(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2012-104064
(P2012-104064A)
(43)公開日　平成24年5月31日(2012.5.31)

(51)Int.Cl.	F I	テーマコード（参考）
G06F 12/16 (2006.01)	G06F 12/16 330C	5B018

審査請求 未請求　請求項の数 8　ＯＬ　（全 27 頁）

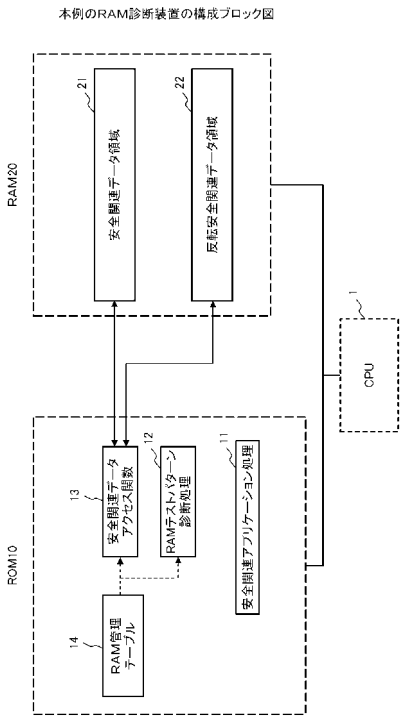
(21)出願番号　特願2010-254250 (P2010-254250)
(22)出願日　平成22年11月12日 (2010.11.12)

(71)出願人　000005234
富士電機株式会社
神奈川県川崎市川崎区田辺新田１番１号
(74)代理人　100074099
弁理士　大菅　義之
(72)発明者　西田　廣治
神奈川県川崎市川崎区田辺新田１番１号
富士電機ホールディングス株式会社内

Ｆターム(参考)　5B018 GA03 HA01 JA12 NA01 QA04 QA16

(54)【発明の名称】ＲＡＭ故障診断装置、そのプログラム

(57)【要約】
【課題】ＲＡＭ故障を確実にかつ早期に検出する。
【解決手段】安全関連アプリケーション処理１１による安全関連データ領域２１内の安全関連データへのアクセス時に、安全関連データアクセス関数１３によって当該安全関連データの格納領域に関してダブルＲＡＭ方式のＲＡＭ診断を行う。これに加えて、ＲＡＭテストパターン診断処理１２によって、定期的に、各安全関連データの格納領域等に対して、複数のテストパターンを用いたＲＡＭ診断を行う。
【選択図】図１



【特許請求の範囲】
【請求項１】
１以上の安全関連データを格納する安全関連データ格納領域を有するＲＡＭの故障診断を行うＲＡＭ診断装置であって、
任意の前記安全関連データに関するアクセス処理の際に、ダブルＲＡＭ方式によるＲＡＭ診断処理を行うダブルＲＡＭ方式診断手段と、
定期的に、少なくとも前記各安全関連データの格納領域を含む各診断対象記憶領域毎に、その診断対象記憶領域内の各アドレスを順次診断対象アドレスとして、該各診断対象アドレス毎に、複数のテストデータの書き込みと、該診断対象アドレスからのデータ読み出し及びテストデータとの一致確認を行うことによって、診断対象アドレスの診断を行うテストパターン方式診断手段と、
を有することを特徴とするＲＡＭ故障診断装置。
【請求項２】
１以上の安全関連データを格納する安全関連データ格納領域を有するＲＡＭの故障診断を行うＲＡＭ診断装置であって、
任意の前記安全関連データに関するアクセス処理の際に、ダブルＲＡＭ方式によるＲＡＭ診断処理を行うダブルＲＡＭ方式診断手段と、
定期的に、少なくとも前記各安全関連データの格納領域を含む各診断対象記憶領域毎に、その診断対象記憶領域内の各アドレスを順次診断対象アドレスとして、該各診断対象アドレスへの診断処理実行毎に、該診断対象アドレスを含む３以上の連続するアドレスに対して複数のテストデータを順次書き込むこと、該診断対象アドレスからデータ読み出すことによって、診断対象アドレスの診断を行うテストパターン方式診断手段と、
を有することを特徴とするＲＡＭ故障診断装置。
【請求項３】
前記複数のテストデータは、前記診断対象アドレスの全ビットをオン・オフ実施させる複数のテストデータ、または／且つ、ビットオン・オフパターンが相互に異なる複数のテストデータであることを特徴とする請求項２記載のＲＡＭ故障診断装置。
【請求項４】
前記診断対象アドレスを含む３以上の連続するアドレスは、該診断対象アドレスとその直前のアドレスと直後のアドレスとの３つのアドレスであり、
前記テストパターン方式診断手段は、前記診断対象アドレスに書き込むテストデータと、前記直前と直後のアドレスに書き込むテストデータとの組み合わせを順次変えながら、該診断対象アドレスからデータを読み出してデータ変化の有無を確認することで、診断対象アドレスの診断を行うことを特徴とする請求項２または３記載のＲＡＭ故障診断装置。
【請求項５】
前記テストパターン方式診断手段は、前記各診断対象アドレスへの診断処理実行毎に、前記診断対象アドレスを含む３以上の連続するアドレスの格納データをレジスタに退避させてから、前記複数のテストデータによる該診断対象アドレスの診断処理を行うことを特徴とする請求項２～４の何れかに記載のＲＡＭ故障診断装置。
【請求項６】
前記テストパターン方式診断手段による診断処理は、前記任意の前記安全関連データに対するアクセス処理を含む安全関連アプリケーション処理よりもタスクレベルが低く設定されており、前記安全関連アプリケーション処理を実行する場合には、前記テストパターン方式診断手段による診断処理は一時中断されることを特徴とする請求項２記載のＲＡＭ故障診断装置。
【請求項７】
前記各安全関連データの格納領域を示すアドレス情報が予め記憶されたアドレス情報記憶手段を更に有し、
前記ダブルＲＡＭ方式診断手段は、前記任意の前記安全関連データに関するアクセス処理に係るパラメータを受け取ると、該パラメータのアドレス情報が上記アドレス情報記憶

手段に記憶されているアドレス情報と一致するか否かを確認し、一致する場合に前記ダブルＲＡＭアルゴリズムによるＲＡＭ診断処理を行うことを特徴とする請求項２記載のＲＡＭ故障診断装置。

【請求項８】

１以上の安全関連データを格納する安全関連データ格納領域を有するＲＡＭの故障診断を行うＲＡＭ診断装置のコンピュータを、

任意の前記安全関連データに関するアクセス処理の際に、ダブルＲＡＭ方式によるＲＡＭ診断処理を行うダブルＲＡＭ方式診断手段と、

定期的に、少なくとも前記各安全関連データの格納領域を含む各診断対象記憶領域毎に、その診断対象記憶領域内の各アドレスを順次診断対象アドレスとして、該各診断対象アドレスへの診断処理実行毎に、該診断対象アドレスを含む３以上の連続するアドレスに対して複数のテストデータを書き込むこと、診断対象アドレスからデータ読み出すことによって、診断対象アドレスの診断を行うテストパターン方式診断手段、

として機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、ＲＡＭの故障診断技術に関する。

【背景技術】

【０００２】

ＲＡＭの故障の種類としては、以下に列挙するものがあり、その対策として各種のＲＡＭ診断手法がとられてきた。例えば、非特許文献１の第二分冊附属書Ａの表１には、以下に列挙するＲＡＭ故障が記載されている。

・データ／アドレスの直流化（固定故障(固着)、固定オープン（断線）、高インピーダンス、信号線間の短絡（ショート）・・・表１の最後に記載のＮＯＴＥ３参照

・メモリセル間の干渉

・アドレス化なし、アドレス化誤り、マルチアドレス化

また、非特許文献２ではソフトエラーによるデータ変化の影響が述べられている。ソフトエラーとはα線、中性子線、ＥＭＩノイズ、内部クロストークによる一時的なエラーである。

【０００３】

尚、ソフトエラーに関しては、一般的に、このエラーが生じても直ちに故障となるものではないことや、テストデータを用いた定期的なＲＡＭ診断方式では、テストデータを書いた後、直ぐに読み出すため、この極短時間の間にデータ変化するような場合にしかエラー検出できない為、ソフトエラーに関してはエラー検出率が下がること等が知られている。また、そもそも、テストデータを用いた定期的なＲＡＭ診断方式では、元々の格納データに対するチェックは行わないので（診断中は退避領域に退避させておく）、元々の格納データがソフトエラーによって変化していたとしても、これを検出することはできない。

【０００４】

特許文献１には、ＲＡＭテストパターンによる診断が開示されている。

非特許文献１にはＲＡＭ診断の各種のアルゴリズムが記載されている。

また、従来より、ダブルＲＡＭアルゴリズム（Double ＲＡＭ with hardware or software comparison and read/write test）と呼ばれる診断手法が知られている。ダブルＲＡＭアルゴリズムの場合、テストデータを用いるのではなく、実際のデータ・リード／ライトの際にＲＡＭ診断を行う。よって、基本的に、アクセスタイミングは任意のタイミングであり、アクセス先も任意であり、またリード・ライトするデータも任意のデータである(定周期で所定のテストパターンをリード／ライトするものとは異なる)。

【０００５】

このダブルＲＡＭアルゴリズムによるＲＡＭ診断処理では、ＲＡＭに書き込むべきデー

タ（書込データ）を指定アドレスに書き込む際、この書込データの反転データを反転アドレス（指定アドレスを反転したアドレス）に書き込む。そして、ＲＡＭからのデータ読み出し時等に、指定アドレスからデータを読み出すと共に、その反転アドレスから反転データを読出し、この反転データを反転させたものが上記指定アドレスのデータと一致するか否かを確認することで、ＲＡＭ故障を検出するものである。

【０００６】

尚、ダブルＲＡＭアルゴリズムの場合、データ書き込み時点から当該データの読出し時点までの、上記極短時間に比べれば長い時間の間に、ソフトエラーが生じれば、エラー検出できるので、ソフトエラーに関するエラー検出率は高い。

【先行技術文献】

【特許文献】

【０００７】

【特許文献１】特開２００７－１４８５３６号公報

【非特許文献】

【０００８】

【非特許文献１】ＪＩＳ　Ｃ　０５０８　電気・電子・プログラマブル電子安全関連系の機能安全

【非特許文献２】「ソフトエラーが引き起こすシステム信頼性への影響」http://ednjapan.rbi-j.com/content/issue/2005/02/feature/feature02.html

【発明の概要】

【発明が解決しようとする課題】

【０００９】

特にプログラマブルコントローラ等の制御分野におけるコントローラ本体（ＣＰＵ、ＲＡＭ等を有する）においては、ＲＡＭには制御用データ等が格納されるので、ＲＡＭに異常が発生した場合、異常な制御が行われる可能性がある。もし、モータやプレス機等が異常動作した場合、その近くにいる作業員等に人的被害が発生する可能性がある。この為、コントローラ本体は、故障発生確率を減少させる為、自己診断（ＲＡＭ診断）をある程度頻繁に行うことが望ましい。

【００１０】

上記ダブルＲＡＭアルゴリズムによるＲＡＭ診断は、各種故障に対応しているが、エラー検出率が低くなる場合もある。例えば、非特許文献１の第二分冊附属書Ａ表１および非特許文献２に記載されている各種故障のうち例えば「データ／アドレスの直流化」の検出に関して、例えば‘０’に固定（固着）したセルに対し‘０’のデータを書き込む場合には当該「‘０’固着異常」を検出できない(‘０’以外のデータ（‘１’）を書くまでは「‘０’固着異常」を検出できない)ことになる。

【００１１】

ダブルＲＡＭアルゴリズムによるＲＡＭ診断では、上記の通りデータリードライトに伴って故障診断する為、セルに書き込むデータは、データ内容次第である為、コントロール出来ない。この為、上記‘０’に固定（固着）したセルに対して長期間ずっと‘０’のデータを書き込み続ける事態も起こり得ることになり、長期間ずっと「‘０’固着異常」を検出できないことになる。尚、「‘１’固着異常」の場合も略同様である。

【００１２】

この様に、ダブルＲＡＭアルゴリズムによるＲＡＭ診断であっても、診断を完全に担保できないという課題がある。

特許文献１には、所定のテストデータのライト・リードを行い、読出しデータがテストデータと一致することを確認する方式（以降ＲＡＭテストパターン方式と呼ぶ）による診断が記載されているが、ＲＡＭテストパターン方式による診断では、上記の通りソフトエラーを検出し難く、例えばオンラインの処理タイミングに発生したソフトエラーを検出できない可能性が高い。

【００１３】

非特許文献１にはＲＡＭ診断の各種のアルゴリズムが記載されている。そのなかで、高い自己診断率を実現できるアルゴリズムとして上記ダブルＲＡＭアルゴリズム（Double ＲＡＭ with hardware software comparison and read/write test）がある。

【００１４】

ここで自己診断率は機能安全規格IEC61508により「（時間当たりの検出できる危険側故障割合）／（時間当たりの全危険側故障割合）」と定義されている。危険側故障とは故障により危険状態になる可能性のある故障と定義されている。あるいは自己診断率はλ_{DD}／λ_{D}と定義されている（ここで、λ_{DD}：自己診断で検出可能な危険側故障確率、λ_{D}：危険側故障確率）。

【００１５】

上記ダブルＲＡＭ方式では、上述した通り、データを書き込む際には、所定のデータ領域にデータを書き込むとともに反転データ領域に反転データを書き込む。例えばアドレス＝‘０１０１’にデータ‘１２３４’を書き込むとともに、反転アドレス＝‘ＦＥＦＥ’に反転データ‘ＥＤＣＢ’を書き込むことになる（尚、何れも１６進数表記）。そして、データを読む際には、データ領域のデータと反転データ領域のデータを読み込み、反転データ領域のデータを反転させたものがデータ領域のデータと一致するか確認する。一致しない場合はＲＡＭ故障と判断する。

【００１６】

ダブルＲＡＭ方式は、実現方法が容易であり、また例えばオンラインの処理タイミングで適用すると、その時点でＲＡＭに上述のソフトエラーが発生している場合、これを検出できる等（尚、ＲＡＭテストパターンによる診断では、処理タイミングが異なるので検出できない。よって、誤ったデータでオンライン処理が実行されることになる）、上述のソフトエラーを含め非特許文献１の第二分冊附属書Ａ表１の故障の殆どを検出できる。しかし、上記の通り、例えば‘０’に固定（固着）したセルに対して殆どの場合は‘０’のデータを書き込む場合等には、‘０’以外のデータを書くまでは異常検出できないなど、自己診断率を完全には担保できない。

【００１７】

また、安全機能はプロセスの安全を担保できる時間内に処理をする必要があるため、ＲＡＭテストパターンによる診断により安全機能の処理を妨げてはならない。尚、この点で、上記ダブルＲＡＭ方式は、安全機能の為の処理（安全関連アプリケーション処理というものとする）の実行の際に、そのデータ・リード／ライトに伴ってＲＡＭ診断を行うことができるので、安全機能の処理を妨げることはない。

【００１８】

また、ＲＡＭテストパターンによる診断は、テストパターンによりＲＡＭ格納内容を書き変える必要があり、ＲＡＭに格納されているデータを一時的に退避させたうえでＲＡＭにテストパターンをリード／ライトする必要がある。この為、ＲＡＭの診断、ＲＡＭデータの退避、復元の迅速かつ安価な実現が課題となる。

【００１９】

また、非特許文献２には、ソフトエラーの対策の１つとしてＥＣＣ（エラー検出訂正）機能の活用が記載されているが、ハードウェアで実装する場合は高価となり、ソフトウェアの実装では各アドレス単位にチェック情報を管理する必要があり、処理が複雑になり現実的ではない。

【００２０】

本発明の課題は、ＲＡＭ故障を確実にかつ早期に検出することができるＲＡＭ故障診断装置等を提供することである。

【課題を解決するための手段】

【００２１】

本発明のＲＡＭ故障診断装置は、基本的に、１以上の安全関連データを格納する安全関連データ格納領域を有するＲＡＭの故障診断を行うＲＡＭ診断装置であって、任意の前記安全関連データに関するアクセス処理の際に、ダブルＲＡＭ方式によるＲＡＭ診断処理を

行うダブルＲＡＭ方式診断手段と、定期的に、少なくとも前記各安全関連データの格納領域を含む各診断対象記憶領域毎に、その診断対象記憶領域内の各アドレスを順次診断対象アドレスとして、該各診断対象アドレス毎に、複数のテストデータの書き込みと、該診断対象アドレスからのデータ読み出し及びテストデータとの一致確認を行うことによって、診断対象アドレスの診断を行うテストパターン方式診断手段とを有する。

【００２２】

本発明のＲＡＭ故障診断装置では、ダブルＲＡＭ方式による診断すなわち任意の安全関連データへのアクセス時の該データ格納領域に対する診断と、テストパターン方式による診断すなわち所定のテストデータを用いた定期的な診断とを併用する。ダブルＲＡＭ方式による診断対象となる各安全関連データの格納領域は、必ず、テストパターン方式による診断対象領域とし、二重に診断を行うようにする。ダブルＲＡＭ方式、テストパターン方式それぞれの弱点を補い合うことにより、自己診断率を向上させることができる。

【００２３】

本発明のＲＡＭ故障診断装置は、１以上の安全関連データを格納する安全関連データ格納領域を有するＲＡＭの故障診断を行うＲＡＭ診断装置であって、任意の前記安全関連データに関するアクセス処理の際に、ダブルＲＡＭ方式によるＲＡＭ診断処理を行うダブルＲＡＭ方式診断手段と、定期的に、少なくとも前記各安全関連データの格納領域を含む各診断対象記憶領域毎に、その診断対象記憶領域内の各アドレスを順次診断対象アドレスとして、該各診断対象アドレスへの診断処理実行毎に、該診断対象アドレスを含む３以上の連続するアドレスに対して複数のテストデータを順次書き込むこと、診断対象アドレスからデータ読み出すことによって、診断対象アドレスの診断を行うテストパターン方式診断手段とを有する。

【００２４】

テストパターン方式のＲＡＭ診断において、例えば診断対象アドレスの信号線と他のアドレスの信号線との短絡等の異常を、検出し易くする。

また、上記ＲＡＭ故障診断装置において、例えば、前記複数のテストデータは、前記診断対象アドレスの全ビットのオン・オフ実施させる複数のテストデータ、または／且つ、ビットオン・オフパターンが相互に異なる複数のテストデータである。

【００２５】

この様な適切な複数のテストデータを適用することで、固着異常や信号線の短絡等の異常を、ほぼ確実に検出できるようにし、自己診断率を向上させることができる。

また、上記ＲＡＭ故障診断装置において、例えば、前記診断対象アドレスを含む３以上の連続するアドレスは、該診断対象アドレスとその直前のアドレスと直後のアドレスとの３つのアドレスであり、前記テストパターン方式診断手段は、前記診断対象アドレスに書き込むテストデータと、前記直前と直後のアドレスに書き込むテストデータとの組み合わせを順次変えながら、該診断対象アドレスからデータ読み出してデータ変化の有無を確認することで、診断対象アドレスの診断を行う。

【００２６】

上記「診断対象アドレスを含む３以上の連続するアドレス」を、診断対象アドレスとその直前、直後のアドレスとの３つのアドレスに限定する。これによって、処理対象を限定し処理の高速化を図ることができると共に、診断対象アドレスとその直前／直後のアドレスとの信号線の短絡等の異常を検出できることで自己診断率を向上させることができる。

【００２７】

また、上記ＲＡＭ故障診断装置において、例えば、前記テストパターン方式診断手段は、前記各診断対象アドレスへの診断処理実行毎に、前記診断対象アドレスを含む３以上の連続するアドレスの格納データをレジスタに退避させてから、前記複数のテストデータによる該診断対象アドレスの診断処理を行う。

【００２８】

勿論、診断処理終了したら、レジスタに退避させていたデータを元に戻す（復帰させる）。従来、データの退避先はＲＡＭ内等としていたが、これをレジスタとすることで、デ

ータの退避、復帰を高速に行えるようになり、以って全体としての処理の高速化も図れる。

【００２９】

また、上記ＲＡＭ故障診断装置において、例えば、前記テストパターン方式診断手段による診断処理は、前記任意の前記安全関連データに対するアクセス処理を含む安全関連アプリケーション処理よりもタスクレベルが低く設定されており、前記安全関連アプリケーション処理を実行する場合には、前記テストパターン方式診断手段による診断処理は一時中断される。

【００３０】

また、上記ＲＡＭ故障診断装置において、例えば、前記各安全関連データの格納領域を示すアドレス情報が予め記憶されたアドレス情報記憶手段を更に有し、前記ダブルＲＡＭ方式診断手段は、前記任意の前記安全関連データに関するアクセス処理に係るパラメータを受け取ると、該パラメータのアドレス情報が上記アドレス情報記憶手段に記憶されているアドレス情報と一致するか否かを確認し、一致する場合に前記ダブルＲＡＭアルゴリズムによるＲＡＭ診断処理を行う。

【発明の効果】

【００３１】

本発明のＲＡＭ故障診断装置等によれば、ＲＡＭ故障を確実にかつ早期に検出することができる。

【図面の簡単な説明】

【００３２】

【図１】本例のＲＡＭ診断装置の構成ブロック図である。

【図２】本例のＲＡＭ診断処理のタスクスケジュールを示す図である。

【図３】（ａ）、（ｂ）は、ＲＡＭ管理テーブルの具体例を示す図である。

【図４】安全関連データの書込処理のフローチャート図である。

【図５】安全関連データの読込処理のフローチャート図である。

【図６】ＲＡＭテストパターン診断処理のフローチャート図である。

【図７】（ａ）～（ｆ）は、図６の処理中に書き込まれるテストパターンの推移を示す図である。

【図８】安全関連アプリケーション処理部の処理フローチャート図である。

【発明を実施するための形態】

【００３３】

以下、図面を参照して本発明の実施の形態について説明する。

図１は、本例のＲＡＭ診断装置の構成ブロック図である。

尚、このＲＡＭ診断装置は、例えば上述したプログラマブルコントローラシステム（制御システム）におけるコントローラ本体に相当する。上記の通り、コントローラ本体（ＣＰＵ、ＲＡＭ等を有する）は、自己診断処理として自己のＲＡＭの診断処理を行うものであり、特にこの様なＲＡＭ診断処理に係る構成を図１に示している。勿論、コントローラ本体は診断処理以外の処理（制御処理等）も行っているが、これについては特に図示／説明しない。

【００３４】

図１に示すＲＡＭ診断装置は、ハードウェア的にはＣＰＵ１、ＲＯＭ１０、ＲＡＭ２０から成る。ＣＰＵ１は、特に図示しないアドレス線、データ線等を介して、ＲＯＭ１０、ＲＡＭ２０にアクセスして、プログラムの読出し・実行や、データ・リード／ライト処理を行う。

【００３５】

ＲＡＭ２０は、本診断装置による診断対象のメモリである。ＲＡＭ２０に対する診断処理は、ＣＰＵ１が、ＲＯＭ１０に予め記憶されている各種アプリケーションプログラム（後述する）を読出し・実行することにより実現される。ＲＯＭ１０は、実行形式のプログラムや固定データを格納するメモリであり、ＲＯＭに限らずフラッシュメモリやＲＡＭ等

であってよい。

【００３６】

ＲＯＭ１０には、上記実行形式のプログラムとして例えば安全関連アプリケーション処理１１，ＲＡＭテストパターン診断処理１２、安全関連データアクセス関数１３等が記憶されている。また、ＲＯＭ１０には、ＲＡＭ管理テーブル１４等が格納されている。

【００３７】

上記安全関連データアクセス関数１３は、例えば上記安全関連アプリケーション処理１１のプログラム内にインライン展開されており、安全関連アプリケーション処理１１の実行中に呼び出されて所定の処理を実行する。安全関連データアクセス関数１３は、ＲＡＭ２０の安全関連データ領域２１内の任意の記憶領域に対して任意のデータをリード／ライトするための関数である。但し、データ・リード／ライトに伴って、上記ダブルＲＡＭ方式のＲＡＭ診断に係る処理も実行するものであり、反転安全関連データ領域２２に対する反転データのリード／ライトも一緒に行うことになる。ライト（書込）時の処理を図４に示し、リード（読込）時の処理を図５に示し、後に説明するものとする。

【００３８】

ＲＡＭ管理テーブル１４は、本方式のＲＡＭ診断の管理をするためのテーブルである。

図３（ａ）、（ｂ）にＲＡＭ管理テーブル１４のデータ構成例を示す。

図３（ａ）、（ｂ）に示すように、ＲＡＭ管理テーブル１４は、例えばＲＡＭテストパターン方式診断管理テーブル３０、ダブルＲＡＭ方式診断管理テーブル４０等の複数のテーブルより成るものである。ＲＡＭテストパターン方式診断管理テーブル３０はＲＡＭテストパターン診断処理１２の処理で参照されるテーブルであり、ダブルＲＡＭ方式診断管理テーブル４０は安全関連データアクセス関数１３の処理で参照されるテーブルである。これらのテーブル３０，４０の詳細については後に説明する。

【００３９】

ＲＡＭテストパターン診断処理１２は、所定のＲＡＭテストパターンによる診断テストを定周期で実施する為のアプリケーションプログラムである。図６に、ＲＡＭテストパターン診断処理１２の処理フローチャート図を示し、後に説明する。

【００４０】

安全関連アプリケーション処理１１は、安全関連データに係わる処理（各種制御処理や上述した安全機能の処理（例えば非常停止ボタン押下に応じた処理等）等）を実行するアプリケーションプログラムである。図８に安全関連アプリケーション処理１１による処理例のフローチャート図を示し、後に説明する。

【００４１】

ここで、ＣＰＵ１は、上記各アプリケーション／関数を実行することで所定の処理機能を実現することになり、これよりＣＰＵ１は図示しないが各種処理機能部を有するものと見做してよく、本説明では以下の通り定義する。

【００４２】

すなわち、図１には示していないが、ＣＰＵ１は、安全関連アプリケーション処理部１１’、ＲＡＭテストパターン診断処理部１２’、安全関連データアクセス処理部１３’の各種処理機能部を有するものである。

【００４３】

ＣＰＵ１は、安全関連アプリケーション処理１１のプログラムを実行することにより上記安全関連アプリケーション処理部１１’の処理機能を実現する。同様に、ＣＰＵ１は、ＲＡＭテストパターン診断処理１２のプログラムを実行することにより上記ＲＡＭテストパターン診断処理部１２’の処理機能と実現する。ＣＰＵ１は、安全関連データアクセス関数１３のプログラムを実行することにより上記安全関連データアクセス処理部１３’の処理機能を実現する。

【００４４】

尚、上述したように安全関連データアクセス関数１３は安全関連アプリケーション１１から呼び出されることから、安全関連データアクセス処理部１３’は、安全関連アプリケ

ーション処理部１１’の処理機能の一部と見做してもよい。

【００４５】

そして、後述する図４、図５は安全関連データアクセス処理部１３’の処理フローチャート図、図６はＲＡＭテストパターン診断処理部１２’の処理フローチャート図、図８は安全関連アプリケーション処理部１１’の処理フローチャート図であると言うこともできる。

【００４６】

次に、図１に示すＲＡＭ２０について説明する。

ＲＡＭ２０には、安全関連データ領域２１、反転安全関連データ領域２２等の記憶領域が存在する。

【００４７】

安全関連データ領域２１は、各種安全関連データを格納する記憶領域であり、ＲＡＭ２０において特に故障診断を必要とするＲＡＭデータ領域を示す。

ここで、安全関連データとは、例えば制御システムにおいてプログラマブルコントローラ本体が制御対象機器の動作を制御することに関するデータ等であり、特に作業員等の安全に係わるデータである。すなわち、例えば、モータやプレス機の動作の開始／停止を指示することに係るデータである。よって、安全関連データに異常が生じた場合、それによって誤った指示が制御対象機器に送られると、突然、モータやプレス機が動作開始する等して近くにいる作業員に危険が生じる可能性がある。このように、（作業員等の）安全に関連するデータであるから、安全関連データと呼ぶ。

【００４８】

よって、ＲＡＭ２０の記憶領域のなかで特に安全関連データに関する記憶領域（上記安全関連データ領域２１）に関しては、異常が発生した場合には確実にかつ早期に検出することが重要である。

【００４９】

尚、特に図示／説明等しないが、ＲＡＭ２０には制御に係るデータであるが安全関連データではないデータ（非安全関連データというものとする）も記憶されていてもよい。同様に、ＲＯＭ１０にはこの様な非安全関連データを用いた制御処理を実行させるプログラムも格納されていてもよい。

【００５０】

また、反転安全関連データ領域２２は、上記安全関連データ領域２１に格納される安全関連データの反転データが格納される記憶領域である。換言すれば、ダブルＲＡＭ方式によるＲＡＭ診断処理で必要とする反転データを格納するＲＡＭデータ領域である。尚、反転データに関しては、既に従来技術の課題において説明している。

【００５１】

上述した構成により、本例のＲＡＭ診断装置では、ＲＡＭテストパターン方式とダブルＲＡＭ方式の２種類の診断方式によるＲＡＭ診断を実行する。この様な２種類のＲＡＭ診断処理を、例えば、図２に示すようにして実行させる。

【００５２】

図２は、本例のＲＡＭ診断処理のタスクスケジュールを示す図である。

図２に示す「安全関連アプリケーション処理」は、上記安全関連アプリケーション処理部１１’の処理であり、その処理中に上記安全関連データアクセス処理部１３’の処理も実行される。つまり、図示の「安全関連アプリケーション処理」では、任意の安全関連データのリード／ライト処理に伴って、この安全関連データの格納領域に対するＲＡＭテスト（ダブルＲＡＭ方式によるＲＡＭテスト）が実行される。また、図２に示す「ＲＡＭテストパターン診断処理」は、上記ＲＡＭテストパターン診断処理部１２’によるＲＡＭ診断処理である。

【００５３】

図２は、イベント駆動の場合の例であり、「安全関連アプリケーション処理」ではダブルＲＡＭ方式によるＲＡＭ診断を実施し、「ＲＡＭテストパターン診断処理」ではＲＡＭ

テストパターン方式によるＲＡＭ診断を実施する。そして、「ＲＡＭテストパターン診断処理」は、「安全関連アプリケーション処理」より低いタスクレベルとしている。従って、「ＲＡＭテストパターン診断処理」を実行中であっても「安全関連アプリケーション処理」の処理実行が優先され、「ＲＡＭテストパターン診断処理」は処理を一時中断しなければならないことになる。

【００５４】

テストのためにＲＡＭデータを書き変えている間は安全機能を実行できないため、例えばプレス機の緊急停止のイベントが発生し、これに応じた「安全関連アプリケーション処理」を実行する場合に「ＲＡＭテストパターン診断処理」が実行中であると問題となる。しかし、本例では上記の通り「ＲＡＭテストパターン診断処理」のタスクレベルが低い為、「ＲＡＭテストパターン診断処理」を中断して「安全関連アプリケーション処理」が実行されるので、問題を解消できる。

【００５５】

尚、時間駆動の場合、「ＲＡＭテストパターン診断処理」は、「安全関連アプリケーション処理」の空き時間に分割して処理する形になる。

図３（ａ）、（ｂ）に、ＲＡＭ管理テーブル１４の具体例を示す。既に述べたように、ＲＡＭ管理テーブル１４は、例えば図３（ａ）に示すＲＡＭテストパターン方式診断管理テーブル３０、図３（ｂ）に示すダブルＲＡＭ方式診断管理テーブル４０等の複数のテーブルより成るものである。

【００５６】

図３（ａ）に示すＲＡＭテストパターン方式診断管理テーブル３０には、ＲＡＭテストパターン方式による診断処理で必要とする管理データが格納される。当該テーブル３０は、図示の“ＲＡＭテストパターン方式診断登録領域番号”３１、管理データ３２から成る。

【００５７】

管理データ３２には、ＲＡＭテストパターン方式によるＲＡＭ診断を行う記憶領域のアドレス範囲（登録領域先頭アドレスと登録領域最終アドレス）が格納される。

“ＲＡＭテストパターン方式診断登録領域番号”３１は、上記各管理データ３２（各診断対象領域のアドレス範囲）に対して任意の管理番号を付与したものであり、本例では図示の通り、番号３１は‘１’から連番で付与する。

【００５８】

この様に、ＲＡＭテストパターン方式診断管理テーブル３０には、ＲＡＭテストパターン方式の診断対象となる各記憶領域（アドレス範囲）が、管理番号（登録領域番号３１）付きで予め登録されている。尚、以下、上記管理データ３２によって示されるアドレス範囲（登録領域先頭アドレスと登録領域最終アドレス）を、登録領域という場合もあるものとする。

【００５９】

ここで、管理データ３２で示されているアドレス範囲（登録領域先頭アドレスと登録領域最終アドレス）は、後述するダブルＲＡＭ方式診断管理テーブル４０で管理されるアドレス範囲を包含しなければならない。すなわち、ダブルＲＡＭ方式による診断対象領域は、ＲＡＭテストパターン方式によっても診断することで二重に診断する必要がある。当該条件を満たせば上記アドレス範囲（登録領域先頭アドレスと登録領域最終アドレス）は任意に設定してよい。よって例えば、ダブルＲＡＭ方式による診断対象領域を含む広範囲の領域を、管理データ３２のアドレス範囲としてもよい。

【００６０】

また、図３（ｂ）に示すダブルＲＡＭ方式診断管理テーブル４０は、“ダブルＲＡＭ方式診断ブロック番号”４１、管理データ４２から成る。

本例のダブルＲＡＭ方式診断管理テーブル４０には、ダブルＲＡＭ方式でのアクセス単位に対応したアドレス範囲を予め登録する。ダブルＲＡＭ方式でのアクセス単位は、任意のデータとその反転データのリード・ライトであり、本例の場合は上記安全関連アプリケ

ーション処理部１１’の処理に係る各安全関連データとその反転データのリード・ライトである。よって、本例における上記ダブルＲＡＭ方式でのアクセス単位に対応したアドレス範囲は、各安全関連データの記憶領域及びその反転データの記憶領域を示すアドレス範囲となる。

【００６１】

これより、図３（ｂ）に示すダブルＲＡＭ方式診断管理テーブル４０は、各々に上記ダブルＲＡＭ方式でのアクセス単位に対応したアドレス範囲が登録された複数のブロックより成り、各ブロックには識別番号（ダブルＲＡＭ方式診断ブロック番号４１）が割り当てられている。尚、本例では“ダブルＲＡＭ方式診断ブロック番号”４１は‘１’から連番で設定する（１、２、３、４・・・）。

【００６２】

そして、各ブロック毎に、管理データ４２として、安全関連データの記憶領域のアドレス範囲（安全関連データ領域先頭アドレスと安全関連データ領域最終アドレス）、当該安全関連データの反転データの記憶領域のアドレス範囲（反転安全関連データ領域先頭アドレスと反転安全関連データ領域最終アドレス）とが登録されている。

【００６３】

尚、管理データ４２が示すアドレス範囲は、必ず、ダブルＲＡＭ方式でのアクセス単位と一致させる。すなわち、上記安全関連アプリケーション処理部１１’は、各安全関連データへのアクセス時（リード／ライト時）に、上記の通り安全関連データアクセス処理部１３’を呼び出すと共にパラメータを渡す。このパラメータには（図４、図５に示しており詳しくは後述するが）アクセス対象の安全関連データの格納領域を示す情報（先頭アドレス、バイト数等）、この安全関連データの反転データの格納領域を示す情報（先頭アドレス、バイト数等）、ブロック番号等が含まれている。

【００６４】

このパラメータが示す上記「格納領域を示す情報」と、上記管理データ４２が示すアドレス範囲とが一致するように、安全関連データアクセス処理部１３’のパラメータとダブルＲＡＭ方式診断管理テーブル４０とを作成しておく。但し、メモリ異常等が原因で一致しない状況となる場合があり、後述するチェック処理によって異常検出することになる。以下、更に詳しく説明する。

【００６５】

上記アクセス対象の安全関連データの格納領域は、当然、図１に示す安全関連データ領域２１内の一部の記憶領域であり、その反転データの格納領域も当然、図１に示す反転安全関連データ領域２２内の一部の記憶領域である。

【００６６】

これら各安全関連データ等の格納領域アドレスは、例えば上記安全関連アプリケーション処理１１のプログラム作成者が、任意に決めて（あるいは予め決まっているアドレスを用いて）、当該プログラム内に記述するものであり、更にこの格納領域アドレスをダブルＲＡＭ方式診断管理テーブル４０に登録しておくものである。すなわち、上記アクセス対象の安全関連データとその反転データの格納領域を示すアドレス範囲を、上記管理データ４２の各アドレス範囲として登録しておく。

【００６７】

ここで、この様にして事前に作成されたダブルＲＡＭ方式診断管理テーブル４０があるならば、上記パラメータとしてはブロック番号等があればよく、ブロック番号を用いて管理テーブル４０を検索すれば、アクセス対象の安全関連データ（及びその反転データ）の格納領域が分かることになる。あるいは、逆に、パラメータに格納領域情報が含まれているならばテーブル４０を参照しなくてもアクセス対象領域が分かることになる。

【００６８】

これに対して、本手法では、上記の通り、テーブル４０を設けると共に、パラメータにはアクセス対象の安全関連データ（及びその反転データ）の格納領域を示す情報も含まれている。これは、後述するように図４、図５の処理においてアドレス範囲チェックを行う

ことで、安全関連アプリケーション処理１１のプログラムやテーブル４０を記憶する記憶領域（例えばＲＯＭ１０内の任意の記憶領域）に異常が生じてデータが変化した場合には、これを検出できるようにする為である。

【００６９】

図４や図５の処理におけるパラメータが示す格納領域と、ダブルＲＡＭ方式診断管理テーブル４０に登録されている格納領域（そのブロック番号４１がパラメータのブロック番号と同一である管理データ４２が示す各領域）とは、本来は一致していなければならない。不一致の場合には、ＲＯＭ１０等に異常が生じた可能性がある。本手法では、ＲＡＭの異常に限らず、この様なＲＯＭ等の異常も検出できる。

【００７０】

以下、図４、図５を参照して、安全関連データアクセス処理部１３’の処理について説明する。図４は安全関連データのライト（書込）処理、図５は安全関連データのリード（読込）処理の処理フローチャート図である。

【００７１】

まず、図４を参照して、安全関連データアクセス処理部１３’による安全関連データのライト（書込）処理について説明する。

上述したように、安全関連データアクセス処理部１３’には、安全関連アプリケーション処理部１１’から呼び出される際にパラメータが与えられる。ライト（書込）処理の際には、図４に示すように、パラメータとして、書込先頭アドレス、反転書込先頭アドレス、書込データ、書込データバイト数、管理テーブルブロック番号が与えられる。尚、パラメータには、更に、ライト（書込）処理であるのかリード（読込）処理であるのかを示す情報等も含まれていてもよい。また、書込データは、書込先頭アドレスと書込データバイト数とによって示される記憶領域に書き込むべき、任意の安全関連データである。

【００７２】

管理テーブルブロック番号は、図３（ｂ）のダブルＲＡＭ方式診断管理テーブル４０のダブルＲＡＭ方式診断ブロック番号４１に相当する番号である。また、図４の処理の処理結果としてのレジスタ渡し関数値は、“正常終了”または“異常終了”となる。

【００７３】

本処理は、基本的に、上記パラメータの書込データを上記書込先頭アドレス及び書込データバイト数によって示される領域に書き込むと共に、上記書込データの反転データを、上記反転書込先頭アドレス及び書込データバイト数によって示される領域に書き込む。そして、書き込んだ書込データ及びその反転データに基づくＲＡＭ診断を行う。更に、上記パラメータの管理テーブルブロック番号等を用いたＲＯＭ１０等のチェックも行う。

【００７４】

図４の処理を実現する上記安全関連データアクセス関数１３は、安全関連アプリケーション処理中において任意の安全関連データを安全関連データ領域２１へダブルＲＡＭ方式にて書き込む場合に使用する関数である。

【００７５】

図４の処理では、まず、上記パラメータの管理テーブルブロック番号を用いて上記管理テーブル４０を検索して、該当するブロック（そのブロック番号４１が上記パラメータのブロック番号と同じであるレコード）の管理データ４２の一部を取得する。すなわち、上記パラメータで指定されたブロックに対応した安全関連データ領域先頭アドレス、安全関連データ領域最終アドレスを読み込む。換言すれば、上記パラメータの書込データの書込先の記憶領域のアドレス範囲（先頭アドレスと最終アドレス）を読み込む。更に、上記パラメータの書込先頭アドレスと書込データバイト数から、上記安全関連データの書込先の記憶領域の最終アドレスを計算する（ステップＳ１１）。

【００７６】

そして、ステップＳ１１において管理テーブル４０から取得した上記安全関連データ領域先頭アドレス及び安全関連データ領域最終アドレスと、上記パラメータの書込先頭アドレス及び計算した最終アドレスとを比較して、両者が一致するか否かを判定する。換言す

れば、書込データの書込先のアドレス範囲が、パラメータと管理テーブル４０とで一致するか否かを判定する（ステップＳ１２）。

【００７７】

書込データの書込先のアドレス範囲が不一致の場合にはアドレス範囲チェックＮＧとなり（ステップＳ１２，ＮＯ）、異常終了となる（ステップＳ２１）。

書込データの書込先のアドレス範囲が一致した場合にはアドレス範囲チェックＯＫとなり（ステップＳ１２，ＹＥＳ）、ステップＳ１３以降の処理へと進む。

【００７８】

尚、各書込データ毎に対応するブロック番号が予め決められている（例えばプログラマが決める）。また、上記ステップＳ１１，Ｓ１２の処理は、特に、ＲＡＭ管理テーブル１４（そのテーブル４０）のアドレスデータまたはパラメータのデータビット化け等を検出するため（上記ＲＯＭ等の異常を検出するため）に行うものである。

【００７９】

ステップＳ１３では、上記パラメータの書込データを上記書込先の記憶領域に書き込む。すなわち、上記パラメータにより指定される記憶領域（書込先頭アドレスから書込データバイト数分の領域）に、書込データを書き込む（ステップＳ１３）。勿論、テーブル４０の安全関連データ領域先頭アドレスから安全関連データ領域最終アドレスまでの領域に、書込データを書き込むものであってもよい（一致検証済みなので、同じことである）。

【００８０】

続いて、上記パラメータの書込データの反転データを書き込む記憶領域に関しても、上記データビット化け等を検出する為の処理を上記書込データの場合と略同様にして行う。すなわち、まず、管理テーブル４０の上記“該当するレコード”から、その管理データ４２の一部として今度は反転安全関連データ領域先頭アドレス、反転安全関連データ領域最終アドレスを読み込む。更に、上記パラメータの反転書込先頭アドレスと書込データバイト数から、書込データの反転データの書込先の記憶領域の最終アドレスを計算する（ステップＳ１４）。

【００８１】

そして、ステップＳ１４において管理テーブル４０から取得した上記反転安全関連データ領域先頭アドレス及び反転安全関連データ領域最終アドレスと、上記パラメータの反転書込先頭アドレス及び計算した最終アドレスとを比較して、両者が一致するか否かを判定する。換言すれば、書込データの反転データの書込先のアドレス範囲が、パラメータと管理テーブル４０とで一致するか否かを判定する（ステップＳ１５）。

【００８２】

書込データの反転データの書込先のアドレス範囲が、パラメータと管理テーブル４０とで一致しない場合には、アドレス範囲チェックＮＧとなり（ステップＳ１５，ＮＯ）、異常終了となる（ステップＳ２１）。一方、書込データの反転データの書込先のアドレス範囲が、パラメータと管理テーブル４０とで一致した場合には、アドレス範囲チェックＯＫとなり（ステップＳ１５，ＹＥＳ）、ステップＳ１６以降の処理へと進む。

【００８３】

ステップＳ１６以降の処理では、まず、上記パラメータの書込データを反転させて“書込データの反転データ”を生成する（ステップＳ１６）。続いて、ステップＳ１６で生成した“書込データの反転データ”を、上記反転データの書込先のアドレス範囲の記憶領域に書き込む。すなわち、例えば上記パラメータの反転書込先頭アドレスから書込データバイト数分の記憶領域に、“書込データの反転データ”を書き込む（ステップＳ１７）。

【００８４】

そして、ステップＳ１３で書込データを書き込んだ記憶領域からデータ（第１データというものとする；正常であれば書込データであるはず）を読出すと共に、ステップＳ１７で“書込データの反転データ”を書き込んだ記憶領域からデータ（第２データというものとする；正常であれば“書込データの反転データ”であるはず）を読み出す。そして、ステップＳ１７で読み出した第２データを反転させて、当該“第２データの反転データ”が

上記第１データと一致するか否かを確認する（ステップＳ１８）。

【００８５】

データ一致の場合には（ステップＳ１９，ＹＥＳ）、正常終了となり、例えば「正常終了」を関数値として呼び出し元に復帰する（ステップＳ２０）。一方、データ不一致の場合には（ステップＳ１９，ＮＯ）、上記異常終了となる（ステップＳ２１）。この場合には、例えば「異常終了」を関数値として呼び出し元に復帰する（ステップＳ２１）。

【００８６】

次に、図５を参照して、安全関連データアクセス処理部１３’による安全関連データのリード（読込）処理について説明する。上記安全関連データアクセス関数１３は、安全関連アプリケーション処理１１が安全関連データ領域２１のデータをダブルＲＡＭ方式にて読み込む場合に使用する関数である。

【００８７】

上述したように、安全関連データアクセス処理部１３’は、安全関連アプリケーション処理部１１’から読み出される際にパラメータが与えられる。リード（読込）処理の際には、図５に示すように、パラメータとして、読込先頭アドレス、反転読込先頭アドレス、読込データバイト数、管理テーブルブロック番号が与えられる。また、図４の処理の処理結果としてのレジスタ渡し復帰データは“読込データ”となる。レジスタ渡し復帰データとして、ＲＡＭ２０の指定領域（読込先頭アドレスから読込データバイト数分の記憶領域）から読み込んだデータ（読込データ）が、予め定められたレジスタ等に格納される。また、レジスタ渡し関数値は“正常終了”または“異常終了”となる。

【００８８】

本処理は、基本的に、上記指定領域（読込先頭アドレスから読込データバイト数分の記憶領域）からデータ(第３データというものとする)を読み込むと共に、“指定領域の反転領域”（上記反転読込先頭アドレスから読込データバイト数分の記憶領域）からデータ(第４データというものとする)を読み込む。換言すれば、所定の記憶領域に記憶されているデータ（上記第３データ）を読み出すと共に、この第３データの反転データが記憶されているはずの記憶領域からデータ（上記第４データ）を読み出す。

【００８９】

そして、上記第４データを反転させて、この“第４データの反転データ”が上記第３データと一致するか否かを確認する。この様にして、ダブルＲＡＭアルゴリズムによるデータリード時のＲＡＭ診断を行う。更に、上記パラメータの管理テーブルブロック番号等を用いたＲＯＭ等の異常チェック（データビット化けの検出等）も、図４の場合と略同様に行う。

【００９０】

図５の処理を実現する上記安全関連データアクセス関数１３は、安全関連アプリケーション処理中において、上記書き込みの場合だけでなく、任意の安全関連データを安全関連データ領域２１から読み込む場合にも使用する関数である。

【００９１】

図５の処理では、まず、上記パラメータの管理テーブルブロック番号を用いて上記管理テーブル４０を検索して、該当するブロック（そのブロック番号４１が上記パラメータのブロック番号と同じであるレコード）の管理データ４２の一部を取得する。すなわち、上記パラメータで指定されたブロックに対応した安全関連データ領域先頭アドレス、安全関連データ領域最終アドレスを読み込む。換言すれば、読込データの格納領域のアドレス範囲（先頭アドレスと最終アドレス）を読み込む。更に、上記パラメータの読込先頭アドレスと読込データバイト数から、上記読込データの記憶領域の最終アドレスを計算する（ステップＳ３１）。

【００９２】

そして、ステップＳ３１において管理テーブル４０から取得した上記安全関連データ領域先頭アドレス及び安全関連データ領域最終アドレスと、上記パラメータの読込先頭アドレス及び計算した最終アドレスとを比較して、両者が一致するか否かを判定する。換言す

れば、読込データの格納領域のアドレス範囲が、パラメータと管理テーブル４０とで一致するか否かを判定する（ステップＳ３２）。

【００９３】

読込データの格納領域のアドレス範囲が不一致の場合にはアドレス範囲チェックＮＧとなり（ステップＳ３２，ＮＯ）、異常終了となる（ステップＳ４０）。

読込データの格納領域のアドレス範囲が一致した場合にはアドレス範囲チェックＯＫとなり（ステップＳ３２，ＹＥＳ）、ステップＳ３３以降の処理へと進む。

【００９４】

上記ステップＳ３１，Ｓ３２の処理は、例えばＲＡＭ管理テーブル１４（そのテーブル４０）のアドレスデータまたはパラメータのデータビット化け等を検出するために行うものである。

【００９５】

ステップＳ３３では、上記パラメータで指定される記憶領域に格納されているデータ（第３データ）を読み込む。すなわち、上記読込先頭アドレスから読込データバイト数分の記憶領域から上記第３データ（読込データ）を読み込む（ステップＳ３３）。勿論、テーブル４０の安全関連データ領域先頭アドレスから安全関連データ領域最終アドレスまでの領域から、上記第３データを読み込むものであってもよい（一致検証済みなので、同じことである）。

【００９６】

ここで、図４で説明したように、データ書き込みの際に、その反転データを反転安全関連データ領域２２に書き込んでいる。リード処理では、この反転データの管理データに関しても、上記データビット化け等（ＲＯＭ異常等）を検出する為の処理を上記読込データの場合と略同様にして行う。

【００９７】

すなわち、まず、管理テーブル４０の上記“該当するレコード”から、その管理データ４２の一部として今度は反転安全関連データ領域先頭アドレス、反転安全関連データ領域最終アドレスを読み込む。更に、上記パラメータの反転読込先頭アドレスと読込データバイト数から、“読込データの反転データ”の格納領域の最終アドレスを計算する（ステップＳ３４）。

【００９８】

そして、ステップＳ３４において管理テーブル４０から取得した上記反転安全関連データ領域先頭アドレス及び反転安全関連データ領域最終アドレスと、上記パラメータの反転読込先頭アドレス及び計算した最終アドレスとを比較して、両者が一致するか否かを判定する。換言すれば、“読込データの反転データ”の格納領域のアドレス範囲が、パラメータと管理テーブル４０とで一致するか否かを判定する（ステップＳ３５）。

【００９９】

“読込データの反転データ”の格納領域のアドレス範囲が不一致の場合にはアドレス範囲チェックＮＧとなり（ステップＳ３５，ＮＯ）、異常終了となる（ステップＳ４０）。一方、“読込データの反転データ”の格納領域のアドレス範囲が一致した場合にはアドレス範囲チェックＯＫとなり（ステップＳ３５，ＹＥＳ）、ステップＳ３６以降の処理へと進む。

【０１００】

ステップＳ３６以降の処理では、まず、上記“読込データの反転データ”の格納領域からデータ（上記第４データ）を読み出す。これは、例えば、パラメータの反転読込先頭アドレスから読込データバイト数分の記憶領域から上記第４データを読み込む（ステップＳ３６）。この第４データは、異常がなければ、上記“読込データ（第３データ）の反転データ”であるはずである。これより、ステップＳ３６で読み込んだ第４データを反転させて（ステップＳ３７）、当該“第４データの反転データ”が上記ステップＳ３で読み込んだ第３データ（読込データ）と一致するか否かを確認する（ステップＳ３８）。

【０１０１】

データ一致の場合には（ステップＳ３８，ＹＥＳ）、正常終了となり、例えば「正常終了」を関数値として読込データと共に呼び出し元に復帰する（ステップＳ３９）。一方、データ不一致の場合には（ステップＳ３８，ＮＯ）、上記異常終了となる（ステップＳ４０）。この場合には、例えば「異常終了」を関数値として呼び出し元に復帰する。

【０１０２】

次に、ＲＡＭテストパターン診断処理１２について説明する。

図６は、上記ＲＡＭテストパターン診断処理部１２’によるＲＡＭテストパターン診断処理のフローチャート図である。本処理は、ＲＡＭの診断間隔として定義された所定の時間周期で起動する。この時間周期は、たとえばプログラマ等が任意に決めてよい。本処理では、ＲＡＭテストパターン方式による診断を実施する。但し、図２の例の場合には、安全関連アプリケーション処理１１実行時には本処理は中断することになる。

【０１０３】

本処理では、上記ＲＡＭ管理テーブル１４のＲＡＭテストパターン方式診断管理テーブル３０に登録されている各記憶領域（上記登録領域）について順次、ＲＡＭテストパターン診断を実行することで、全ての登録領域についてＲＡＭテストパターン診断を行う。また、各登録領域毎に、その登録領域内の各アドレスについて順次、ＲＡＭテストパターン診断を実行することで、当該登録領域内の全てのアドレスについてＲＡＭテストパターン診断を行う。尚、以下の説明では図３（ａ）に示す例を用いるものとし、上記各登録領域は、図３（ａ）に示す登録領域番号３１に従って区別して呼ぶものとする。すなわち、ＲＡＭテストパターン方式診断登録領域番号３１が‘１’である登録領域を登録領域１、登録領域番号３１が‘２’である登録領域を登録領域２等と呼ぶものとする。

【０１０４】

図６の処理では、まず、上記ＲＡＭテストパターン方式診断管理テーブル３０から登録領域１の管理データ３２（登録領域１のアドレス範囲；登録領域先頭アドレスと登録領域最終アドレス）を読み込む（ステップＳ５１）。

【０１０５】

そして、上記登録領域１のアドレス範囲内の各アドレスを、順次、診断チェック対象アドレス（以下、対象アドレスというものとする）として、各対象アドレスに対してステップＳ５３～Ｓ６５のテストパターン方式のＲＡＭ診断を実行する（但し、途中で一度でもステップＳ５７の判定がＹＥＳになったら、ステップＳ６８～Ｓ７０の処理を実行し、本処理は終了する）。

【０１０６】

概略的には、最初は上記登録領域先頭アドレスを対象アドレスとしてテストパターン方式のＲＡＭ診断を実行し、ステップＳ６６の処理によって対象アドレスを順次変えながらテストパターン方式のＲＡＭ診断を実行していき、最後は上記登録領域最終アドレスを対象アドレスとしてテストパターン方式のＲＡＭ診断を行い、異常が無ければ登録領域１に関するＲＡＭ診断は完了したことになる。そして、ステップＳ６７の処理により、今度は登録領域２が診断対象となり、登録領域１の場合と略同様にしてＲＡＭ診断を実行し、異常が無ければ（図３（ａ）の例によれば）本処理は終了となる。

【０１０７】

以下、詳細に説明する。尚、本方式では基本的に複数のテストデータパターンを用いるものとし、本例ではテストデータパターン１、テストデータパターン２の２つのテストデータパターンを用いるものとする。更に、図６に示す例では、テストデータパターン１として１６進数‘５５５５’を用い、テストデータパターン２として１６進数‘ＡＡＡＡ’を用いるものとするが、この例に限らない。

【０１０８】

例えば更に複雑な故障を検出するため、テストデータパターンをＮ個設定することもできる。例えば、Ｎ＝１０のテストデータパターンの例として下記が挙げられる。

16進数‘5555’,16進数‘AAAA’,16進数‘3333’,16進数‘9999’,16進数‘CCCC’,16進数‘6666’,16進数‘0000’,16進数‘FFFF’,16進数‘F0F0’,16進数‘0F0F’。

【０１０９】
本手法において複数のテストデータパターンとして例えば16進数‘5555’と16進数‘AAAA’とを用いるのは、例えば対象アドレスの全ビット列のオンオフを試行することを１つの目的としている。尚、言うまでもないが、16進数‘５’は４ビット‘０１０１’であり、16進数‘Ａ’は４ビット‘１０１０’である。よって、16進数‘５’と‘Ａ’とを順次書き込むことで４ビットの全てのビットについてオンオフを試行することができる。
【０１１０】
これより、例えば、上述した‘０’固着状態のビットに対して‘０’を書き込んでも異常検出できないが‘１’を書き込むことで異常を検出することができる。その逆に、上述した‘１’固着状態のビットに対して‘１’を書き込んでも異常検出できないが‘０’を書き込むことで異常を検出することができる。対象アドレスの全ビットのオンオフを試行することで、対象アドレスの全ビットの固着異常を確実に検出できる。
【０１１１】
あるいは、信号線の短絡を検出できる。例えば、対象アドレスの１ビット目と２ビット目とが短絡（ショート）していた場合、‘１０１０’を書き込むと‘１１１０’となるので（１ビット目の‘１’につられて２ビット目も‘１’になるので）、異常を検出できる。尚、１ビット目とは最上位ビット（最も左側のビット）を意味する。
【０１１２】
但し、１ビット目と３ビット目とが短絡（ショート）していた場合には、‘１０１０’を書き込むと‘１０１０’となるので（１ビット目と３ビット目の両方が‘１’であるので）、異常を検出できない。これは、‘０１０１’の場合でも同様である。従って、この様な異常は、上述した複数のテストデータパターンとして16進数‘5555’と16進数‘AAAA’とを用いる場合には、検出できないことになる。
【０１１３】
更に、後述するように対象アドレスだけでなくその前後のアドレスにもテストデータパターンを格納するのは、対象アドレスのＲＡＭメモリセルのチェック方法の１つとしてその前後のセルによる対象アドレスのデータの変化をチェックすることを目的としている。
【０１１４】
すなわち、上記信号線の短絡（ショート）が生じるのは、対象アドレス内のセル同士に限らず、対象アドレスのセルと他のアドレスのセルとの間で生じる場合もある。これより、例えば、対象アドレスの１ビット目とその直前（または直後）のアドレスの１ビット目とに信号線の短絡（ショート）があった場合、対象アドレスに上記‘０１０１’を書き込んだだけでは異常検出できないし、その前後アドレスに上記‘０１０１’を書き込んでも異常検出できないが、その前後アドレスに上記‘１０１０’を書き込むことで対象アドレスデータが上記‘０１０１’から‘１１０１’に変化することになり、これを以って異常検出できる。このように、本手法では、対象アドレスのセルと前後アドレスのセルとの信号線間の短絡等も検出することができる。
【０１１５】
尚、信号線間の短絡は、上記前後アドレスに限らず、可能性としてはＲＡＭ内の他の全てのアドレスが有り得る。よって、信号線間の短絡故障を完全に検出しようとするならば、対象アドレスと他の全てのアドレスに対して上記複数のテストデータパターンを適用する必要があるが、当然、処理負荷が異常に増大し、処理時間が非常に掛かることになり、定周期の制御処理や安全機能の妨げとなり、現実的には実行困難である。
【０１１６】
本手法では、テストデータパターンの適用を、対象アドレスとその前後アドレスとすることで、処理負荷を増大させることなく高速に診断処理を行うことが可能となる。また、対象アドレスと前後アドレスのチェックだけでもある程度高い自己診断率を実現できる。特に、上記のように適切なテストデータパターンを用いることで、効率よく高い確率で故障検出することが可能となる。
【０１１７】

但し、上記の通り、例えば１ビット目と３ビット目とが短絡（ショート）していた場合のように、２つのテストパターンのみでは故障検出できない場合もある。この為、上記１０個のテストパターンを全て用いることが望ましい。これにより、例えば１ビット目と３ビット目とが短絡（ショート）していた場合、例えば16進数‘CCCC’＝‘１１００’を書き込むと‘１１１０’になるので、異常を検出できる。これは一例であり、他にも様々な故障があるが、上記１０個のテストパターンを全て用いることで殆どの故障に対応することができる。

【０１１８】

尚、上記Ｎ＝１０のテストデータパターンを用いる場合において、例えば上記１６進数‘５５５５’と‘ＡＡＡＡ’との組と同様の組（例えば、１６進数‘３３３３’と‘ＣＣＣＣ’との組（３＝‘００１１’、Ｃ＝‘１１００’）を用いてもよく、この様な組を例えば「ビットオンオフパターンが相互に異なる複数のテストパターン」または「各アドレスの全ビットのオンオフを実施させる複数のテストパターン」等と呼ぶものとする。

【０１１９】

本例では、「ビットオンオフパターンが相互に異なる複数のテストパターン」（各アドレスの全ビットのオンオフを実施させる複数のテストパターン）の一例として、上記の通り１６進数‘５５５５’と‘ＡＡＡＡ’の２つのテストデータパターンを用いる例に従って説明する。

【０１２０】

まず、上記の通り、最初は登録領域１が処理対象の記憶領域となり、その先頭アドレス（管理データ３２の登録領域先頭アドレス）が、最初の診断チェック対象アドレスとなる（ステップＳ５２）。

【０１２１】

そして、まず、対象アドレスとその前後のアドレスへの割り込みのマスクをする（ステップＳ５３）。これによりステップＳ５４以降の処理によって診断チェック対象アドレス等のデータを書き変えている間に、他の処理が対象アドレス等をアクセスしないようにする。

【０１２２】

続いて、対象アドレスとその前後のアドレスとの少なくとも３つ以上のアドレスのデータを、レジスタに退避する。更に、上記２つのテストデータパターンの一方（テストデータパターン１；本例では１６進数‘５５５５’）をテストデータとする（ステップＳ５４）。

【０１２３】

尚、上記「前後のアドレス」とは、本例では対象アドレスの直前のアドレスと直後のアドレスを意味するものとするが、この例に限らない。例えば、直前及びその前のアドレス、直後及びその後のアドレスの合計４つのアドレスを、前後のアドレスとしてもよい。この場合には、対象アドレスも含めて５つのアドレスのデータを、レジスタに退避することになる。また、この場合には、この５つのアドレスにテストデータパターンを書き込むことになる。

【０１２４】

そして、上記ステップＳ５４で設定したテストデータ（‘５５５５’）を対象アドレスに書き込む。これによって（異常が無ければ）図７（ａ）に示す状態となる。続いて、テストデータをテストデータパターン１（本例では１６進数‘５５５５’）とする（ステップＳ５５）。尚、ここではこの処理によってテストデータが変化するわけではないが（‘５５５５’のままである）、後述するように変化する場合がある。

【０１２５】

次に、対象アドレスの前後のアドレスに（上記の通り、ここでは対象アドレスの直前のアドレスと直後のアドレス）、上記ステップＳ５５で設定したテストデータ（‘５５５５’）を書き込む（ステップＳ５６）。これによって（異常が無ければ）図７（ｂ）に示す状態となる。更に、対象アドレスからデータを読み込む（ステップＳ５６）。この読込デ

ータは、異常が無ければ、上記ステップＳ５５で対象アドレスに書き込んだテストデータ（‘５５５５’）であるはずである。これより、ステップＳ５６で対象アドレスから読み込んだデータが、上記ステップＳ５５で対象アドレスに書き込んだテストデータ（‘５５５５’）と一致しているか否かを確認する（ステップＳ５７）。

【０１２６】

データ不一致の場合には（ステップＳ５７，ＮＯ）、ＲＡＭ２０（その対象アドレスのセル）は異常であり、異常終了の為のステップＳ６８～Ｓ７０の処理を実行する。すなわち、まず、対象アドレスとその前後アドレスのデータをレジスタから復元する（ステップＳ６８）。つまり、ステップＳ５４でレジスタに退避させていた対象アドレスのデータとその前後アドレスのデータを、それぞれ、対象アドレスとその前後アドレスに戻す。続いて、割り込みのマスクを解除する（ステップＳ６９）。最後に、所定のＲＡＭ異常検出に対応したＲＡＭ故障処理を行い（ここでは特に説明しない）（ステップＳ７０）、本処理を終了する。

【０１２７】

一方、データ一致の場合には（ステップＳ５７，ＹＥＳ）、ステップＳ５８へ移行する。

ステップＳ５８では、対象アドレスの前後のアドレスに対して、上記テストデータパターン２（本例では１６進数‘ＡＡＡＡ’）を適用済みか否かを判定する。ここでは、未だ、パターン２は適用していないので（ステップＳ５８，ＮＯ）、上記テストデータパターン２（本例では１６進数‘ＡＡＡＡ’）を新たなテストデータに設定して（ステップＳ６５）、ステップＳ５６に戻る。

【０１２８】

これより、ステップＳ５６の処理では、今度は、対象アドレスの前後のアドレスそれぞれに、上記ステップＳ６５で設定したテストデータ（‘ＡＡＡＡ’）を書き込む。これによって（異常が無ければ）図７（ｃ）に示す状態となる。上記の通りステップＳ５６では更に対象アドレスからデータを読み込む処理を行う。

【０１２９】

そして、再び上記ステップＳ５７の処理を行う。今度も、異常が無ければ、対象アドレスの格納データは‘５５５５’のままであるはずである。しかしながら、例えば対象アドレスと前後のアドレスとで配線が短絡等していることで、前後のアドレスに格納するデータによっては対象アドレスの格納データが変化する場合があり得る。

【０１３０】

例えば、まず、１６進数を４bit表記すると、例えば１６進数‘０’は‘0000’であり、１６進数‘Ｆ’は‘1111’であり、１６進数‘５’は‘0101’であり、１６進数‘Ａ’は‘1010’である。ここで、仮に、対象アドレスとその直前のアドレスとで最上位ビットの配線同士が短絡しているものとする。この場合、図７（ｂ）に示す状態では対象アドレスとその前のアドレスには両方とも１６進数‘５５５５’であるので、両方とも最上位ビットは‘０’であり、配線ショートによる影響は受けない（前後アドレスに‘５５５５’を格納しても対象アドレスの格納データは変化なし）。

【０１３１】

しかしながら、前後アドレスに１６進数‘ＡＡＡＡ’を格納すると、直前アドレスの最上位ビットが‘１’になるので、配線ショートによる影響により、対象アドレスも最上位ビットが‘１’になってしまう。つまり、‘0101’が‘1101’になってしまう（１６進数‘５’が１６進数‘Ｄ’になってしまう）。本手法によれば、例えばこの様な故障を確実に検出することができる。

【０１３２】

上記ステップＳ５７の判定結果がＹＥＳならば、再び上記ステップＳ５８の判定を行い、今度は既にパターン２を前後アドレスに適用済みなので判定ＹＥＳとなり（ステップＳ５８，ＹＥＳ）、ステップＳ５９へ移行する。ステップＳ５９では、対象アドレスに対して上記テストデータパターン２（本例では１６進数‘ＡＡＡＡ’）を適用済みか否かを判

定する。ここでは、未だ、パターン２は適用していないので（ステップＳ５９，ＮＯ）、上記テストデータパターン２（本例では１６進数‘ＡＡＡＡ’）を新たなテストデータに設定して（ステップＳ６４）、ステップＳ５５に戻る。

【０１３３】

これにより、上記ステップＳ５５の「対象アドレスにテストデータを書き込む」処理は、今度は対象アドレスに１６進数‘ＡＡＡＡ’を書き込む処理となり、その結果（正常であれば）図７（ｄ）に示す状態となる。上記の通りステップＳ５５では更にテストデータを１６進数‘５５５５’とする処理が行われるので、次のステップＳ５６の処理によって前後アドレスに１６進数‘５５５５’を書き込むことで、その結果（正常であれば）図７（ｅ）に示す状態となる。上記の通り、ステップＳ５６では更に対象アドレスからデータを読み出す。ここでは（正常であれば）１６進数‘ＡＡＡＡ’を読み出すことになる。

【０１３４】

続くステップＳ５７では、ステップＳ５６で対象アドレスから読み出したデータが、ステップＳ５５で対象アドレスに書き込んだデータ（ここでは１６進数‘ＡＡＡＡ’）と一致するか否かを確認する。上述したように、ＲＡＭ２０に異常が無ければデータ一致するはずである（ステップＳ５７，ＹＥＳ）。

【０１３５】

続くステップＳ５８では、対象アドレスの前後のアドレスに対して、上記テストデータパターン２（本例では１６進数‘ＡＡＡＡ’）を適用済みか否かを判定する。ここでは、未だ、パターン２は適用していないので（ステップＳ５８，ＮＯ）、上記テストデータパターン２（本例では１６進数‘ＡＡＡＡ’）を新たなテストデータに設定して（ステップＳ６５）、ステップＳ５６に戻る。

【０１３６】

これより、ステップＳ５６の処理では、今度は、対象アドレスの前後のアドレスそれぞれに、上記ステップＳ６５で設定したテストデータ（‘ＡＡＡＡ’）を書き込む。これによって（異常が無ければ）図７（ｆ）に示す状態となる。上記の通りステップＳ５６では更に対象アドレスからデータを読み込む処理を行う。

【０１３７】

そして、再び上記ステップＳ５７の処理を行う。今度も、異常が無ければ、対象アドレスの格納データは‘ＡＡＡＡ’のままであるはずである。しかしながら、例えば対象アドレスと前後のアドレスとで配線が短絡等していることで、前後のアドレスに格納するデータによっては対象アドレスの格納データが変化する場合があり得る。

【０１３８】

上記ステップＳ５７の判定処理でデータ一致した場合には（ステップＳ５７，ＹＥＳ）、本例では上記２つのテストパターンによるＲＡＭ診断は終了したので、ステップＳ５８、Ｓ５９の判定は何れもＹＥＳとなり、ステップＳ６０へ移行する。

【０１３９】

但し、テストデータパターンがＮ個ある場合は、Ｎ個のテストデータパターン全てを適用したか否かをチェックし、未終了の場合はテストデータパターンを変更してステップＳ５４から処理再開する（但し、データをレジスタに退避する処理は行わなくてよい（既に退避済みであるので））。例えば、16進数‘3333’を新たなテストデータパターン１とし、16進数‘CCCC’を新たなテストデータパターン２とする。この場合、処理再開直後のステップＳ５４～Ｓ５６の処理では、対象アドレス及びその前後アドレスに16進数‘3333’が書き込まれることになる。

【０１４０】

上記テストデータパターンがＮ個ある場合の処理は、別の説明の仕方をするならば例えば下記の通りとなる。

すなわち、変数ｎ（ｎ；１～Ｎ）を設け、ｎの初期値は‘１’とする。そして、テストデータパターンｎとテストデータパターンｎ＋１とをペアにして、１つのペアについて上記ステップＳ５４～Ｓ５９、Ｓ６４，Ｓ６５の一連の処理が完了したら、ｎを＋２インク

リメントすることで、テストデータパターンのペアを更新していく。この場合、ステップＳ５９がＹＥＳになったら更に「“ｎ＋１”＝Ｎ？」の判定を行い、判定ＮＯならば上記ペアの更新を行うようにしてもよい。尚、この場合、例えば図６におけるテストデータパターン１をテストデータパターンｎに置き換え、テストデータパターン２をテストデータパターンｎ＋１に置き換えてもよい。また、尚、この場合、ステップＳ６６の直後またはステップＳ６７の直後に、ｎを初期値‘１’へとリセットする。

【０１４１】

上記ステップＳ５９の判定がＹＥＳとなったら、まず、上記ステップＳ６８と略同様に、対象アドレスとその前後アドレスのデータを、レジスタから復元する（ステップＳ６０）。続いて、上記ステップＳ６９と略同様に、割り込みのマスクを解除する（ステップＳ６１）。そして、現在の対象アドレスが、現在の処理対象の登録領域（ここでは上記登録領域１）の最後のアドレス（管理データ３２の登録領域最終アドレス）であるか否かを判定する（ステップＳ６２）。

【０１４２】

つまり、現在の処理対象の登録領域内の全アドレスについてＲＡＭ診断が実行されたか否かを判定し、未だ最後のアドレスまで診断していないならば（ステップＳ６２，ＮＯ）、チェック対象を次のアドレスに更新する（ステップＳ６６）。つまり、現在の対象アドレスの“直後のアドレス”が、新たな対象アドレスとなる。そして、ステップＳ５３に戻り、上述した処理を繰り返す。

【０１４３】

ステップＳ６２の判定がＹＥＳとなったら、全ての登録領域についてＲＡＭ診断実行したか否かを確認し（ステップＳ６３）、未処理の登録領域がある場合には（ステップＳ６３，ＮＯ）、次の登録領域を処理対象として（例えば登録領域番号を更新して（＋１インクリメント等））(ステップＳ６７)、ステップＳ５２に戻る。本例では登録領域１に関する診断処理が終了したら、続いて登録領域２の診断処理を実行することになる。全ての登録領域についてＲＡＭ診断実行完了したならば（ステップＳ６３，ＹＥＳ）、本処理を終了する。尚、図３（ａ）に示す例では、登録領域２（登録領域番号＝‘２’の管理データ３２が示す領域）についてＲＡＭ診断実行完了した時点で、ステップＳ６３の判定がＹＥＳとなり、本処理を終了することになる。

【０１４４】

次に、図８を参照して、安全関連アプリケーション処理部１１’（安全関連アプリケーション処理１１）の処理について説明する。

図８は、安全関連アプリケーション処理部１１’の処理フローチャート図である。

【０１４５】

尚、図８に示すステップＳ８２、Ｓ８６の処理に係る関数（安全関連データアクセス関数１３）は、安全関連アプリケーション処理１１のプログラム内にインライン展開する。

図８において、安全関連アプリケーション処理部１１’は、所定のアプリケーション処理（制御処理等）を実行し（ステップＳ８１）、このアプリケーション処理に係り安全関連データをＲＡＭ２０から読込むイベントが発生した場合には上記安全関連データアクセス関数１３を呼び出し、上記図４で説明した各種パラメータを渡す（ステップＳ８２）。これによって、上述した図４の処理が実行されることになる。そして、上記の通り、正常終了／異常終了の何れかの関数値が返されてくるので、異常終了の場合は（ステップＳ８３，ＮＯ）所定の故障処理を実行し（この故障処理については特に説明しない）（ステップＳ８９）、本処理は終了する。

【０１４６】

正常終了の場合（ステップＳ８３，ＹＥＳ）上記アプリケーション処理を続行し（ステップＳ８４）、処理中に安全関連データをＲＡＭ２０に書き込むイベントが発生した場合には、安全関連データ領域２１に書き込むデータ等をパラメータとして設定し（図５で説明した各種パラメータの設定）（ステップＳ８５）、上記安全関連データアクセス関数１３を呼び出す（ステップＳ８６）。これにより上述した図５の処理が実行されることにな

る。

【０１４７】

そして、上記の通り、正常終了／異常終了どちらかの関数値が返されてくるので、異常終了であったならば（ステップＳ８７，ＮＯ）上記ステップＳ８９の処理を実行し、正常終了であったならば（ステップＳ８７，ＹＥＳ）上記アプリケーション処理を続行する（ステップＳ８８）。図示していないが、その後も、アプリケーション処理実行中に、ＲＡＭ２０に対する安全関連データの読込／書込イベントが発生する毎に、上述した関数１３の呼び出しの処理が行われ、アプリケーション処理が最後まで実行されたら本処理は終了する。

【０１４８】

以上説明した本例のＲＡＭ診断装置によれば、下記の様々な効果が得られる。

すなわち、本例のＲＡＭ診断装置によれば、ダブルＲＡＭ方式によるオンライン処理のタイミングでのＲＡＭ診断と、定周期起動でのＲＡＭテストパターン方式によるＲＡＭ診断（特に適切なテストパターンを用いる場合、例えば上述した例のように「全ビットのオン・オフ実施させる複数のテストパターン」または／且つ「ビットオン・オフパターンが相互に異なる複数のテストパターン」を用いる診断）の両方を行うことにより、高い自己診断率を実現し、故障による危害度の高い鉄道関係や化学プラント関係にも当該ＲＡＭ診断を適用したシステムを使用することができる。

【０１４９】

また、ＲＡＭテストパターン方式によるＲＡＭ診断において、診断チェック対象アドレスとその前後のアドレス等の３つ以上の連続したアドレス（対象アドレスを含む３つ以上の連続したアドレス）をテストパターン書込対象アドレスとして、対象アドレス以外のアドレス（例えば前後アドレス）のＲＡＭセルへのデータ書き込みによる対象アドレスのＲＡＭセルへの影響有無を診断することで、例えば信号線の短絡等の故障を確実に検出でき、高い自己診断率を実現できる。

【０１５０】

また、ダブルＲＡＭ方式と他の診断方式とを組み合わせたＲＡＭ診断として、例えば、ダブルＲＡＭ方式とギャルパット方式との組み合わせや、ダブルＲＡＭ方式とアブラハム方式との組み合わせ等も考えられる。これらの組み合わせの場合でも、ダブルＲＡＭ方式単体の場合に比べれば、自己診断率の高いＲＡＭ診断を実現することができる。但し、本例のダブルＲＡＭ方式とＲＡＭテストパターン方式との組み合わせによれば、これらギャルパット方式やアブラハム方式との組み合わせに比べて、容易な処理方法で診断実現でき処理の高速化を図ることができる。勿論、ダブルＲＡＭ方式単体の場合に比べて、自己診断率の高いＲＡＭ診断を実現することができる。

【０１５１】

上記処理の高速化は、例えば上述したような適切なテストパターンを用いることや、対象アドレスに係るテストパターン書込対象を「前後のアドレス」に限っている（他の全てのアドレス等とはしない）ことや、データ退避にレジスタを用いること等によって実現できる。

【０１５２】

更に、ＲＡＭテストパターン方式によるＲＡＭ診断の場合には、処理対象アドレスの格納データを退避させてから診断処理を行い診断処理完了したらデータ復旧させる（退避先から元に戻す）が、本手法では、上記３つ以上の連続したアドレスの格納データをレジスタに退避するので、高速なデータ退避およびデータ復旧が実現でき、ＣＰＵへの処理負荷が小さく、かつＲＡＭ診断処理が高速化することによりＲＡＭ診断処理によってＣＰＵが占有される時間が短縮することで安全機能への影響が小さくなる。また、従来ではデータ退避先としてＲＡＭ内に退避領域を設けていたが、本手法ではこの様なＲＡＭ退避領域が必要なくなる。

【０１５３】

また、ダブルＲＡＭ方式によるＲＡＭ診断において、ＲＡＭアクセス時にアクセス前に

、アクセス先アドレスに関して関数のパラメータのアドレスとＲＡＭ診断管理テーブルに登録されているアドレスとを比較チェックする機能を備えることにより、ＲＡＭ診断管理テーブルが配置されているメモリ（本例ではＲＯＭ１０）の異常または関数のパラメータが配置されているメモリの異常、パラメータのデータビット化け等を検出することができ、所定の故障処理（特に説明しない）を実行することで、当該異常が危険源となる故障を事前に除去することができる。

【０１５４】

本手法では、ダブルＲＡＭ方式のみではＲＡＭ故障診断を完全に担保できない（確実な故障検出を実現できない。または故障検出できるが早期検出が困難）という課題に対して、「データ／アドレスの直流化」の故障（固定（固着）故障、固定オープン、高インピーダンス、信号線間の短絡等）を検出できるＲＡＭテストパターン方式による診断を併用することで、確実かつ早期にＲＡＭ故障を検出できるようになる。

【０１５５】

また、ＲＡＭテストパターン方式を適用する際に、対象アドレスを含む連続した３つ以上のアドレス（例えば対象アドレスとその前後（隣接）アドレス等）に対して、複数のテストパターンを適用して対象アドレスを診断する方法を用いることで、対象アドレスに対する他のアドレス（例えば隣接したアドレス）のセルの影響をチェックできる。これによって、例えば上記隣接したアドレスのセルとの信号線間の短絡等を、確実に検出出来るようになる。

【０１５６】

また、当該診断方式において上記３つ以上の連続したアドレスのデータをレジスタに退避することで、ＲＡＭ退避領域の削減と、ＲＡＭデータの待避および復元処理を高速に行うことが実現できる。

【０１５７】

尚、ＲＡＭテストパターン方式による診断を適用してもオンライン処理のタイミングで‘０’固定、‘１’固定のＲＡＭ異常を検出できない可能性があるが、当該確率はダブルＲＡＭ方式のみを使用する場合と比較して大幅に減少する。

【０１５８】

ＲＡＭの各種の故障モードによる故障を複数の診断手法により検知することでＲＡＭ故障を確実に検出できる。例えば、ダブルＲＡＭ方式をオンラインの処理タイミングで適用し、かつＲＡＭテストパターン方式による診断を同一領域に対して適用する。これによりソフトエラーの検知とその他のＲＡＭの故障モードによる故障検知をすることができる。

【０１５９】

また、ＲＡＭテストパターン方式による診断と二重化ＣＰＵで同一処理を行って処理結果を突き合わせる診断とを組み合わせる方法、ＲＡＭテストパターン方式による診断と同一ＣＰＵで同一処理を複数回行って処理結果を突き合わせる診断とを組み合わせる方法などがある。

【０１６０】

なお、固定データの場合はあらかじめ固定データの領域を複数設定しそれぞれにＣＲＣデータなどのチェックコードを付加し、アクセスするタイミングでチェックコードのチェックを行うことで故障を検出できる。

【０１６１】

ＲＡＭテストパターン方式による診断にはIEC61508にギャルパット、アブラハム、ウオークパス、チェックボード、マーチなどの診断手法が例として記載されおり、必要とする自己診断率に合わせて診断手法を適用する。

【０１６２】

ダブルＲＡＭ方式をオンラインの処理タイミングで適用し、かつＲＡＭテストパターンによる診断を同一領域に対して適用する場合、ＲＡＭテストパターンによる診断を行い、かつ安全機能をプロセスの安全を担保できる時間内に実施するため、ＲＡＭテストパターンによる診断をダブルＲＡＭ方式による診断を行う処理より低いタスクレベルに割りつけ

る。この場合、テストのためにＲＡＭデータを書き変えている間は安全機能を実行できないため、当該時間を短くする処理方法をとる。当該例として本発明の実施例に示す方法やギャルパットやアブラハムで一度にテストするアドレスの範囲を１つのＲＡＭチップが実装されているアドレス範囲内に限定する方法などがある。

【０１６３】

本発明の実施例に示すＲＡＭテストパターンによる診断では、連続した３つ以上の診断対象アドレスに例えば、16進数‘5555’や16進数‘AAAA’などのテストパターンを書き込み、当該アドレスのデータを読み込む。これによりデータの直流化（固定故障、固定オープン、高インピーダンス、信号線間の短絡）、隣接するメモリセル間の干渉、アドレス化無し等の各種ＲＡＭ故障を検出することができる。

【０１６４】

本方式では当該アドレスのデータ退避について、上述したレジスタに退避する方法またはＲＡＭに退避領域を設けて退避する方法のいずれも採用することができる。但し、上記の通り、レジスタに退避する方法を採ることで高速化が図れる。ＲＡＭテストパターンによる診断に関して、診断対象アドレスとその前後アドレス等のような連続した３つ以上の処理対象アドレスの格納データの退避について、上記実施例では、レジスタに処理対象アドレスのデータを退避する方式を提案している。このことで退避用ＲＡＭ領域の削減と高速処理が実現できる。一般的にＣＰＵはアクセスのパフォーマンスをあげるためレジスタアクセスはＲＡＭアクセスより高速にアクセスできるよう設計されている。

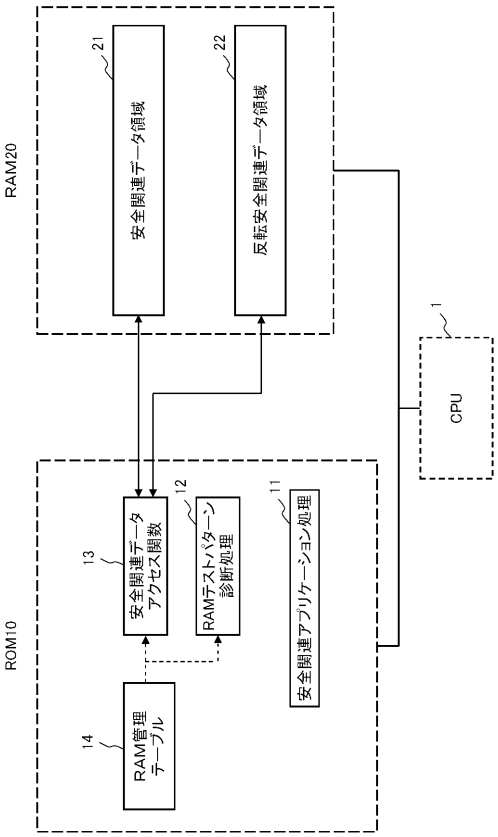
【符号の説明】

【０１６５】

１　ＣＰＵ
１０　ＲＯＭ
１１　安全関連アプリケーション処理
１２　ＲＡＭテストパターン診断処理
１３　安全関連データアクセス関数
１４　ＲＡＭ管理テーブル
１１’　安全関連アプリケーション処理部
１２’　ＲＡＭテストパターン診断処理部
１３’　安全関連データアクセス処理部
２０　ＲＡＭ
２１　安全関連データ領域
２２　反転安全関連データ領域
３０　ＲＡＭテストパターン方式診断管理テーブル
３１　ＲＡＭテストパターン方式診断登録領域番号
３２　管理データ
４０　ダブルＲＡＭ方式診断管理テーブル
４１　ダブルＲＡＭ方式診断ブロック番号
４２　管理データ

【図１】

本例のRAM診断装置の構成ブロック図



【図３】

(a)、(b)は、RAM管理テーブルの具体例を示す図

RAM管理テーブル 14

RAMテストパターン方式診断管理テーブル 30

31 RAMテストパターン方式診断登録領域番号	32 管理データ
1	登録領域先頭アドレス
	登録領域最終アドレス
2	登録領域先頭アドレス
	登録領域最終アドレス

(a)

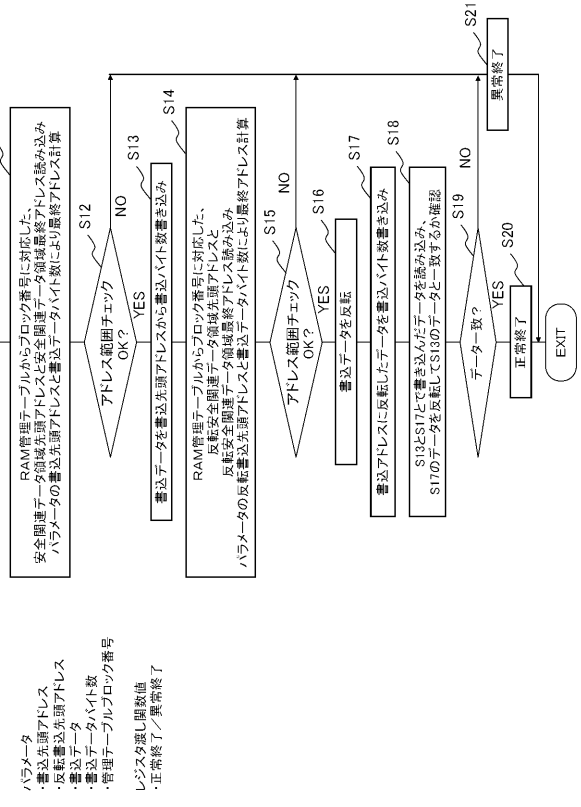
ダブルRAM方式診断管理テーブル 40

41 ダブルRAM方式診断ブロック番号	42 管理データ
1	安全関連データ領域先頭アドレス
	安全関連データ領域最終アドレス
	反転安全関連データ領域先頭アドレス
	反転安全関連データ領域最終アドレス
2	安全関連データ領域先頭アドレス
	安全関連データ領域最終アドレス
	反転安全関連データ領域先頭アドレス
	反転安全関連データ領域最終アドレス
3	安全関連データ領域先頭アドレス
	安全関連データ領域最終アドレス
	反転安全関連データ領域先頭アドレス
	反転安全関連データ領域最終アドレス

(b)

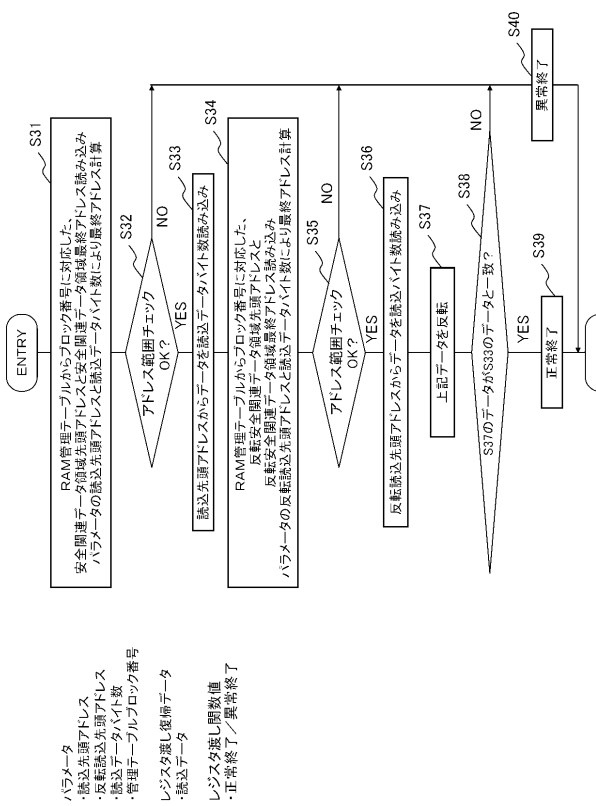
【図４】

安全関連データの書込処理のフローチャート図



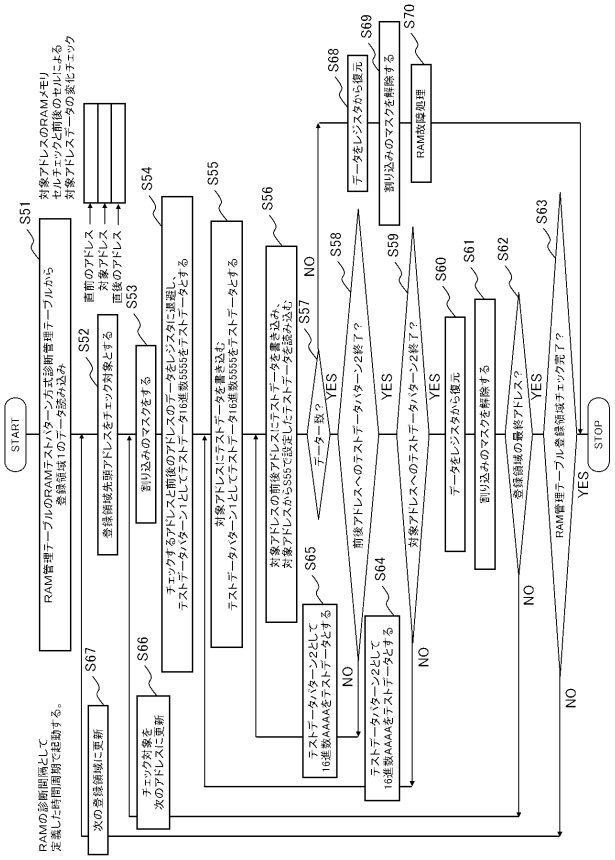
【図５】

安全関連データの読込処理のフローチャート図



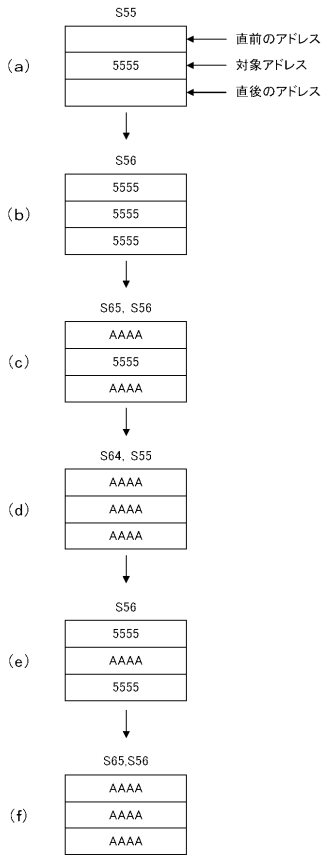
【図６】

RAMテストパターン診断処理のフローチャート図



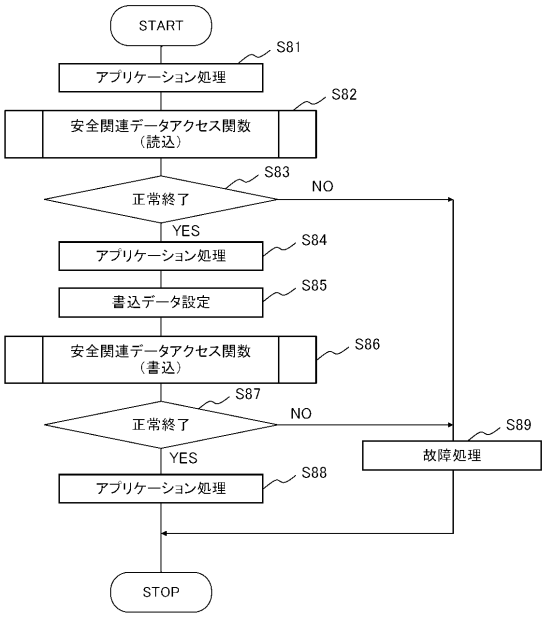
【図７】

(a)～(f)は、図6の処理中に書き込まれるテストパターンの推移を示す図



【図８】

安全関連アプリケーション処理部の処理フローチャート図



【図２】

本例のRAM診断処理のタスクスケジュールを示す図