1 2 8 に送信する。エラーアドレスレジスタ 1 2 8 は当該制御信号を受信すると、アドレスキュー 1 0 6 からアドレスを取得する。これによれば、図 6 のステップ S 0 0 2 及び図 7 のステップ S 1 0 2 において、リードエラーが既に発生しているか否かを判定することができる。処理は終了する。

【 0 0 7 4 】

続いて、ステップ S 1 0 2 において、エラーアドレスレジスタ 1 2 8 にエラー発生アドレスが格納されていると判定された場合の処理について説明する。

【 0 0 7 5 】

ステップ S 1 0 6 において、コンパレータ 1 0 3 は、データリードアドレスとエラー発生アドレスとが一致するか否かを判定する。具体的には、エラーアドレスレジスタ 1 2 8 に格納されているエラー発生アドレスと、アドレスバス 3 0 6 を経由してアドレスキュー 1 0 6 に送信されてきたアドレスとが一致するか否かを判定する。データリードアドレスとエラー発生アドレスとが一致した場合は、処理はステップ S 1 1 0 へ移行する。一方、データリードアドレスとエラー発生アドレスとが一致しなかった場合は、処理はステップ S 1 0 7 へ移行する。

【 0 0 7 6 】

ステップ S 1 0 7 からステップ S 1 0 9 の処理はステップ S 1 0 3 からステップ S 1 0 5 の処理と同様である。まず、ステップ S 1 0 7 において、E C C C H K 1 2 2 は、リードデータに付加された誤り訂正符号に基づいて、リードデータの誤り検出及び誤り訂正を行う。次に、ステップ S 1 0 8 において、E C C C H K 1 2 2 は、リードエラーを検出したか否かを判定する。E C C C H K 1 2 2 がリードエラーを検出した場合は、処理はステップ S 1 0 9 へ移行し、E C C C H K 1 2 2 は、エラー発生アドレスをエラーアドレスレジスタ 1 2 8 に格納するために、制御信号をエラーアドレスレジスタ 1 2 8 に送信する。一方、E C C C H K 1 2 2 は、リードデータでリードエラーを検出しなかった場合は、処理は終了する。なお、本実施例では、複数のエラー発生アドレスをエラーレジスタに格納できるとして説明した。複数のエラー発生アドレスを格納するためには、エラーアドレスレジスタ 1 2 8、コンパレータ 1 0 3 及びリードカウンタ 1 3 0 を格納するエラー発生アドレスの数に合わせて増やせば良い。また、例えば、エラーアドレスレジスタ 1 2 8、コンパレータ 1 0 3 及びリードカウンタ 1 3 0 をそれぞれ 1 つずつ配置しても良い。

10

20

30

40

50

【 0 0 7 7 】

続いて、ステップ S 1 0 6 において、データリードアドレスとエラー発生アドレスとが一致したと判定された場合の処理について説明する。

【 0 0 7 8 】

ステップ S 1 1 0 において、リードデータにタイムスタンプ及びパリティビットが付加されているか否かが判定される。リードデータにタイムスタンプ及びパリティビットが付加されている場合は、処理はステップ S 1 1 1 へ移行する。一方、リードデータにタイムスタンプ及びパリティビットが付加されていない場合は、処理はステップ S 1 0 7 へ移行する。ここで、データエラー発生アドレスに格納されていたデータにタイムスタンプ及びパリティビットが付加されていない状態とは、エラー発生アドレスがエラーアドレスレジスタ 1 2 8 に格納されてはいるが、当該エラー発生アドレスにはタイムスタンプ及びパリティビットが付加されたデータが未だ書き込まれておらず、誤り訂正符号が付加されたデータが書き込まれている状態のことである。この場合は、E C C H K 1 2 2 がリードデータに付加されている誤り訂正符号に基づいて、リードエラーの検出及びリードデータの訂正を行うため、処理はステップ S 1 0 7 へ移行する。ステップ S 1 0 7 からステップ S 1 0 9 の処理は上述したとおりなので、その説明を省略する。

10

【 0 0 7 9 】

ステップ S 1 1 1 において、P C 1 2 6 は、リードデータに付加されているパリティチェックビットに基づいてパリティチェックを行う。さらに、E C C G E N 1 2 4 は、リードデータに基づいて誤り訂正符号を生成し、リードデータに付加されているタイムスタンプ及びパリティチェックビットを誤り訂正符号に置き換える。これによって、タイムスタンプ及びパリティビットが付加されたデータを誤り訂正符号が付加されたデータに戻す。処理はステップ S 1 1 2 へ移行する。

20

【 0 0 8 0 】

ステップ S 1 1 2 において、P C 1 2 6 は、パリティチェックに基づいて、リードエラーを検出したか否かを判定する。リードエラーを検出した場合は、処理はステップ S 1 1 3 へ移行する。一方、リードエラーを検出なかった場合は、処理はステップ S 1 1 4 へ移行し、リードカウンタ 1 3 0 はリード回数をカウントアップする。これによれば、エラー発生アドレスにおいて、タイムスタンプ及びパリティビットが付加されたデータがライトされてから、リードエラーが発生するまでのデータのリード回数を取得することができる。

30

【 0 0 8 1 】

ステップ S 1 1 3 において、エラー発生アドレス、タイムスタンプ及びパリティビットが付加されたデータのリード回数、データライト時及びエラー検出時のタイムスタンプがエラーレジスタ 1 1 6 に格納される。エラー発生アドレスは、エラーアドレスレジスタ 1 2 6 から取得される。タイムスタンプが付加されたデータのリード回数は、リードカウンタ 1 3 0 から取得される。データライト時のタイムスタンプは、データに付加されたタイムスタンプからタイムスタンプ解析回路 1 3 2 によって取得される。エラー検出時のタイムスタンプは、タイムスタンプ生成回路 1 1 8 からタイムスタンプ解析回路 1 3 2 によって取得される。処理は終了する。

40

【 0 0 8 2 】

エラーレジスタ 1 1 6 に格納されたエラー発生アドレス、リード回数、データライト時のタイムスタンプ、エラー検出時のタイムスタンプは、専用のファームウェアにより読み出される。これらの情報から、エラーが発生したアドレスにおいて、データがライトされてからリードエラー発生までのリード回数を得ることができる。さらに、データライト時のタイムスタンプとエラー検出時のタイムスタンプとの差分を取ることで、エラーが発生したアドレスに、データがライトされてからリードエラー発生までの経過時間を取得することができる。ここで、タイムスタンプとは、上述したように、システムクロックに基づいて、カウントアップした値である。ファームウェアから当該カウントアップされた値を読み出し、タイムスケールに基づいて時間情報に変換することで、当該経過時間を取

50

得することができる。

【0083】

最後に本発明の有効性について説明する。本発明によれば、エラーが発生したアドレスに、データがライトされてからリードエラーが発生するまでの経過時間とリード回数からシステムの不具合の原因を推定することが可能になる。

【0084】

例えば、リードエラーが発生するまでの経過時間がメモリモジュールのリフレッシュサイクルよりも短く、かつ、リード回数が100であったとする。この場合、リード回数からデータが正常にライトされたことを保証することができる。何故なら、正常にライトされていなければ、100回も正常にリードできるとは考え難いからである。ゆえに、不具合の原因がメモリデータバスなどの伝送路やメモリモジュールのメモリセルにある確率は低いと考えられる。よって、リフレッシュサイクルが長い場合、コンデンサに電荷を蓄えることができず、データが消失してしまったと推定することができる。このような場合は、再度、リフレッシュサイクルを設定して試験を行えば良い。リフレッシュサイクルを短くして試験を行った結果、不具合が発生しなければ、リフレッシュサイクルが最適でなかったと結論付けることができる。

10

【0085】

また、例えば、リードエラー発生までの経過時間が短く、かつ、リード回数が1であったとする。この場合、1回目のリードでエラーが発生しているため、メモリモジュールのメモリセルにデータが書き込まれた時点で不具合が発生していると推定できる。ゆえに、不具合の原因がメモリデータバスなどの伝送路やメモリモジュールのメモリセルにある確率が高いと推定することができる。よって、例えば、別の試験を行うことで、メモリデータバスなどの伝送路に不具合がないことを証明できれば、メモリセルに不具合の原因がある確率が高いと推定することができる。

20

【0086】

以上の実施の形態は、本発明をより良く理解させるために具体的に説明したものであって、別形態を制限するものではない。従って、発明の趣旨を変更しない範囲で変更可能である。

【図面の簡単な説明】

【0087】

【図1】システムの全体構成図である。

【図2】本発明の実施形態の例を示す図である。

【図3】1回のメモリアクセス動作で送受信するデータ単位におけるデータ形式を表す図(その1)である。

【図4】1回のメモリアクセス動作で送受信するデータ単位におけるデータ形式を表す図(その2)である。

【図5】本発明によるメモリコントローラの実施例である。

【図6】本発明のメモリへのライト動作の制御説明図である。

【図7】本発明のメモリからのリード動作の制御説明図である。

【符号の説明】

【0088】

- 100 メモリコントローラ
- 102 ライトデータキュー
- 103 コンパレータ
- 104 リードデータキュー
- 106 アドレスキュー
- 108 タイムスタンプ書き込み部
- 110 データチェック部
- 112 タイムスタンプ読み出し部
- 114 エラーアドレスカウント部

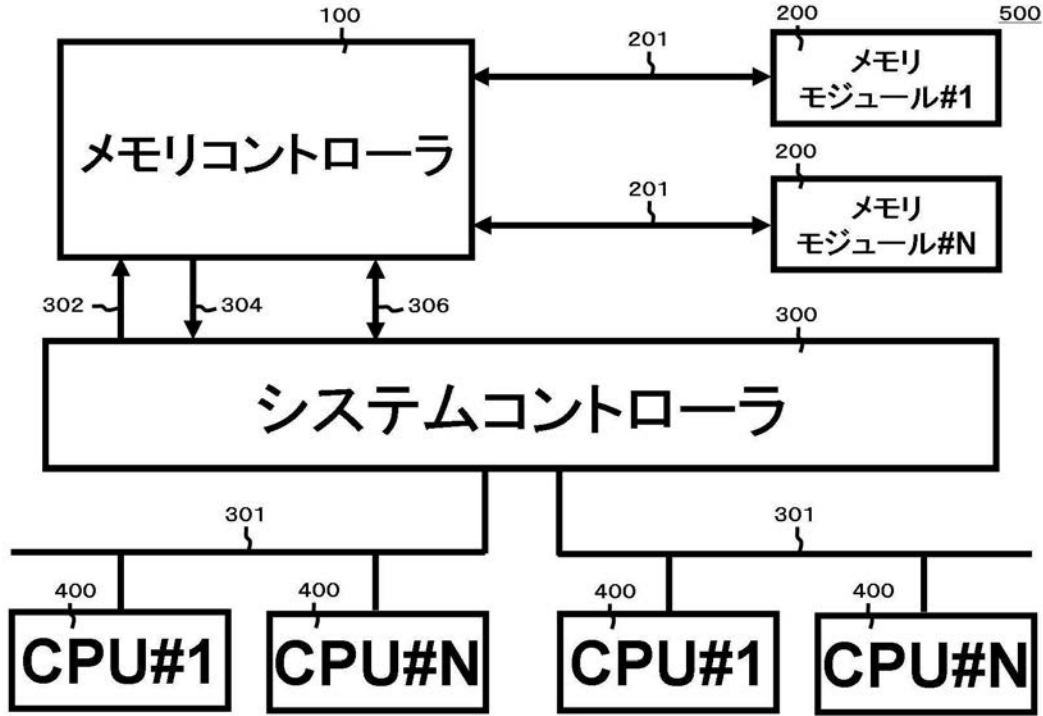
30

40

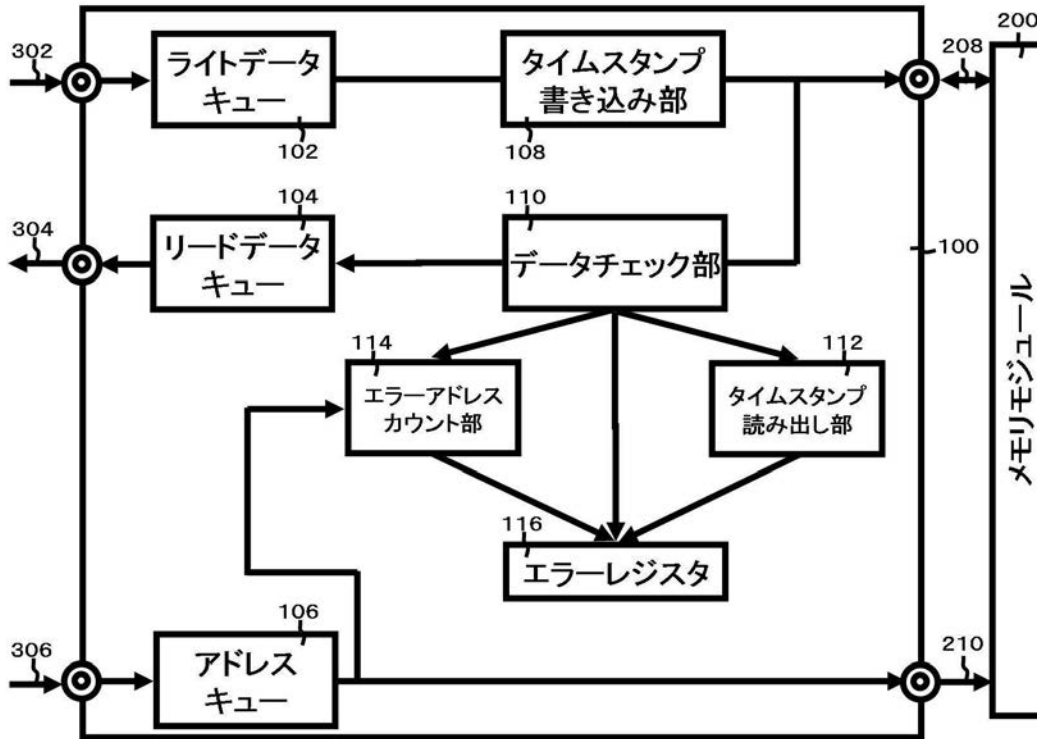
50

1 1 6	エラーレジスタ	
1 1 8	タイムスタンプ生成回路	
1 2 0	P G	
1 2 2	E C C C H K	
1 2 4	E C C G E N	
1 2 6	P C	
1 2 8	エラーアドレスレジスタ	
1 3 0	リードカウンタ	
1 3 2	タイムスタンプ解析回路	
2 0 0	メモリモジュール	10
2 0 1	メモリデータバス	
2 0 2	データ	
2 0 4	誤り訂正符号	
2 0 6	タイムスタンプフィールド	
2 0 7	パリティビット	
2 0 8	メモリデータバス	
2 1 0	メモリアドレスバス	
3 0 0	システムコントローラ	
3 0 1	C P Uデータバス	
3 0 2	ライトデータバス	20
3 0 4	リードデータバス	
3 0 6	アドレスデータバス	
4 0 0	C P U	
5 0 0	メモリシステム	
6 0 2	セレクタ	
6 0 4	セレクタ	

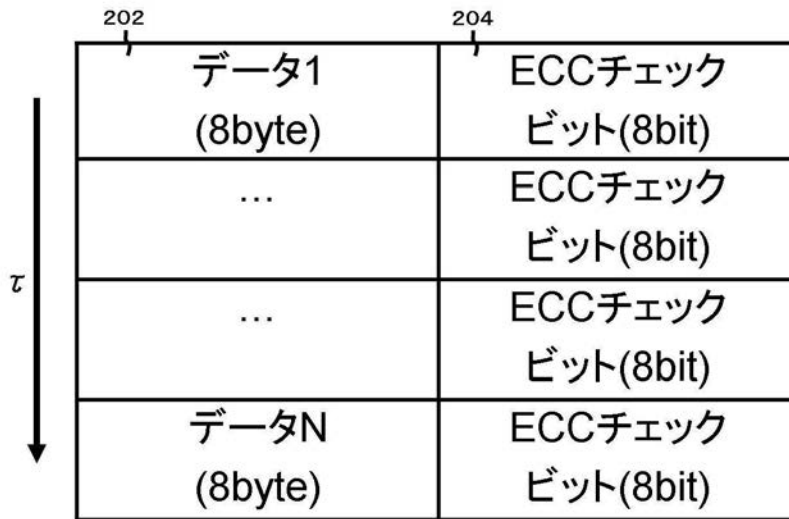
【 図 1 】



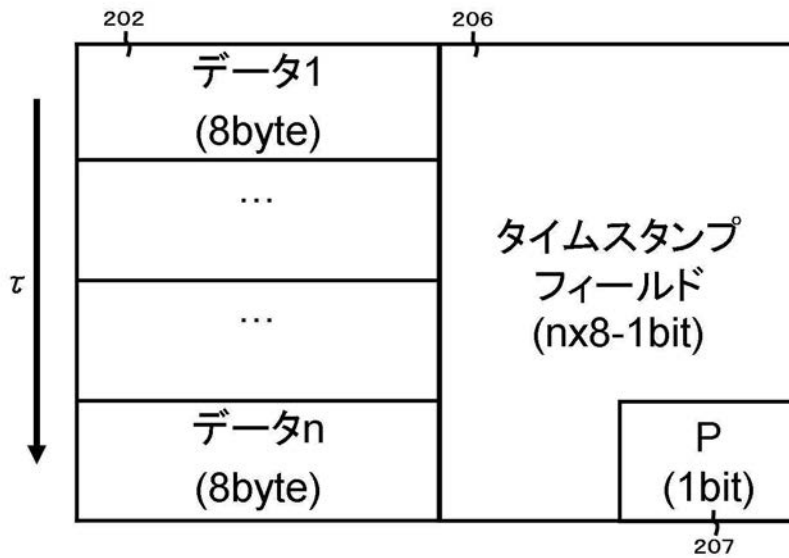
【 図 2 】



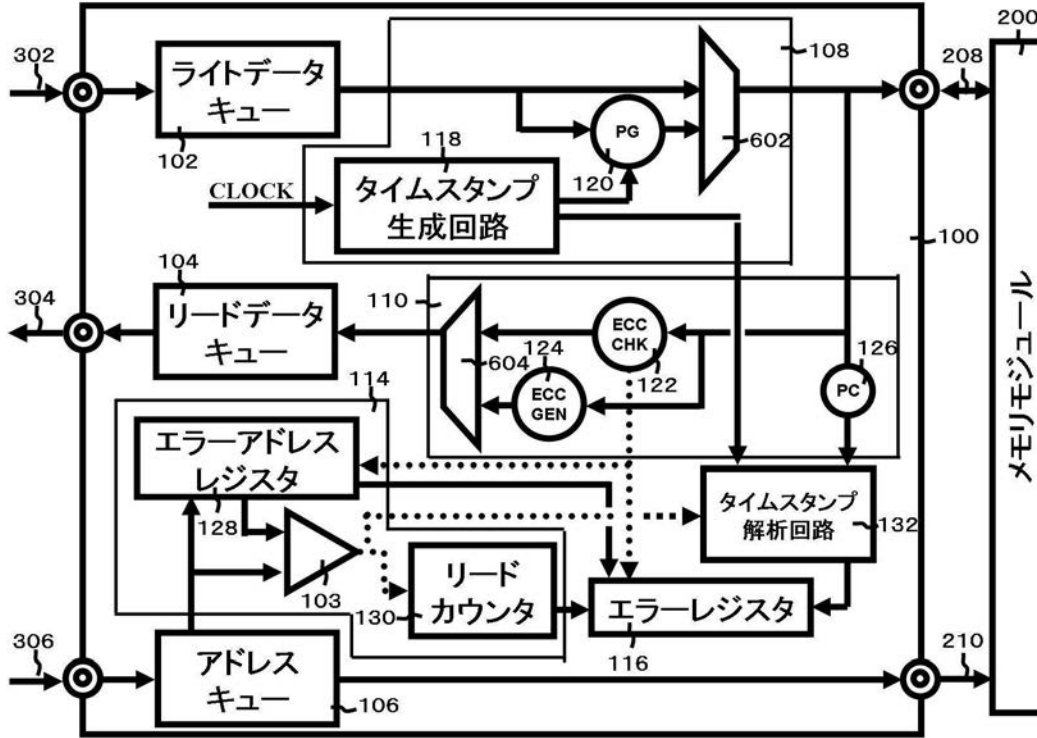
【 図 3 】



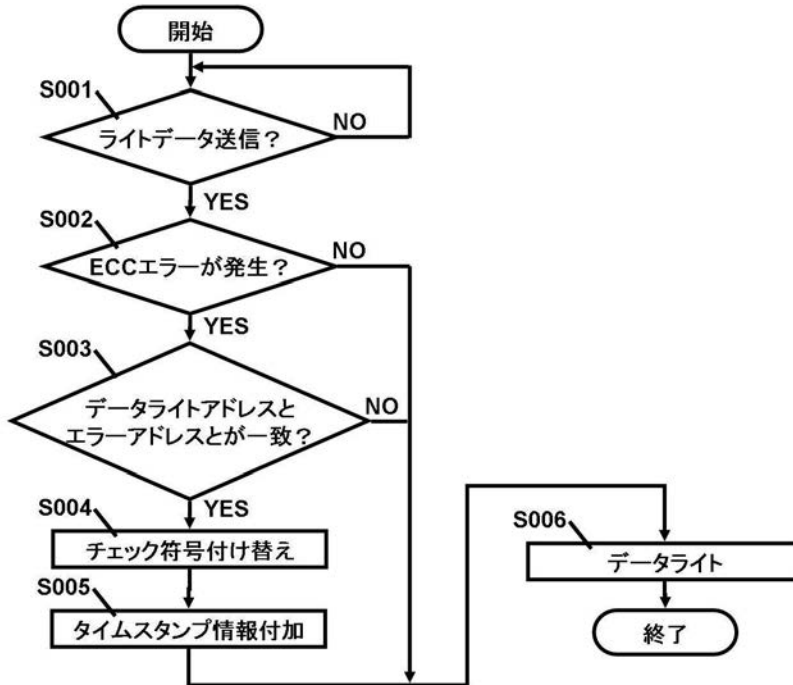
【 図 4 】



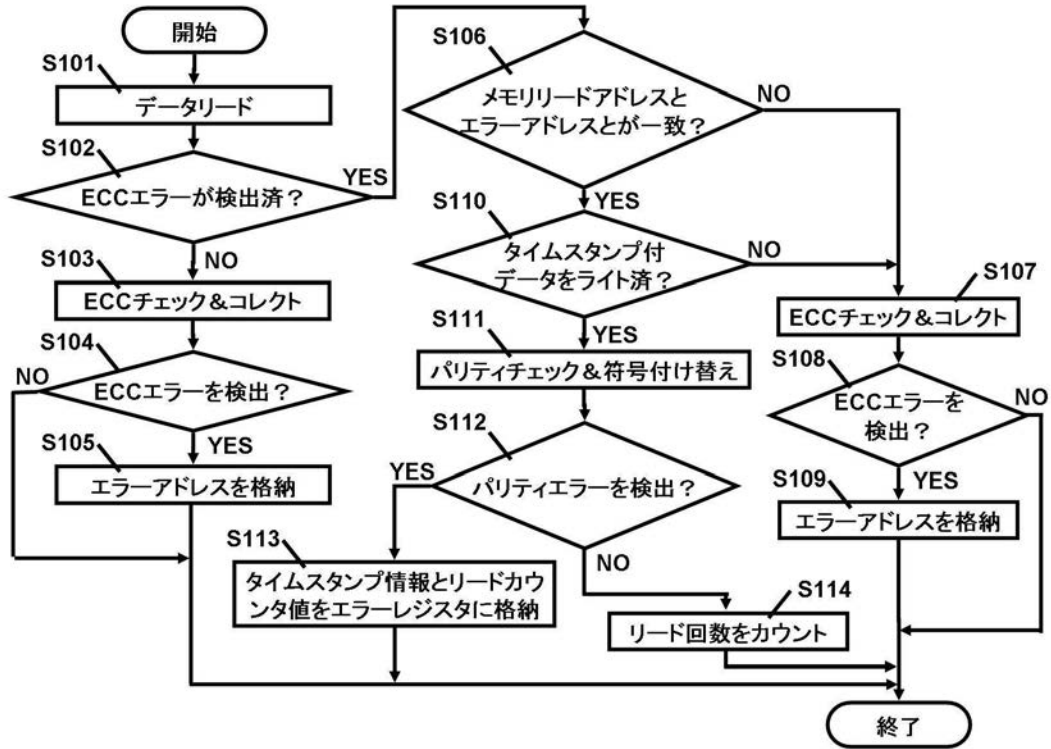
【 図 5 】



【 図 6 】



【図7】



フロントページの続き

(56)参考文献 特開平10-055320(JP,A)
特開平02-197954(JP,A)

(58)調査した分野(Int.Cl., DB名)
G06F 12/16