(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2012-133458
(P2012-133458A)
(43)公開日　平成24年7月12日(2012.7.12)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 11/20	(2006.01)	G06F	11/20	310F	5B034
G06F 9/50	(2006.01)	G06F	9/46	462B	

審査請求 未請求　請求項の数 10　ＯＬ　（全 18 頁）

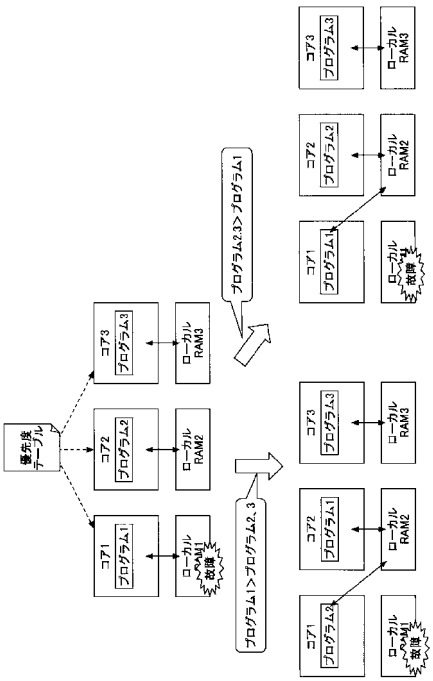
(21)出願番号　特願2010-283151 (P2010-283151)
(22)出願日　平成22年12月20日 (2010.12.20)

(71)出願人　000003207
トヨタ自動車株式会社
愛知県豊田市トヨタ町１番地
(74)代理人　100070150
弁理士　伊東　忠彦
(72)発明者　山田　一美
愛知県豊田市トヨタ町１番地　トヨタ自動車株式会社内

Ｆターム(参考)　5B034 BB11　CC01　CC02　DD01

(54)【発明の名称】マイコン、リソース割り当て方法

(57)【要約】
【課題】リソースを有効に活用できる信頼性の高いマイコンを提供すること。
【解決手段】ローカルメモリ１５とコア１３が対に配置され、コアが対のローカルメモリを作業メモリとするＣＰＵ１１を２つ以上有するマイコン１００であって、コア又はローカルメモリの異常を検出する異常検出手段４１と、各プログラム３１の優先度が登録された優先度テーブルと、コア又はローカルメモリの異常が検出された場合、優先度の高いプログラムほど優先的に、コアが該コアと対のローカルメモリを作業メモリとして実行するように、コアと作業メモリの組合せを変更するリソース割り当て変更手段４３と、を有することを特徴とする。
【選択図】図１



【特許請求の範囲】
【請求項１】
ローカルメモリとコアが対に配置され、コアが対のローカルメモリを作業メモリとしてプログラムを実行するＣＰＵを２つ以上有するマイコンであって、
コア又はローカルメモリの異常を検出する異常検出手段と、
各プログラムの優先度が登録された優先度テーブルと、
コア又はローカルメモリの異常が検出された場合、優先度の高いプログラムほど優先的に、コアが該コアと対のローカルメモリを作業メモリとして実行するように、コアと作業メモリの組合せを変更するリソース割り当て変更手段と、
を有するマイコン。
【請求項２】
前記リソース割り当て変更手段は、コアが該コアと対のローカルメモリを作業メモリとして実行できない優先度の低いプログラムを、コアが該コアと対でないローカルメモリを作業メモリとして実行するように、コアとローカルメモリの組合せを変更する、
請求項１記載のマイコン。
【請求項３】
前記異常検出手段が異常を検出した第１のコアが実行していた第１のプログラムの優先度が、第２のコアが実行している第２のプログラムよりも高い場合、
前記リソース割り当て変更手段は、前記第２のコアが前記第１のプログラムを前記第２のコアと対の第２のローカルメモリを作業メモリとして実行するようにコアと作業メモリの組合せを変更する、
請求項１又は２記載のマイコン。
【請求項４】
前記リソース割り当て変更手段は、前記第２のコアが前記第２のプログラムを前記第１のコアと対の第１のローカルメモリを作業メモリとして実行するように、コアと作業メモリの組合せを変更する、
請求項３記載のマイコン。
【請求項５】
前記異常検出手段が異常を検出した第１のコアが実行していた第１のプログラムの優先度が、第２のコアが実行している前記第２のプログラムよりも低い場合、
前記リソース割り当て変更手段は、前記第２のコアが第１のプログラムを前記第１のコアと対の第１のローカルメモリを作業メモリとして実行するように、コアと作業メモリの組合せを変更する、
請求項１～４いずれか１項記載のマイコン。
【請求項６】
前記異常検出手段が異常を検出した第１のローカルメモリを作業メモリとして第１のコアが実行していた第１のプログラムの優先度が、第２のローカルメモリを作業メモリとして第２のコアが実行していた第２のプログラムよりも高い場合、
前記リソース割り当て変更手段は、前記第２のコアが前記第１のプログラムを前記第２のコアと対の前記第２のローカルメモリを作業メモリとして実行するように、コアと作業メモリの組合せを変更する、
請求項１～５いずれか１項記載のマイコン。
【請求項７】
前記リソース割り当て変更手段は、前記第１のコアが前記第２のプログラムを前記第２のコアと対の前記第２のローカルメモリを作業メモリとして実行するように、コアと作業メモリの組合せを変更する、請求項６記載のマイコン。
【請求項８】
前記異常検出手段が異常を検出した第１のローカルメモリを作業メモリとして第１のコアが実行していた第１のプログラムの優先度が、第２のローカルメモリを作業メモリとして第２のコアが実行していた第２のプログラムよりも低い場合、

前記リソース割り当て変更手段は、前記第１のコアが前記第１のプログラムを前記第２のコアと対の前記第２のローカルメモリを作業メモリとして実行するように、コアと作業メモリの組合せを変更する、請求項６又は７記載のマイコン。
【請求項９】
前記異常検出手段がコアの異常を検出した場合、異常が検出されたコアを停止するコア停止手段と、
を有する請求項１～８いずれか１項記載のマイコン。
【請求項１０】
ローカルメモリとコアが対に配置され、コアが対のローカルメモリを作業メモリとするＣＰＵを２つ以上有するマイコンのリソース割り当て方法であって、
異常検出手段が、コア又はローカルメモリの異常を検出するステップと、
コア又はローカルメモリの異常が検出された場合、リソース割り当て変更手段が、各プログラムの優先度が登録された優先度テーブルを参照して、
優先度の高いプログラムほど優先的に、コアが該コアと対のローカルメモリを作業メモリとして実行するように、コアと作業メモリの組合せを変更するステップと、
を有するリソース割り当て方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、コアやＲＡＭの異常を検出可能なマイコンに関し、特に、プログラムをコアやマイコンに動的に割り当てるマイコン等に関する。
【背景技術】
【０００２】
車載装置の電子化が進み、車載装置を制御するための多くの電子制御装置が車両に搭載されるようになった。電子制御装置にはマイコンが搭載されマイコンには１つ以上のＣＰＵが搭載されるが、車載装置を制御するマイコンはその性質上、高い信頼性が要求される。
【０００３】
また、近年ＣＰＵのマルチコア化やマルチＣＰＵ化が進んでおり、マイコンが複数のコアやＣＰＵを有することを利用してマイコンの信頼性を向上させる技術が考案されている（例えば、特許文献１参照。）。特許文献１には、コアが故障した場合、プロセッサユニット内に代替実行可能なプロセッサコアが存在しなければ、最も優先度の低いプロセッサユニット内のプロセッサコアにアプリケーションを代替実行させるマルチプロセッサシステムが開示されている。このように、コアに故障が生じても、他の正常なコアが代替することで、マイコン全体の信頼性を向上させることができる。
【先行技術文献】
【特許文献】
【０００４】
【特許文献１】特開２００９－０６９９６３号公報
【発明の概要】
【発明が解決しようとする課題】
【０００５】
ところで、コアやＣＰＵがプログラムを実行する場合は、ＲＡＭなどの作業メモリが必要になるため、故障する可能性はコアやＣＰＵだけでなくＲＡＭにもある。しかし、特許文献１に記載されたマルチプロセッサシステムはＣＰＵとＲＡＭの故障の切り分けについて考慮されていないという問題がある。例えば、コアやＣＰＵに故障が生じたがＲＡＭに故障が生じていない場合、ＲＡＭを有効に活用することで高い信頼性を維持したままリソースを有効に活用できる。この逆に、ＲＡＭに故障が生じたがコアやＣＰＵに故障が生じていない場合も同様である。
【０００６】

本発明は、上記課題に鑑み、リソースを有効に活用できる信頼性の高いマイコンを提供することを目的とする。

【課題を解決するための手段】

【０００７】

本発明は、上記課題に鑑み、ローカルメモリとコアが対に配置され、コアが対のローカルメモリを作業メモリとするＣＰＵを２つ以上有するマイコンであって、コア又はローカルメモリの異常を検出する異常検出手段と、各プログラムの優先度が登録された優先度テーブルと、 コア又はローカルメモリの異常が検出された場合、優先度の高いプログラムほど優先的に、コアが該コアと対のローカルメモリを作業メモリとして実行するように、コアと作業メモリの組合せを変更するリソース割り当て変更手段と、を有するマイコンを提供する。

【発明の効果】

【０００８】

リソースを有効に活用できる信頼性の高いマイコンを提供することができる。

【図面の簡単な説明】

【０００９】

【図１】ローカルＲＡＭの異常が検出された場合のコアとローカルＲＡＭの組合せをプログラムの優先度に応じて変更する過程を説明する図の一例である。

【図２】コアの異常が検出された場合のコアとローカルＲＡＭの組合せをプログラムの優先度に応じて変更する過程を説明する図の一例である

【図３】マイコンのハードウェア構成図の一例である。

【図４】プログラム１～３の優先度が登録された優先度テーブルの一例である。

【図５】コア１～３の機能ブロック図の一例である。

【図６】異常の検出結果を模式的に示す図の一例である。

【図７】各コアに共通の割り当て制御の手順を示すフローチャート図の一例である。

【図８】マルチプロセッサ型のマイコンのハードウェア構成図の一例である。

【図９】マルチコア型又はマルチプロセッサ型のマイコンにおいて、コア１、２のフェールセーフ処理を１つのコアが集中的に制御する場合の機能ブロック図の一例である。

【図１０】コア３がコア１、２を監視する手順のシーケンス図の一例である。

【発明を実施するための形態】

【００１０】

以下、本発明を実施するための形態について図面を参照しながら説明する。

本実施形態では、コアに少なくとも１つのローカルＲＡＭが対に配置されたＣＰＵにおいて、何らかの異常が検出された場合に、コアがワークメモリとするローカルＲＡＭを、プログラムの優先度に応じて柔軟に組み替えるマイコンについて説明する。

【００１１】

図１は、ローカルＲＡＭの異常が検出された場合のコアとローカルＲＡＭの組合せをプログラムの優先度に応じて変更する過程を説明する図の一例である。３つのコア１～３がそれぞれローカルＲＡＭ１～３をワークメモリにしてプログラム１～３を実行している。また、予め優先度テーブルにプログラム１～３の優先度が定義されている。

【００１２】

例えばローカルＲＡＭ１の異常が検出された場合、コア１は優先度テーブルを参照して、コア２又はコア３がプログラム１よりも優先度の低いプログラムを実行しているか否かを判定する。

（１）コア２がプログラム１よりも優先度の低いプログラム２を実行している場合、コア１はローカルＲＡＭ２をワークメモリとしたまま優先度の低いプログラム２を自機に割り当てる。

【００１３】

また、コア２はローカルＲＡＭ２をワークメモリとして、優先度の高いプログラム１を自機に割り当てる。

【００１４】
プログラム２はプログラム１よりも優先度が低いので、コア１がコア２のローカルＲＡＭ２をワークメモリにプログラム２を実行することにより処理速度が低下しても、許容しやすい。また、プログラム１はプログラム２よりも優先度が高いが、コア２が対のローカルＲＡＭ２をワークメモリにしてプログラム１を実行するので、処理速度が低下することを防止できる。
（２）コア２又はコア３がプログラム１よりも優先度の低いプログラムを実行していない場合、コア１はローカルＲＡＭ２をワークメモリとしてプログラム１を実行する。
【００１５】
この場合、プログラム１が最も優先度が低いので、コア１がコア２のローカルＲＡＭ２をワークメモリにプログラム１を実行することにより処理速度が低下しても、許容しやすい。
【００１６】
図２は、コアの異常が検出された場合のコアとローカルＲＡＭの組合せをプログラムの優先度に応じて変更する過程を説明する図の一例である。コア１の異常が検出された場合、例えばコア２は優先度テーブルを参照して、コア２又はコア３がプログラム１よりも優先度の低いプログラムを実行しているか否かを判定する。
（３）例えばコア２がプログラム１よりも優先度の低いプログラム２を実行している場合、コア２はローカルＲＡＭ１をワークメモリとして優先度の低いプログラム２を実行する。また、コア２はローカルＲＡＭ２をワークメモリとして優先度の高いプログラム１を実行する。
【００１７】
プログラム２はプログラム１よりも優先度が低いので、コア２がコア１のローカルＲＡＭ１をワークメモリにプログラム２を実行することにより処理速度が低下しても、許容しやすい。また、プログラム１はプログラム２よりも優先度が高いが、コア２が対のローカルＲＡＭ２をワークメモリにしてプログラム１を実行するので、処理速度が低下することを防止できる。
（４）コア２又はコア３がプログラム１よりも優先度の低いプログラムを実行していない場合、コア２はローカルＲＡＭ１をワークメモリとしてプログラム１を実行する。
【００１８】
この場合、プログラム１が最も優先度が低いので、コア２がコア１のローカルＲＡＭ１をワークメモリにプログラム２を実行することにより処理速度が低下しても、許容しやすい。
【００１９】
このように、本実施形態のマイコンは、コア又はローカルＲＡＭに異常が生じても、異常が生じていないリソースを活用してプログラムの実行を継続できる。その際、マイコンは、プログラムの優先度を考慮して、コアが優先度の高いプログラムを優先的にコアと対のローカルＲＡＭをワークメモリにして実行するようにコアとワークメモリの組み合わせを決定する。
【００２０】
異常がない、コアと対のローカルＲＡＭの数よりもプログラムの数の方が多い場合、優先的に割り当てられないプログラムは、コアが対でないローカルＲＡＭをワークメモリにして実行する。
【００２１】
よって、優先度の高いプログラムの実行速度の低下を防止しながら、優先度の低いプログラムの実行を継続することができる。
【００２２】
〔構成〕
図３は、マイコン１００のハードウェア構成図の一例を示す。マイコン１００は、マルチレイヤバス２８を介して接続されたプロセッサ２１、ＤＭＡＣ２２、ＳＤＲＡＭ２３、

Ｉ／Ｏブリッジ２５及びＲＯＭ２４を有する。プロセッサ２１は、３つのＣＰＵ（以下、区別する場合ＣＰＵ１～３という）１１を有するが、ＣＰＵ１１の数は２つ以上であれば４つ以上でもよい。各ＣＰＵ１～３は、コア間通信部１２、リセット部１４、コア１３、及び、ローカルＲＡＭ１５を有する。ローカルＲＡＭ１５は１つのＣＰＵ１１内に複数個存在してもよい。なお、各ＣＰＵ１～３の機能を区別するため、ＣＰＵ内の各機能にＣＰＵ１～３と同じ番号を付す。

【００２３】

コア１～３は、ＰＣ（プログラムカウンタ）、命令バッファ、ＩＦＵ（Instruction Fetch Unit）、ＤＥＣ（DECoder）、ＲＦ（Register Fetch）、ＲＥＧ（REGister）、ＳＨ（Shifter）、ＡＬＵ、ＭＵＬ及びＦＰＵ等の演算回路、汎用レジスタ、及び、ロード／ストアユニット等を有する。これらが１クロック毎に命令実行段階の各ステージを実行することでいわゆるパイプライン制御が実現されている。

【００２４】

ローカルＲＡＭ１はコア１に専用のメモリ（一次キャッシュ）であり、ローカルＲＡＭ２はコア２に専用のメモリであり、ローカルＲＡＭ３はコア３に専用のメモリである。専用のメモリとは、コア１はローカルＲＡＭ１～３を使用することができるが、ローカルＲＡＭ１に比べ、コア１がローカルＲＡＭ２，３をワークメモリとする際には制限があることをいう。例えば、コア２，３によるローカルＲＡＭ２、３の使用許可が必要であったり、コア２，３がローカルＲＡＭ２，３の容量を所定量以上使用している場合は使用できなかったり、コア２，３がローカルＲＡＭ２，３にアクセスしている場合は待機する必要があったりすることである。また、コア１がローカルＲＡＭ２，３にアクセスする際は、コア通信バス１７又はマルチレイヤバス２８を経由する必要があるので、バスの調停など物理的な距離や構成に起因して、コア１のローカルＲＡＭ２，３に対するアクセスの優先度は低くなる。これも制限の１つである。

【００２５】

コア１～３は、ＲＯＭ２４に記憶されたフェールセーフプログラム３２及びＯＳ３３を実行する。プログラム３１（以下、区別する場合プログラム１～３という）は、ＯＳ３３などによりコア１～３に任意に割り当てられて、各コア１～３が実行することが可能である。起動時の割り当ては決まっていてもいなくてもよいが、起動時、コア１～３は専用のローカルＲＡＭ１～３をワークメモリとする。以下では、コア１～３と専用のローカルＲＡＭ１～３との組み合わせを「専用組み合わせ」という。

コア間通信部１～３は、コア通信バス１７を介してコア１～３が相互に通信することを可能にする。コア１～３は、コア間通信部１～３により、他のコアの異常の有無を監視したり、ローカルＲＡＭ１～３にアクセスする。コア間通信部１～３はローカルＲＡＭ１～３を介してデータを相互に通信することができる。また、コア間通信部１～３は、専用組み合わせでないコア１～３からローカルＲＡＭ１～３へのアクセス要求を遮断するスイッチや、専用組み合わせでないコア１～３からアクセス可能なローカルＲＡＭ１～３の領域を一部に制限するアクセス制限の機能を備える。なお、コア１～３は、マルチレイヤバス２８を経由して互いに通信することもできる。

【００２６】

リセット回路１～３は、自コア又は他コアをリセットする回路である。自コアをリセットするのは例えばマイコン１００の起動時にＣＰＵ１～３をそれぞれ自コアが初期化してＯＳ３３等を起動するためである。他コアをリセットするのは、主に他コアを停止させるためである。リセット回路１はリセット回路２，３にリセット信号（Highアクティブでも Lowアクティブでもよい）を継続的に出力することができる（結線は不図示）。リセット回路２，３についても同様である。リセット信号が継続的に出力されるとクロック信号が停止したままになるので、リセット信号が入力されたコア２，３は停止したままとなる。

【００２７】

ＤＭＡＣ２２は、マルチレイヤバス２８に接続された各ブロックからのＳＤＲＡＭ２３へのアクセス要求を調停すると共に、Ｉ／ＯブリッジからＳＤＲＡＭ２３へ又はＳＤ

ＲＡＭ２３からＩ／Ｏブリッジ２５へ、コア１～３を介在することなくデータを転送する。ＤＭＡＣ２２は転送終了によりコア１～３に適宜割込みをかけることできる。

【００２８】

ＳＤＲＡＭ２３は、コア１～３にプログラム又はデータを提供するためのメモリ（二次キャッシュ）である。ＳＤＲＡＭ２３の少なくとも一部はコア１～３により共有される。ＲＯＭ２４にはプログラム１～３、フェールセーフプログラム３２、及び、ＯＳ３３が記憶されている。説明の都合上、プログラム１～３は３つとしたがプログラムの数は４つ以上でもよい。この場合、複数のプログラム３１が割り当てられるコア１３が存在することになる。

【００２９】

Ｉ／Ｏブリッジ２５は、マルチレイヤバス２８とＩ／Ｏバス２７の間で相互に周波数や電圧値などを変換する。Ｉ／Ｏバス２７には種々のＩ／Ｏ２６が接続されている。Ｉ／Ｏ２６はマイコン１００と外部の機器を接続するインタフェースであり、各種のセンサ、アクチュエータ、ＣＡＮコントローラ等が接続される。なお、このマイコン１００は例えば車両のＥＣＵ（electronic control unit）に適用される。

【００３０】

〔プログラムのコアへの割り当て〕

本実施形態のマイコン１００は、プログラム１～３の優先度を考慮してプログラム１～３が実行されるコア１～３とローカルＲＡＭ１５の組合せを任意に変更するので、いわゆるＳＭＰ（Symmetric Multi‐processing）の実行態様といえる。または、あるプログラムだけ割り当てられるコアを固定にしておき、その他のプログラムをマイコン１００がコア１～３に動的に割り当てることもできる（ＢＭＰ（Bound Multi‐processing））。

【００３１】

本実施形態のマイコン１００は、ＯＳ３３がコア１～３とローカルＲＡＭ１５の組合せを制御するものとするが、この機能はプログラム３１が提供してもよい。ＯＳ３３は、プログラム１～３の優先度と状態（実行可能、実行中、実行待ち等）に応じてコア１～３にプログラム１～３を割り当てる。プログラム１～３は、互いに独立して実行可能な場合も、相互に関連している場合もある。前者の場合、ＯＳ３３は優先度にだけ基づいてコア１～３にプログラム１～３を割り当てればよい。

【００３２】

図４（ａ）は、プログラム１～３の優先度が登録された優先度テーブルの一例を示す。優先度テーブルはＲＯＭ２４に記憶されており、コア１～３が参照することができる。図では、プログラム１の優先度を「高」、プログラム２の優先度を「中」、プログラム１の優先度を「低」としたがあくまで一例である。

【００３３】

後者の場合（相互に関連している場合）が生じるのは、例えばプログラム２がプログラム１の処理結果を利用する場合等である。このような場合、プログラム２はサービスコールをＯＳ３３に要求して、実行待ち状態となる。プログラム１は処理が完了すると同様にサービスコールをＯＳ３３に要求してプログラム２を実行可能状態にする。ＯＳ３３は、このような状態遷移を監視して、優先度を考慮しながらコア１～３にプログラム１～３を割り当てていく。

【００３４】

図４（ｂ）は、ＯＳ３３が割り当てたコア１～３とプログラム１～３の関係の一例を示す。コアとプログラムの数が等しいので、プログラム１～３に優先度の違いがあっても、各コア１～３に１つのプログラム１～３が割り当てられている。コア１～３が実行しているプログラム１～３の情報はコア間通信部１～３を介して交換されるので、各コア１～３は最新の割り当て状況を共有することができる。

【００３５】

なお、一般的なＳＭＰではいずれかのコアがＯＳ３３を実行して、ＯＳ３３が複数のコアへのプログラムの割り当てを制御するが、こうするとＯＳ３３を実行するコア１３に異

常が生じた場合に、コア１～３とワークメモリの組合せの変更が困難になる。そこで、本実施形態は、コア１～３とワークメモリの組合せを変更する機能（後述する割り当て制御部）は各コア１～３が有しているものとする。

【００３６】

具体的には、例えば、コア１～３又はローカルＲＡＭ１～３の異常を検出したコア１３が、ＯＳ３３又はＯＳ３３と同様の機能を使って、異常が検出されたコア１～３又はローカルＲＡＭ１～３で実行されていたプログラム１～３を、その優先度に応じて、専用割り当て又は専用割り当てでない組み合わせのコア１３とローカルＲＡＭ１５に割り当てる。

【００３７】

〔機能ブロック図〕

図５は、コア１～３の機能ブロック図の一例である。コア１～３はそれぞれ、異常監視部４１、停止部４２、及び、割り当て制御部４３を有する（各ＣＰＵの機能を区別するため、機能ブロックにＣＰＵと同じ番号を付す。）。各機能ブロックは、コア１～３がフェールセーフプログラム３２及びＯＳ３３の少なくとも一方を実行することで実現される。

【００３８】

＜異常監視部＞

異常監視部４１は、コア１３及びローカルＲＡＭ１５の異常を監視する。監視形態はいくつか考えられるが、１つの異常監視部４１はコア１～３の全ての異常を監視する能力、及び、ローカルＲＡＭ１～３の全ての異常を監視する能力を有するものとし、適宜、監視形態を変える。

【００３９】

（１）監視形態

Ａ．コア１～３及びＲＡＭ１～３が正常な場合

・コアの監視

異常監視部１はコア２の異常を、異常監視部２はコア３の異常を、異常監視部３はコア１の異常を、それぞれ監視する。コア１３に異常が生じると異常監視部４１の監視結果の信頼性も低下するので、各異常監視部１～３が隣のコア１～３の異常を監視することで、異常のあるコア自身が自機の異常を監視することを回避している。

・ローカルＲＡＭの監視

異常監視部１はローカルＲＡＭ１の異常を、異常監視部２はローカルＲＡＭ２の異常を、異常監視部３はローカルＲＡＭ３の異常を、それぞれ監視する。専用組み合わせのコア１３とローカルＲＡＭ１５はプログラムの実行速度が速いので、このように監視することで負荷を抑制しやすい。コアの監視のように隣のローカルＲＡＭを監視してもよい。

【００４０】

Ｂ．いずれかのコアの異常が検出された場合

・コアの監視

例えば、コア１の異常が検出されたとして説明する。コア１に異常が生じることにより、異常監視部１はコア２の異常を監視できないが、異常監視部１を監視していた異常監視部３はもはやコア１の異常を監視する必要がない。そこで、異常監視部３はコア２の異常を監視する。異常監視部２はコア３の異常を監視したままである。このように、コア１の異常が検出されても、異常監視部１～３が監視対象を切り替えることでコア２，３の監視を継続できる。

・ローカルＲＡＭの監視

例えば、コア１の異常が検出されたとして説明する。コア１の異常が生じることにより、異常監視部１はローカルＲＡＭ１の異常を監視できない。このため、異常監視部３が、コア１と共にローカルＲＡＭ１の異常を監視する。すなわち、異常監視部２はローカルＲＡＭ２の異常を監視したままであり、異常監視部３はローカルＲＡＭ３とローカルＲＡＭ１の異常を監視する。このように、コア１の異常が検出されても、異常監視部１～３が監視対象を増やすことでローカルＲＡＭ１～３の監視を継続できる。

【００４１】

Ｃ．いずれかのローカルＲＡＭの異常が検出された場合
・コアの監視
コア１～３は正常なので、異常監視部１～３によるコア１～３の監視形態に変更はない。
・ローカルＲＡＭの監視
例えば、ローカルＲＡＭ１の異常が検出されたとして説明する。すでに、ローカルＲＡＭ１の異常が検出されたので、異常監視部１はローカルＲＡＭ１の監視を終了する。したがって、引き続き、異常監視部２はローカルＲＡＭ２を監視し、異常監視部３はローカルＲＡＭ３を監視する。

【００４２】

このように、コア１～３の１つ又はローカルＲＡＭ１～３の１つに異常が検出されても、異常監視部１～３は異常の監視を継続できる。

【００４３】

（２）異常の監視方法
Ａ．コアの異常

コアの異常を検出する方法としては、コア同士の処理結果を比較する方法が知られている。例えば、異常監視部１は、コア２に診断プログラムの実行を要求し、同じ診断プログラムをコア１で実行して処理結果を比較する。コア１～３には、ＳＨ、ＡＬＵ、ＭＵＬ及びＦＰＵ等の演算器が実装されているので、コアを監視するためにはこれら全ての演算器の異常を監視することが好ましい。このため、診断プログラムには、各演算器のみを１回以上使用するいくつかの関数が記述されており、各演算器毎の関数の処理結果を別々に出力するようになっている。

【００４４】

異常監視部１は、コア２とコア１の演算器毎の処理結果を比較して、演算器毎に異常の有無を監視することができる。個別の演算器の異常が検出されれば、後の解析に有効な情報となる。本実施形態では、１つでも演算器に異常が検出されればコアに異常があると判定する。

【００４５】

異常監視部１～３は、コア１～３に異常があると判定すると、コア間通信部１～３を介して異常が検出されたコア１３をコア１～３を通知する（例えば、コア１にコア１の異常を通知する必要はない。）。これにより、各コア１～３が異常状態を共有できる。

【００４６】

なお、コア１～３による診断プログラムの処理結果は、コアが正常であれば固定の値を取るので、異常監視部１は、コア２に診断プログラムの実行を要求し、その処理結果と予め記憶している診断プログラムの処理結果とを比較してもよい。コア１が診断プログラムを実行する必要がないので負荷を低減できる。

【００４７】

異常を監視するタイミングは、例えば、監視対象のコア１～３の負荷状態が比較的低い場合や、負荷状態が低くなくても定期的なタイミング等である。負荷状態を知らせるため、各コアは例えばコアの使用率が所定値以下になると、他のコアに負荷状態が低下したことを通知する。

【００４８】

Ｂ．ローカルＲＡＭの異常

ローカルＲＡＭ１～３の異常を検出する方法には種々の方法がある。例えば、“０”を書き込んで“１”のままのbitがあるか否か（又はその逆）、なんらかのデータを読み書きして読み書きの前後で一致するか否か、全領域に“1010…”のパターンのデータを書き込み読み出せるか否か、メモリ上のまとまった領域を別の領域に正しくコピーできるか否か、等の方法がある。異常監視部１～３は、ローカルＲＡＭ１～３に対しこのような診断を行い、異常の有無を判定する。

【００４９】

異常監視部１～３は、ローカルＲＡＭ１～３の異常が検出された場合、初期化するなど

の回復を試みて異常が解消されない場合はフェールセーフを行う。例えば、コア間通信部１～３を構成する複数のチップの１つを分離可能であれば、異常が生じたチップを使用しないようにする。このような分離が困難であったあり、又は、チップの一部を分離可能でもローカルＲＡＭ１～３の所定の割合以上が使用不可になるような場合、異常監視部１～３はローカルＲＡＭ１～３に異常があると判定する。

【００５０】

異常監視部１～３は、ローカルＲＡＭ１～３に異常があると判定すると、コア間通信部１～３を介して異常が検出されたローカルＲＡＭ１～３をコア１～３に通知する。これにより、各コア１～３が異常状態を共有できる。

【００５１】

図６（ａ）は、異常の検出結果を模式的に示す図の一例である。コア１～３は、コア１～３又はローカルＲＡＭ１～３の異常の通知を受け付けると、現在の異常状態を更新して記憶する。したがって、各コア１～３は、図６（ａ）のような異常状態を共有して、異常が検出されていないコアやローカルＲＡＭを組合せて、プログラム１～３を割り当てることができる。

【００５２】

異常を監視するタイミングは、例えば、マイコン１００の起動直後や、自コアの負荷状態が比較的低い場合などである。負荷状態は例えばコアの使用率に基づいて判定する。

【００５３】

＜停止部＞

図５に戻り、停止部１～３は異常が検出されたコア１～３を停止する。停止部１～３と、停止部１～３が停止するコア１～３の関係は、異常監視部１～３が監視する対象のコア１～３と同じである。すなわち、停止部１はコア２に異常が検出された場合コア２を停止させ、停止部２はコア３に異常が検出された場合コア３を停止させ、停止部３はコア１に異常が検出された場合コア１を停止させる。

【００５４】

異常監視部１～３が監視しているコア１～３は既知なので、自コアの異常監視部１～３が監視しているコア１～３に異常が検出されたことが図６（ａ）のような異常状態から明らかになると、停止部１～３は異常が検出されたコアを停止する。停止するには上記のリセット回路１～３を作動させ、停止対象のコア１３にリセット信号を継続的に出力すればよい。

【００５５】

停止部１～３はコア１～３を停止させると、又は、コア１～３を停止させる所定時間前に、コア間通信部１～３を介して停止させたコア１～３をコア１～３に通知する（例えば、コア１にコア１の停止を通知する必要はない。）。所定時間前に通知することで、コアが停止する前に、他のコアが必要なデータをローカルＲＡＭ１５から取得することができる。図６（ｂ）は、停止状態のコアが登録された図の一例である。これにより、各コア１～３が停止されたコアを共有できる。

【００５６】

＜割り当て制御部＞

図５に戻り、割り当て制御部１～３は、コア１～３やローカルＲＡＭ１～３の異常状態に更新があると、コア１～３とローカルＲＡＭ１～３の組合せの制御を開始する。

【００５７】

図７は、各コアに共通の割り当て制御の手順を示すフローチャート図の一例である。異常監視部１～３は、それぞれ個別にコア１～３及びローカルＲＡＭ１～３の異常を検出している（Ｓ１０）。

【００５８】

まず、コア１～３のいずれかの異常が検出された場合（Ｓ２０のＹｅｓ）、割り当て制御部１～３は、異常が検出されたコア１～３が実行していたプログラム１～３と、他のコア１～３が実行しているプログラム１～３の優先度を比較する（Ｓ３０）。この判定を行

うのは、異常が検出されたコア１～３を監視していたコア１～３の割り当て制御部４３である。
・コア１の異常：割り当て制御部３は、プログラム１と、プログラム２，３の優先度を比較する。
・コア２の異常：割り当て制御部１は、プログラム２と、プログラム１，３の優先度を比較する。
・コア３の異常：割り当て制御部２は、プログラム３と、プログラム２，３の優先度を比較する。
【００５９】
異常が検出されたコア１～３が実行していたプログラム１～３よりも、優先度の低いプログラム１～３がない場合（Ｓ４０のＮｏ）、異常が検出されたコア１～３が実行していたプログラム１～３の優先度が最も低いことになるので、割り当て制御部１～３は異常が検出されたコア１～３が実行していたプログラム１～３を自コアに割り当てる（Ｓ５０）。ただし、ワークメモリは元のローカルＲＡＭ１～３のままとする。
・コア１の異常：割り当て制御部３は、ワークメモリをローカルＲＡＭ１としたままプログラム１をコア３に割り当てる。
・コア２の異常：割り当て制御部１は、ワークメモリをローカルＲＡＭ２としたままプログラム２をコア１に割り当てる。
・コア３の異常：割り当て制御部３は、ワークメモリをローカルＲＡＭ３としたままプログラム３をコア２に割り当てる。
【００６０】
こうすることで、実行速度は低下しても優先度の低いプログラム１～３を継続して実行することができる。
【００６１】
異常が検出されたコア１～３が実行していたプログラム１～３よりも、優先度の低いプログラムがある場合（Ｓ４０のＹｅｓ）、割り当て制御部１～３は優先度のより高いプログラム１～３が専用組み合わせのコアとローカルＲＡＭ１～３で実行されるように、コア１～３とコア１～３がワークメモリとするローカルＲＡＭ１～３との組合せを変更する（Ｓ６０）。この場合、専用組み合わせでないコアとローカルＲＡＭで実行されるのは、最も優先度の低いプログラム３１である。
・コア１の異常
プログラムＰ１、P2＞Ｐ３の場合
割り当て制御部３は、ローカルＲＡＭ３をワークメモリにしてプログラム１をコア３に割り当てる。割り当て制御部３は、プログラム３（低）をコア３に割り当てたまま、ワークメモリをローカルＲＡＭ１に変更する。
プログラムＰ１、P3＞Ｐ２の場合
割り当て制御部３は、ローカルＲＡＭ２をワークメモリにしてプログラム１をコア２に割り当てる。割り当て制御部３は、プログラム２（低）をコア２に割り当てたまま、ワークメモリをローカルＲＡＭ１に変更する。
・コア２の異常
プログラムＰ２、P3＞Ｐ１の場合
割り当て制御部１は、ローカルＲＡＭ１をワークメモリにしてプログラム２をコア１に割り当てる。割り当て制御部１は、プログラム１（低）をコア１に割り当てたまま、ワークメモリをローカルＲＡＭ２に変更する。
プログラムＰ２、P1＞Ｐ３の場合
割り当て制御部１は、ローカルＲＡＭ３をワークメモリにしてプログラム２をコア３に割り当てる。割り当て制御部１は、プログラム３（低）をコア３に割り当てたまま、ワークメモリをローカルＲＡＭ２に変更する。
・コア３の異常
プログラムＰ１、P3＞Ｐ２の場合

割り当て制御部２は、ローカルＲＡＭ２をワークメモリにしてプログラム３をコア２に割り当てる。割り当て制御部２は、プログラム２（低）をコア２に割り当てたまま、ワークメモリをローカルＲＡＭ３に変更する。
プログラムＰ２、P３＞Ｐ１の場合
割り当て制御部２は、ローカルＲＡＭ１をワークメモリにしてプログラム３をコア１に割り当てる。割り当て制御部２は、プログラム１をコア１に割り当てたまま、ワークメモリをローカルＲＡＭ３に変更する。
【００６２】
こうすることで、優先度のより高いプログラム１～３は専用組み合わせで処理されるので実行速度の低下を防止でき、優先度のより低いプログラム１～３も継続して実行することができる。
【００６３】
なお、コア１～３にプログラム１～３を割り当てるとは、各コア１～３の割り当て制御部１～３の監視対象にプログラム１～３を登録することであり、具体的には、プログラム１～３のＩＤや先頭アドレスを他のコアの割り当て制御部１～３に通知することである。割り当て制御部１～３は、実行可能状態のプログラム１～３を実行キューに登録して順番に実行する。
【００６４】
また、コア１～３に異常が検出された場合、コア１～３が複数のプログラム１～３を実行することになるので、割り当て制御部１～３は優先度の高いプログラムの実行頻度を優先度の低いプログラムよりも大きくする。
【００６５】
また、コア１～３がワークメモリとして使用するローカルＲＡＭ１～３を変更するとは、変更前のローカルＲＡＭ１～３から移動先のローカルＲＡＭにコア１～３のアクセス先を移動することである（プログラムやデータを移動する）。プログラム１～３は、ベースレジスタの値からのオフセットを指定することでアクセスされる再配置可能なプログラムである。割り当て制御部１～３はこれを利用して、プログラム１～３が専用組み合わせのコア１～３とローカルＲＡＭ１～３で実行される場合は、ベースレジスタにローカルＲＡＭ１～３におけるプログラム１～３の先頭アドレスを設定する。
【００６６】
コア１～３が専用組み合わせでないローカルＲＡＭ１～３をワークメモリとする場合、例えば、プログラムによってはコア間通信部１～３やマルチレイヤバス２８を介してローカルＲＡＭ１～３にアクセスする必要がある。この場合、割り当て制御部１～３は、ベースレジスタにローカルＲＡＭ１～３を指定するアドレスとローカルＲＡＭ１～３におけるプログラムの先頭アドレスを組み合わせて設定する。これにより、コア１～３は、コア間通信部１～３又はマルチレイヤバス２８を介して他コアのローカルＲＡＭ１～３にアクセスすることができる。
【００６７】
次に、ローカルＲＡＭ１～３のいずれかの異常が検出された場合（Ｓ１２０のＹｅｓ）、割り当て制御部１～３は、異常が検出されたローカルＲＡＭ１～３をワークメモリにしていたプログラム１～３と、他のプログラム１～３の優先度を比較する（Ｓ１３０）。この判定を行うのは、ローカルＲＡＭ１～３の異常を検出したコアの割り当て制御部４３である。
・ローカルＲＡＭ１の異常
割り当て制御部１は、プログラム１と、プログラム２，３の優先度を比較
・ローカルＲＡＭ２の異常
割り当て制御部２は、プログラム２と、プログラム１，３の優先度を比較
・ローカルＲＡＭ３の異常
割り当て制御部３は、プログラム３と、プログラム１，２の優先度を比較
異常が検出されたローカルＲＡＭ１～３をワークメモリにしていたプログラム１～３よ

りも優先度が低いプログラム１～３がない場合（Ｓ１４０のＮｏ）、異常が検出されたローカルＲＡＭ１～３をワークメモリとするプログラム１～３の優先度が最も低いことになるので、割り当て制御部１～３は異常が検出されたローカルＲＡＭ１～３をワークメモリとするプログラム１～３を、他コアがワークメモリとするローカルＲＡＭ１～３をワークメモリにして実行を継続する（Ｓ１５０）。この「他コア」は例えばコア間通信部１～３の空き容量が最も大きいコアである。または、比較的低い優先度のプログラム１～３を実行しているコアでもよい。ここでは、監視対象のコア１～３に割り当てるとする。

・ローカルＲＡＭ１の異常

割り当て制御部１は、プログラム１のワークメモリをローカルＲＡＭ２に変更する。

・ローカルＲＡＭ２の異常

割り当て制御部２は、プログラム２のワークメモリをローカルＲＡＭ３に変更する。

・ローカルＲＡＭ３の異常

割り当て制御部３は、プログラム３のワークメモリをローカルＲＡＭ１に変更する。

【００６８】

こうすることで、各コア１～３は、実行速度は低下しても優先度の低いプログラム１～３を継続して実行することができる。

【００６９】

異常が検出されたローカルＲＡＭ１～３をワークメモリにしていたプログラム１～３よりも優先度が低いプログラム１～３がある場合（Ｓ１４０のＹｅｓ）、割り当て制御部１～３は優先度のより高いプログラム１～３を、コア１～３が専用組み合わせのローカルＲＡＭ１～３をワークメモリとして実行するように、コア１～３とローカルＲＡＭ１～３の組合せを変更する（Ｓ１６０）。この場合、専用組み合わせでないコアとローカルＲＡＭで実行されるのは、最も優先度の低いプログラム３１である。

・ローカルＲＡＭ１の異常

プログラム１、３＞プログラム２

割り当て制御部１は、最も優先度の低いプログラム２を実行しているコア２にプログラム１を割り当て、専用組み合わせになるようにワークメモリをローカルＲＡＭ２に変更する。また、割り当て制御部１は、ワークメモリを変更することなく、プログラム１の割り当て先のコア２が実行していたプログラム２を自コア（コア１）に割り当てる。

プログラム１、２＞プログラム３

プログラム２、コア２，ローカルＲＡＭ２を、プログラム３、コア３，ローカルＲＡＭ３で置き換えた関係になる。

・ローカルＲＡＭ２の異常

プログラム１，２＞プログラム３

割り当て制御部２は、最も優先度の低いプログラム３を実行しているコア３にプログラム２を割り当て、専用組み合わせになるようにワークメモリをローカルＲＡＭ３に変更する。また、割り当て制御部２は、ワークメモリを変更することなく、プログラム２の割り当て先のコア３が実行していたプログラム３を自コア（コア２）に割り当てる。

プログラム２、３＞プログラム１

プログラム３、コア３，ローカルＲＡＭ３を、プログラム１、コア１，ローカルＲＡＭ１で置き換えた関係になる。

・ローカルＲＡＭ３の異常

プログラム２，３＞プログラム１

割り当て制御部３は、最も優先度の低いプログラム１を実行しているコア１にプログラム３を割り当て、専用組み合わせになるようにワークメモリをローカルＲＡＭ１に変更する。また、割り当て制御部３は、ワークメモリを変更することなく、プログラム３の割り当て先のコア１が実行していたプログラム１を自コア（コア３）に割り当てる。

プログラム１，３＞プログラム２

プログラム１、コア１，ローカルＲＡＭ１を、プログラム２、コア２，ローカルＲＡＭ２で置き換えた関係になる。

【００７０】
こうすることで、優先度のより高いプログラム１～３は専用組み合わせで処理されるので実行速度の低下を防止でき、優先度のより低いプログラム１～３も継続して実行することができる。
【００７１】
また、ローカルＲＡＭ１～３に異常が検出された場合、複数のプログラム１～３が１つのローカルＲＡＭ１～３を共有することになるので、割り当て制御部１～３はローカルＲＡＭ１～３の領域毎にプログラム１～３を割り当てる。この際、優先度の高いプログラム１～３に多くの領域を割り当てることで（例えば、優先度が高と中又は中と低の組み合わせでは３対２、優先度が高と低の組み合わせでは３対１などとする）、メモリスワップを少なくでき、優先度の高いプログラム１～３をより高速に実行できるようになる。なお、領域の割り当ては、例えばコア間通信部１～３により実現される。
【００７２】
なお、ローカルＲＡＭ１～３の一部の異常が検出された場合に、一部のプログラムだけを移動することもできる。例えば、コア１がローカルＲＡＭ１をワークメモリにしてプログラム１，４を実行中、ローカルＲＡＭ１の一部に異常が検出された場合、優先度の高いプログラム１はコア１がローカルＲＡＭ１をワークメモリして実行を継続し、優先度の低いプログラム４はコア１がローカルＲＡＭ２又は３をワークメモリして実行する。こうすることで、ローカルＲＡＭ１の一部の異常が検出された場合に、プログラム１～４の配置を最低化できる。
【００７３】
以上説明したように、本実施形態のマイコン１００は、コア１～３又はローカルＲＡＭ１～３に異常が検出された場合、優先度が高いプログラムを専用組み合わせで実行するようにコア１３とローカルＲＡＭ１５の組合せを変更するので、優先度の高いプログラム３１は正常時と同等の速度で実行が継続される。また、優先度の低いプログラムは実行速度は低下しても継続して実行される。
【００７４】
〔変形例〕
＜マルチプロセッサ＞
図８は、マルチプロセッサ型のマイコン１００のハードウェア構成図の一例を示す。図８において図３と同一部には同一の符号を付しその説明は省略する。このマイコン１００は図３と同様にＣＰＵ１～３を有しているが、各ＣＰＵ１～３は１つのプロセッサ２１に搭載されている。この場合、プロセッサ２１とＣＰＵ１～３はほぼ同じ意味になる場合がある。したがって各プロセッサ２１は、ＤＭＡＣ２２の調停を受けてマルチレイヤバス２８を介して他のプロセッサ２１と通信する。
【００７５】
しかしながら、異常監視部１～３の監視形態及び監視方法、停止部１～３の停止方法、並びに、割り当て制御部１～３のプログラム１～３の割り当て方法はマルチコアの場合と同じである。
【００７６】
したがって、本実施形態のコア１～３がプログラム１～３を実行する際のコア１～３とローカルＲＡＭ１～３の組合わせの変更は、マルチプロセッサ型のマイコン１００についても有効である。
【００７７】
＜マスター・スレーブ型＞
図９は、マルチコア型又はマルチプロセッサ型のマイコン１００において、コア１、２のフェールセーフ処理を１つのコア１３が集中的に制御する場合の機能ブロック図の一例を示す。この場合のハードウェア構成は図３又は図８のどちらでもよい。また、図１０は、コア３がコア１、２を監視する手順のシーケンス図の一例である。
【００７８】

図９に示すように、コア３が異常監視部４１、停止部４２及び割り当て制御部４３を有する。コア３は、異常監視等の専用のコアでもよいし、図３などのようにプログラム３１を実行してもよく、処理能力などにより設計できる。

【００７９】

図１０のステップＳ３００とＳ４００はいずれもコア３が図７と同様の手順を行うステップである。コア３は、ステップＳ３００ではコア１に対し、Ｓ４００ではコア２に対し、それぞれ図７の制御を行う。すなわち、異常監視部４１は、コア１、２に異常が検出される否かを監視し、また、ローカルＲＡＭ１，２に異常が検出されるか否かを監視する。コア１又は２に異常が検出された場合、停止部４２はコア１又はコア２を停止する。

【００８０】

例えば、コア１に異常が検出された場合、割り当て制御部４３は、コア２が優先度の高いプログラム１，２を、ローカルＲＡＭ２をワークエリアにして実行するように、コア１、２とローカルＲＡＭ２の組合せを変更する。優先度の低いプログラム１，２は、コア２がローカルＲＡＭ１をワークエリアにして実行することになる。コア２に異常が検出された場合はこの逆になる。

【００８１】

また、ローカルＲＡＭ１に異常が検出された場合、割り当て制御部４３は、コア２が優先度の高いプログラム１又は２をローカルＲＡＭ２をワークエリアにして実行するように、コア１、２とプログラム１，２の組合せを変更する。優先度の低いプログラム１又は２は、コア１がローカルＲＡＭ２をワークエリアにして実行することになる。ローカルＲＡＭ２に異常が検出された場合はこの逆になる。

【００８２】

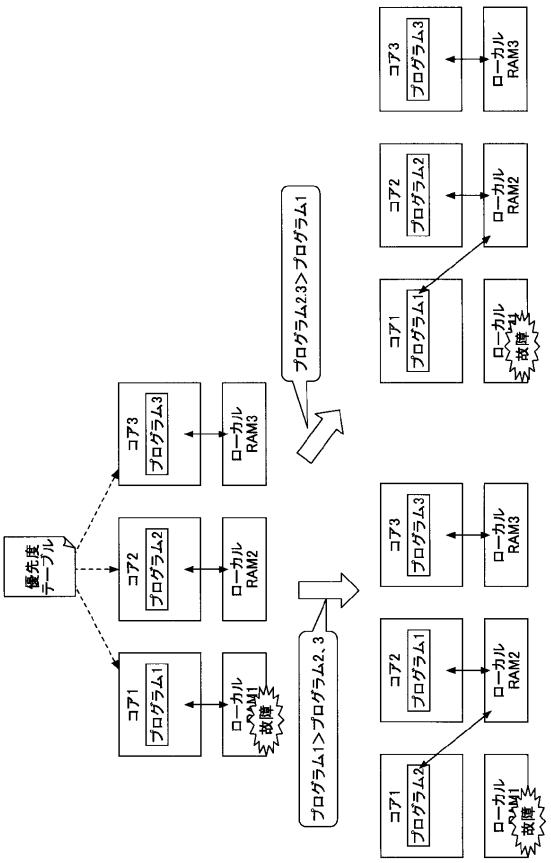
このように、本実施形態のプログラムとワークメモリの組合せは、各コアがそれぞれ制御することも、マスター・スレーブ型で１つのコアが制御することもできる。

【符号の説明】

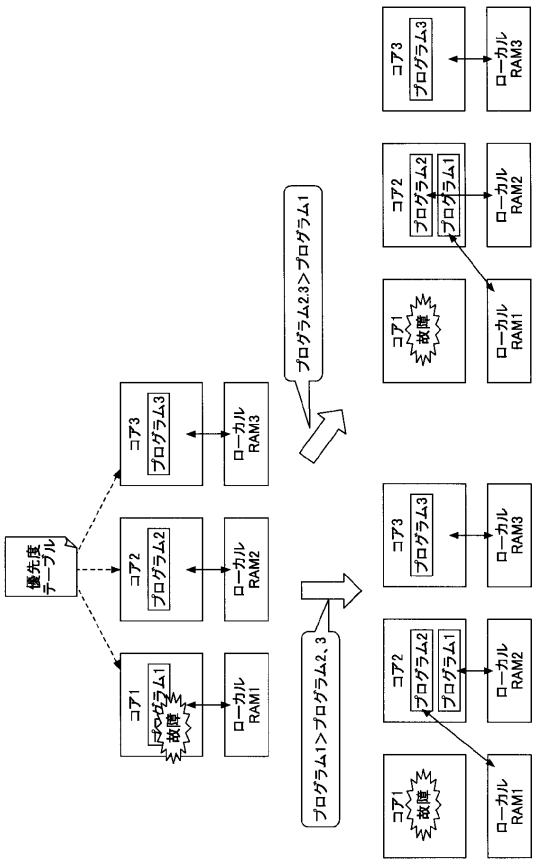
【００８３】

１２　コア間通信部
１３　コア
１４　リセット回路
１５　ローカルＲＡＭ
２１　プロセッサ
２２　ＤＭＡＣ
２３　ＳＤＲＡＭ
２４　ＲＯＭ
２５　Ｉ／Ｏブリッジ
２８　マルチレイヤバス
４１　異常監視部
４２　停止部
４３　割り当て制御部
１００　マイコン

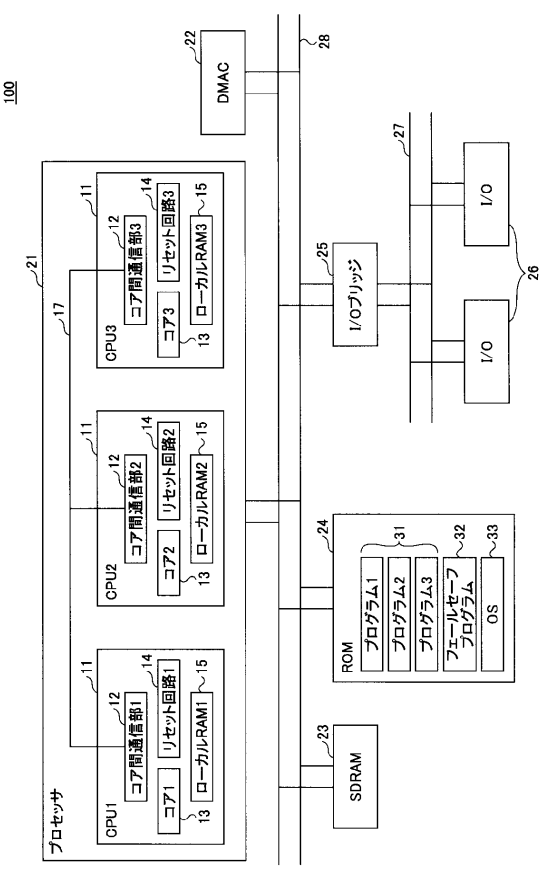
【図１】



【図２】



【図３】



【図４】

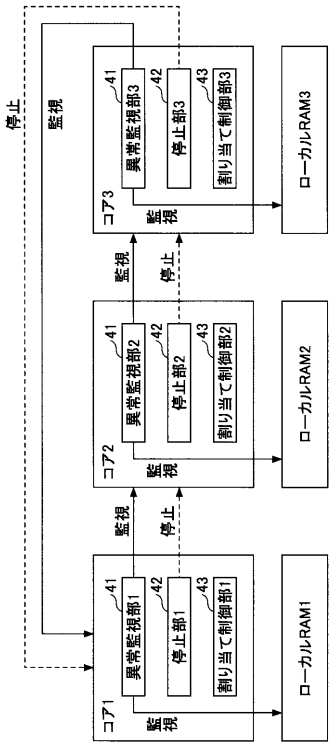
(a)

プログラム	優先度
プログラム1	高
プログラム2	中
プログラム3	低

(b)

コア	割り当てプログラム
コア1	プログラム1
コア2	プログラム2
コア3	プログラム3

【図５】



【図６】

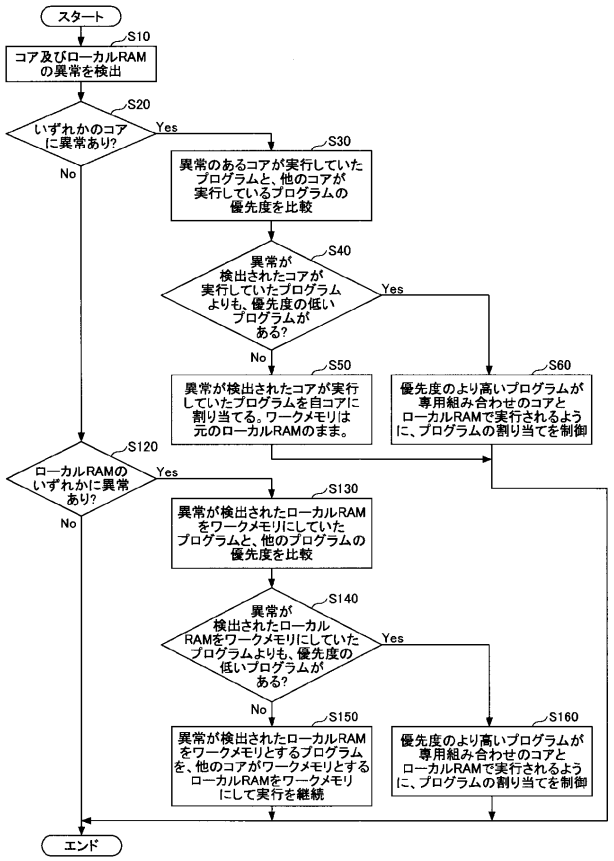
(a)

コア異常	ローカルRAM異常
コア1	—
—	—
—	—

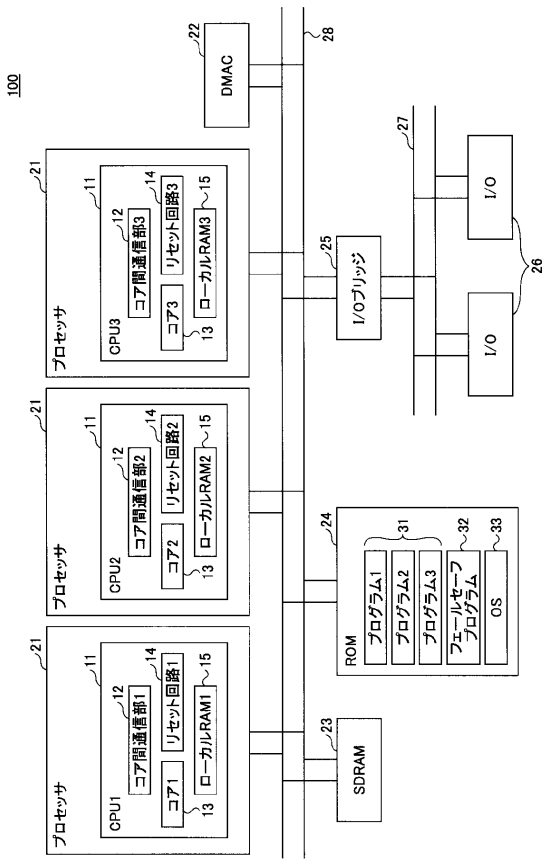
(b)

停止コア
コア1
—
—

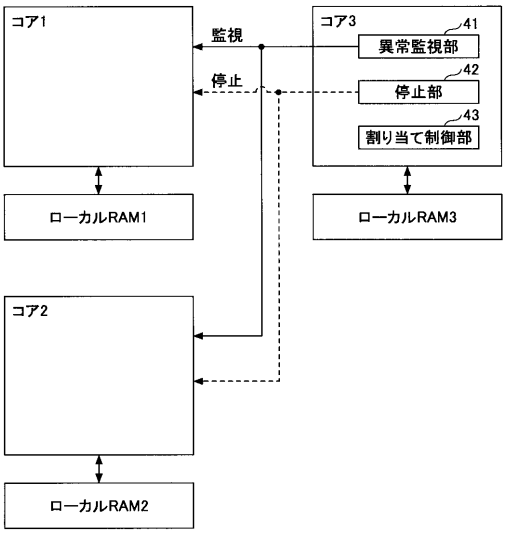
【図７】



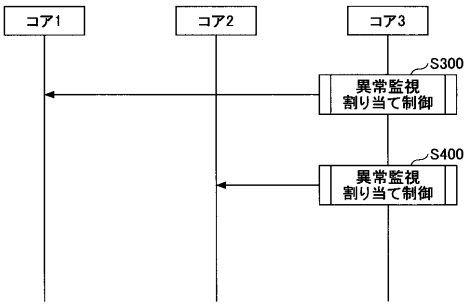
【図８】



【図９】


コア1
監視
停止
コア3
41
異常監視部
42
停止部
43
割り当て制御部
ローカルRAM1
ローカルRAM3
コア2
ローカルRAM2

【図１０】


コア1
コア2
コア3
S300
異常監視
割り当て制御
S400
異常監視
割り当て制御