

【公報種別】特許法第 17 条の 2 の規定による補正の掲載

【部門区分】第 6 部門第 3 区分

【発行日】平成 19 年 12 月 20 日 (2007.12.20)

【公表番号】特表 2004-500623 (P2004-500623A)

【公表日】平成 16 年 1 月 8 日 (2004.1.8)

【年通号数】公開・登録公報 2004-001

【出願番号】特願 2001-525526 (P2001-525526)

【国際特許分類】

G 0 6 F 12/16 (2006.01)

G 0 6 F 11/10 (2006.01)

G 1 1 C 29/42 (2006.01)

【F I】

G 0 6 F 12/16 3 2 0 B

G 0 6 F 12/16 3 2 0 L

G 0 6 F 11/10 3 2 0 A

G 0 6 F 11/10 3 2 0 F

G 0 6 F 11/10 3 3 0 B

G 1 1 C 29/00 6 3 1 D

【手続補正書】

【提出日】平成 19 年 11 月 5 日 (2007.11.5)

【手続補正 1】

【補正対象書類名】明細書

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【書類名】明細書

【発明の名称】R A M モジュールにデータ語を記憶する方法と回路装置

【特許請求の範囲】

【請求項 1】 R A M モジュールにデータ語を記憶する方法において、
 少なくとも 1 つのデータ語を R A M モジュールに書き込む際に少なくとも 1 つのデータ語からチェックビット語を発生し、
 チェックビット語を記憶し、
 R A M モジュールから少なくとも 1 つのデータ語を読み出す際に前記チェックビット語を読み出し、
 読み出された少なくとも 1 つのデータ語から、チェックビット語を新たに発生し、
 読み出された前記チェックビット語を新たに発生した前記チェックビット語と比較し、
 この両チェックビット語が一致しないときに、エラーメッセージを発生することを特徴とする方法。

【請求項 2】 チェックビット語がパリティビットの決定によって発生させられることを特徴とする請求項 1 記載の方法。

【請求項 3】 各データ語から 1 つの 2 ビットパリティ語を発生し、この場合各々のデータ半語からそれぞれ 1 つのパリティビットを決定することを特徴とする請求項 2 記載の方法。

【請求項 4】 多数のデータ語から 1 つのパリティ語を発生し、このパリティ語のパリティビットをそれぞれ、すべてのデータ語の同じ桁から決定することを特徴とする請求項 1 または 2 記載の方法。

【請求項 5】 C R C 語を演算することによってチェックビット語を発生することを特徴とする請求項 1 記載の方法。

【請求項 6】 その都度多数のデータ語を 1 つのメモリ語にまとめ、このメモリ語から、対応する CRC 語を演算することを特徴とする請求項 5 記載の方法。

【請求項 7】 RAM モジュールにデータ語を記憶するための回路装置において、少なくとも 1 つのデータ語の書き込みおよび読取りの際に少なくとも 1 つのデータ語からチェックビット語を発生するための第 1 の回路ユニット (2 1) と、データ語のためのチェックビット語を割り当てて記憶するための多数のレジスタ (1 1 i , 6 1 i) と、データ語を読み取る際に、割り当てられたチェックビット語を、第 1 の回路ユニット (2 1) によって新たに発生したチェックビット語と比較し、チェックビット語が一致しないときにエラーメッセージ (F) を発生するための第 2 の回路ユニット (2 2) とを備えていることを特徴とする回路装置。

【請求項 8】 レジスタの数は、第 1 の 2 ビットパリティレジスタ (1 1 i) によって形成され、各々のデータ語に 1 つの 2 ビットパリティレジスタが割り当てられていることを特徴とする請求項 7 記載の回路装置。

【請求項 9】 レジスタの数は、CRC レジスタ (6 1 i) によって形成され、それぞれ 4 個のデータ語に 1 つの CRC レジスタが割り当てられていることを特徴とする請求項 7 記載の回路装置。

【請求項 10】 それぞれ 4 個のデータ語を 1 つのメモリ語として記憶するためのマルチプレクサ (7 1) と、1 つのメモリ語から CRC 語を演算し、割り当てられた CRC レジスタ (6 1 i , 7 4) に CRC 語を記憶するための CRC 演算装置 (7 3) とを備えていることを特徴とする請求項 9 記載の回路装置。

【請求項 11】 データ語が 3 2 ビット語であり、CRC 語が 9 ビット語であることを特徴とする請求項 10 記載の回路装置。

【請求項 12】 ビットがそれぞれすべてのデータ語の同じ桁から決定されるチェックビット語を記憶するための第 2 のレジスタ (1 2) と、第 2 のレジスタ (1 2) の内容から決定されるチェックビット語を記憶するための第 3 のレジスタ (1 3) を備えていることを特徴とする請求項 7 ~ 11 のいずれか一つに記載の回路装置。

【発明の詳細な説明】

【0001】

本発明は、特に安全上重要な用途のための、RAM モジュールにデータ語を記憶するための方法と回路装置に関する。

【0002】

RAM (随時書き込み読み出しメモリ) モジュールは一般的に知られており、広く普及している。RAM は多数の用途のためのデータを繰り返し記憶および読み出す役目を果たす。RAM モジュールに記憶されたデータの安全性は、メモリアーキテクチャを設計する際に特別な注意を払う必要がある。公知のアーキテクチャの場合、データ安全性はモジュールの完全に冗長的な設計によって比較的に確実に保証される。しかし、この場合回路コストとシリコン接合の必要性が比較的に高いという重要な欠点がある。

【0003】

そこで、本発明の根底をなす課題は、データ安全性に関する制約を受け入れることなく、接合の要求が大幅に少ない、RAM モジュールにデータ語を記憶するための方法と回路装置を提供することである。

(1) 本発明の第 1 の構成によれば、RAM モジュールにデータ語を記憶する方法において、少なくとも 1 つのデータ語を RAM モジュールに書き込む際に少なくとも 1 つのデータ語からチェックビット語を発生し、チェックビット語を記憶し、RAM モジュールから少なくとも 1 つのデータ語を読み出す際に前記チェックビット語を読み出し、読み出された少なくとも 1 つのデータ語から、チェックビット語を新たに発生し、読み出された前記チェックビット語を新たに発生した前記チェックビット語と比較し、この両チェックビット語が一致しないときに、エラーメッセージを発生することを特徴とする方法が得られる。

(2) 本発明の第 2 の構成によれば、チェックビット語がパリティビットの決定によっ

て発生させられることを特徴とする(1)記載の方法が得られる。

(3) 本発明の第3の構成によれば、各データ語から1つの2ビットパリティ語を発生し、この場合各々のデータ半語からそれぞれ1つのパリティビットを決定することを特徴とする(2)記載の方法が得られる。

(4) 本発明の第4の構成によれば、多数のデータ語から1つのパリティ語を発生し、このパリティ語のパリティビットをそれぞれ、すべてのデータ語の同じ桁から決定することを特徴とする(1)または(2)記載の方法が得られる。

(5) 本発明の第5の構成によれば、CRC語を演算することによってチェックビット語を発生することを特徴とする(1)記載の方法が得られる。

(6) 本発明の第6の構成によれば、その都度多数のデータ語を1つのメモリ語にまとめ、このメモリ語から、対応するCRC語を演算することを特徴とする(5)記載の方法が得られる。

(7) 本発明の第7の構成によれば、RAMモジュールにデータ語を記憶するための回路装置において、少なくとも1つのデータ語の書き込みおよび読取りの際に少なくとも1つのデータ語からチェックビット語を発生するための第1の回路ユニット(21)と、データ語のためのチェックビット語を割り当てて記憶するための多数のレジスタ(11i, 61i)と、データ語を読み取る際に、割り当てられたチェックビット語を、第1の回路ユニット(21)によって新たに発生したチェックビット語と比較し、チェックビット語が一致しないときにエラーメッセージ(F)を発生するための第2の回路ユニット(22)とを備えていることを特徴とする回路装置が得られる。

(8) 本発明の第8の構成によれば、レジスタの数は、第1の2ビットパリティレジスタ(11i)によって形成され、各々のデータ語に1つの2ビットパリティレジスタが割り当てられていることを特徴とする(7)記載の回路装置が得られる。

(9) 本発明の第9の構成によれば、レジスタの数は、CRCレジスタ(61i)によって形成され、それぞれ4個のデータ語に1つのCRCレジスタが割り当てられていることを特徴とする(7)記載の回路装置が得られる。

(10) 本発明の第10の構成によれば、それぞれ4個のデータ語を1つのメモリ語として記憶するためのマルチプレクサ(71)と、1つのメモリ語からCRC語を演算し、割り当てられたCRCレジスタ(61i, 74)にCRC語を記憶するためのCRC演算装置(73)とを備えていることを特徴とする(9)記載の回路装置が得られる。

(11) 本発明の第11の構成によれば、データ語が32ビット語であり、CRC語が9ビット語であることを特徴とする(10)記載の回路装置が得られる。

(12) 本発明の第12の構成によれば、ビットがそれぞれすべてのデータ語の同じ桁から決定されるチェックビット語を記憶するための第2のレジスタ(12)と、第2のレジスタ(12)の内容から決定されるチェックビット語を記憶するための第3のレジスタ(13)を備えていることを特徴とする(7)~(11)のいずれか一つに記載の回路装置が得られる。

【0004】

上述の課題は(1)記載の方法に従い、少なくとも1つのデータ語をRAMモジュールに書き込む際に少なくとも1つのデータ語からチェックビット語を発生し、チェックビット語を記憶し、RAMモジュールから少なくとも1つのデータ語を読み出す際に前記チェックビット語を読み出し、読み出された少なくとも1つのデータ語から、チェックビット語を新たに発生し、読み出された前記チェックビット語を新たに発生した前記チェックビット語と比較し、この両チェックビット語が一致しないときに、エラーメッセージを発生することによって解決される。

【0005】

上述した課題は更に、(7)記載の装置に従い、少なくとも1つのデータ語の書き込みおよび読取りの際に少なくとも1つのデータ語からチェックビット語を発生するための第1の回路ユニットと、データ語のためのチェックビット語を割り当てて記憶するための多数のレジスタと、データ語を読み取る際に、割り当てられたチェックビット語を、第1の

回路ユニットによって新たに発生したチェックビット語と比較し、チェックビット語が一致しないときにエラーメッセージを発生するための第2の回路ユニットとを備えていることによって解決される。

【0006】

この解決策の特別な利点は、冒頭に述べた完全に冗長的な設計の場合とほぼ同じデータ信頼性で、必要なシリコン接合、ひいては回路設計努力およびコストが大幅に低減されることにある。

【0007】

第1及び7以外の構成（従属請求項）は本発明の有利な実施例に対応している。

【0008】

本発明の他の詳細、特徴および効果は、図に基づく好ましい実施の形態の次の説明から明らかになる。

【0009】

RAMモジュールは図1に従って、多数の32ビットのデータ語レジスタ10a・・・10i・・・10xからなる語指向のアレイ10を含んでいる。このデータ語は互いに列をなして配置して示してある。各々のデータ語レジスタには2ビットのパリティ語レジスタ11a・・・11i・・・11xが割り当てられている。それによって、2ビットのパリティアレイ11が生じる。更に、32ビットのパリティ語レジスタ12が設けられ、このパリティ語レジスタには2ビットのパリティ語レジスタ13が割り当てられている。

【0010】

データ交換のために、この構造体は公知のごとくバスインターフェースユニット14に接続されている。このバスインターフェースユニットを介してCPUバスとの接続を行うことができる。バスインターフェースユニット14は更に、図2、3に示した書込み操作および読取り操作時にパリティ語を発生および比較するための回路ユニットを備えている。

【0011】

RAMモジュールに書き込むために、図2に従って、当該のデータ語が32ビットデータバス20から第1の回路ユニット21に送られる。この第1の回路ユニットによって、各々のデータ語について2ビットのパリティ語が発生させられる。続いて、データ語はRAMモジュール内のデータ語レジスタ10iに書き込まれ、2ビットパリティ語は関連する2ビットパリティ語レジスタ11iに書き込まれる。

【0012】

RAMモジュールからデータ語を読み取るために、図3に従って、アドレス指定されたデータ語が先ず最初に第1の回路ユニット21に送られる。同時に、関連する2ビットパリティ語が第2の回路ユニット22に伝送される。第1の回路ユニット21では、読み出されたデータ語から2ビットパリティ語が発生させられる。このパリティ語は第2の回路ユニット22に送られ、そこでRAMモジュールから直接読み出された2ビットパリティ語と比較される。この両パリティ語が一致していないと、エラー信号Fが発生させられるかまたは適当なエラーフラグがセットされる。2ビットパリティ語が一致すると、読み出されたデータ語がデータバス20に伝送される。

【0013】

図4では、各32ビットデータ語が第1と第2の16ビット半語HWからなっている。この場合、各半語から2ビットパリティ語のビットBが発生させられる。

【0014】

これらの語指向のパリティを自動的に発生および自動的に比較することにより、RAMモジュールからの読み出し時に、個々のビット誤りを直ちに“オンライン”で検出することができる。

【0015】

一層高いエラー防止性を達成するために、2ビットパリティ発生は、各々のデータ語について多項式で計算されたCRC（周期的冗長検査）語を有するCRCチェックによって

置き換えることができる。データ語の長さとはCRC語の長さの適切な比を達成するために、メモリアーキテクチャは、記憶されたデータ語（メモリ語）の長さがデータバスにおけるデータ語の長さの数倍になるように選定される。32ビットのデータ語長さの場合、メモリ語は好ましくは128ビットの長さを有し、最適なエラー防止のためのCRC語は9ビットの長さを有する。

【0016】

図5はバスインターフェースユニット14を介して32ビットデータバス（図示していない）に接続された然るべき構造体を示している。

【0017】

RAMモジュールは互いに列状に配置して示した多数の128ビットメモリ語レジスタ60a・・・60xからなるアレイ60を備えている。各々のメモリ語レジスタには、例えばそれぞれ9ビットを有するCRCレジスタ61a・・・61xが割り当てられている。それによって、CRCアレイ61が生じる。

【0018】

アレイ60とバスインターフェースユニット14の間には、ユニット70が接続配置されている。このユニットはそれぞれ4つの32ビットデータ語のためのマルチプレクサ71と、4つの32ビットデータ語を収容するためのCRC演算レジスタ72を備えている。ユニット70は更に、CRC演算装置73を備えている。このCRC演算装置によって、9ビットCRC語が128ビットCRC演算レジスタ72から公知の計算方法で計算され、バスインターフェースユニット14に接続された9ビットCRCレジスタ74に一時記憶される。

【0019】

書込みと読取り操作は図2，3に示した方法とほぼ同じ方法で行われる。

【0020】

RAMモジュールに書き込む際にその都度、バスインターフェースユニット14を経て供給された4つの32ビットデータ語がマルチプレクサ71によって、128ビットCRC演算レジスタ72に周期的に順々に記憶されるので、128ビットメモリ語が生じる。これから、CRC演算装置73によって9ビットCRC語が計算され、9ビットCRCレジスタ74に記録される。続いて、128ビットCRC演算レジスタ72の内容がRAMアレイの128ビットメモリ語レジスタ60iの一つに記憶され、9ビットCRCレジスタ74の内容が関連する9ビットCRC語レジスタ61iに記憶される。

【0021】

新しい32ビットデータ語（または小さな語単位）をRAMモジュールに書き込む際に、当該の128ビットメモリ語レジスタ60iのCRC語を新たに計算する必要がある。これは、新しいデータ語を書き込む前に先ず最初に当該の128ビットメモリ語レジスタ60iの内容を完全に読み出し、CRC演算レジスタ72に記憶しなければならないことを意味する。それによって、CRC演算装置73によって新しいデータ語に基づいて、9ビットCRC語が新たに演算され、CRCレジスタ74に記憶される。両レジスタ72，74の内容は対応するレジスタ60i，61iに伝送される。

【0022】

新しい32ビットデータ語を書き込む前に、例えばソフトウェアによって所定の時間間隔でトリガされるエラーチェックを行うべきである場合に、上述のように先ず最初に、当該の128ビットメモリ語レジスタ60iの内容と、対応するCRCレジスタ61iの内容が読み出される。続いて、CRC演算装置73によって新たに9ビットCRC語が発生させられ、読み出されたCRC語と比較される。この両CRC語が一致しないと、エラー信号F（または対応すエラーフラグ）が発生させられる。CRC語が一致すると、上述のように、新しい32ビットデータ語を含む128ビットメモリ語から、新しい9ビットCRC語が演算され、両者が対応する128ビットメモリ語レジスタ60iまたはRAMモジュールの対応する9ビットCRCレジスタ61iに読み込まれる。

【0023】

エラーチェックは、データバス 20 の R A M モジュールからデータ語を読み出すべきときにも行うことができる。そのために、当該のデータ語を含むメモリ語レジスタ 60 i の内容は C R C 演算レジスタ 72 に伝送され、それから新たに C R C 語が演算計算される。この C R C 語は対応する C R C 語レジスタ 61 i に記憶された C R C 語と比較される。両語が一致しないと、エラーメッセージ F が発生させられるかあるいは然るべきエラーフラグがセットされる。両語が一致すると、読み出された 32 ビットデータ語がデータバス 20 に伝送される。続いて、C R C 演算レジスタ 72 の内容が対応する 128 ビットメモリ語レジスタ 60 i に戻される。

【0024】

図 6 は 32 ビットデータ語のための複数のメモリ語レジスタ 10 a , 10 b . . . 10 x と、32 ビットパリティ語レジスタ 12 を示している。この場合、各々の桁について、例示的に値 0 または 1 を有するビットが示してある。

【0025】

図 4 , 5 に示した語指向のチェックビット発生と異なり、図 6 では、カラム指向のパリティが発生させられる。このパリティの場合、すべてのデータ語のそれぞれ同じ桁について 1 つのパリティビットが決定される。このパリティビットは 32 ビットパリティ語レジスタ 12 の関連する桁に書き込まれる。これにより、32 ビットパリティ語が生じる。更に、この 32 ビットパリティ語について、語指向のパリティに関する図 4 に基づく説明と同様に、2 ビットパリティ語を発生し、2 ビットパリティ語レジスタ 13 (図 1 参照) に記憶することができる。前述と同様に更に、図 5 の実施の形態の場合カラム指向のパリティチェックを 128 ビットの長さのデータ語によって行うことができる。

【0026】

R A M モジュールの 1 つの語レジスタ 10 i への新しいデータ語の書込み中、先ず最初に R A M モジュール内の書き込むべきメモリ桁のデータ語の内容、すなわち例えば 32 ビットデータ語レジスタ 10 i と 32 ビットパリティ語レジスタ 12 の内容が読み出される。続いて、カラム指向の 32 ビットパリティ語が新たに決定されて書き込まれる。

【0027】

それに続いて、新しいデータ語が対応するデータ語レジスタ 10 i に再書込みされ、32 ビットパリティ語レジスタ 12 の内容が新たに決定される。続いて、32 ビットパリティ語に関して再び 2 ビットパリティを発生し、2 ビットパリティ語レジスタ 13 (図 1 参照) に記憶することができる。

【0028】

普通の読み取り操作の間、好ましくはエラーチェックは行われなない。付加的なエラーチェックは、前述のように例えば読取り操作の間の時点で、すべてのデータ語レジスタ 10 i の内容を読み出し、カラム指向の 32 ビットパリティ語を新たに発生し、そしてパリティ語レジスタ 12 に記憶されたパリティ語と比較することによって行うことができる。パリティ語が一致しないと、エラーメッセージ F が発生させられるかあるいは適当なエラーフラグがセットされる。パリティ語が一致すると、読み出されたデータ語がデータバス 20 に伝送される。ここで説明した、R A M 全体のカラム指向のエラーチェックについての実施の形態は、書込み操作または読取り操作時には適切ではなく、所定の時間の時間間隔をおいて行われる。この時間間隔は使用されるソフトウェアによって設定することができる。このエラーチェックを行うか否かの決定は好ましくは使用されるソフトウェアによって行われる。

【0029】

32 ビットパリティ語の 2 ビットパリティ語はデータ語の 2 ビットパリティ語について図 2 ~ 4 に基づいて説明した方法と同じ方法で、エラーチェックを行うために使用可能である。

【0030】

カラム指向のパリティの代わりに、カラム指向の C R C (周期的冗長検査) サムを求め、エラーチェックのために使用することができる。この場合にも、語の書込みおよびまた

は読み取りの前に、先ず最初にすべてのデータ語レジスタ 10 i とチェックビットレジスタ 12 の内容が読み出され、新たに CRC 語が決定される。この CRC 語が記憶された CRC 語と一致しないときには、エラーメッセージ F が発生させられるかあるいは対応するエラーフラグがセットされる。両 CRC 語が一致すると、書込み操作または読み取り操作が、カラム指向のパリティ語発生について上述したように決定される。

【 0 0 3 1 】

カラム指向のパリティと周期的に行われるパリティチェックまたは CRC チェックサムと周期的な CRC 演算によって更に、アドレスデコード内のエラーと二重ビットエラーと他のエラーを検出することができる。チェックまたは演算は好ましくは適当なソフトウェアによって行われる。

【図面の簡単な説明】

【図 1】

第 1 のメモリアーキテクチャを概略的に示す図である。

【図 2】

書込み操作の過程を概略的に示す図である。

【図 3】

読み取り操作の過程を概略的に示す図である。

【図 4】

語指向のパリティの発生を示す図である。

【図 5】

第 2 のメモリアーキテクチャを概略的に示す図である。

【図 6】

カラム指向のパリティの発生を示す図である。