(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2015-106226
(P2015-106226A)
(43)公開日　平成27年6月8日(2015.6.8)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 11/20	(2006.01)	G06F	11/20	310E	5B027
G06F 11/22	(2006.01)	G06F	11/22	360L	5B034
G06F 11/30	(2006.01)	G06F	11/30	F	5B042
G06F 11/14	(2006.01)	G06F	11/14	310K	5B048

審査請求 未請求　請求項の数 10　ＯＬ　（全 16 頁）

(21)出願番号　特願2013-246981 (P2013-246981)
(22)出願日　平成25年11月29日 (2013.11.29)

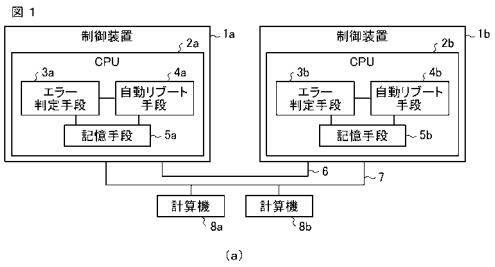
(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目７番３号
(74)代理人　100073759
弁理士　大岩　増雄
(74)代理人　100088199
弁理士　竹中　岑生
(74)代理人　100094916
弁理士　村上　啓吾
(74)代理人　100127672
弁理士　吉澤　憲治
(72)発明者　木本　寿郎
東京都千代田区丸の内二丁目７番３号　三菱電機株式会社内

最終頁に続く

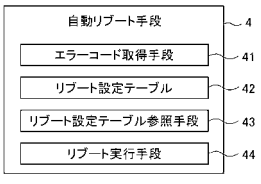
(54)【発明の名称】二重化システム

(57)【要約】
【課題】二重化システムにおける片系運転状態の時間を短縮し、システムの信頼性の向上を図る。
【解決手段】各制御装置１ａ、１ｂが、エラーコード取得手段４１、リブート設定テーブル４２、リブート設定テーブル参照手段４３、およびリブート実行手段４４を含む自動リブート手段４ａ、４ｂを備え、自装置のリブートを自動的に実行する。これにより、保守員による部品交換等の作業が必要な故障を除く故障を、自動的且つ迅速に復旧することが可能となり、片系運転状態の時間を短縮することができ、システムの信頼性が向上する。
【選択図】図１

図１



(a)



(b)

【特許請求の範囲】
【請求項１】
同一の機能を有する二台の制御装置が通信可能に接続され、一方は稼動系、他方は待機系として運用される冗長構成の二重化システムであって、前記制御装置は、
自装置に故障が発生した際に原因を特定しエラーコードを出力するエラー判定手段と、
各エラーコードに対応する故障内容と再起動を実行するか否かを定義したリブート設定テーブルを格納する記憶手段と、
前記エラー判定手段から取得したエラーコードを前記リブート設定テーブルに参照し自装置の再起動を実行するか否かを決定するリブート設定テーブル参照手段と、
前記リブート設定テーブル参照手段による参照結果に基づいて自装置の再起動を実行するリブート実行手段と、
自装置の運転状態を稼動系から待機系または待機系から稼動系に移行する稼動状態切り替え手段を備えたことを特徴とする二重化システム。
【請求項２】
前記リブート設定テーブルは、各エラーコードに対応する故障を、その内容に関連して軽度の故障と重度の故障に分類していることを特徴とする請求項１記載の二重化システム。
【請求項３】
前記制御装置と通信可能に接続され前記制御装置の故障を監視する監視装置を備え、前記制御装置は、自装置の運転状態および故障を前記監視装置に通知する通知手段を有することを特徴とする請求項１または請求項２に記載の二重化システム。
【請求項４】
前記通知手段は、自装置に発生した故障が前記リブート実行手段による再起動を実行できない故障であった場合に、その旨を前記監視装置に通知することを特徴とする請求項３記載の二重化システム。
【請求項５】
前記リブート設定テーブルは、各エラーコードに対応する故障を、その内容に関連して軽度の故障と重度の故障に分類しており、前記通知手段は、前記リブート実行手段による再起動の原因が重度の故障であった場合に、その旨を前記監視装置に通知することを特徴とする請求項３記載の二重化システム。
【請求項６】
稼動系として運用中の前記制御装置は、自装置に故障が発生した場合、前記稼動状態切り替え手段により自装置を稼動系から待機系に移行した後、前記リブート実行手段による再起動を実行し、該再起動の原因が軽度の故障であった場合には、待機系から稼動系に移行することを特徴とする請求項２記載の二重化システム。
【請求項７】
前記制御装置の中央処理装置は、複数のプロセッサコアを搭載しており、前記リブート実行手段は、再起動を実行する前に、故障時に主として使用されていた前記プロセッサコアを記憶しており、該プロセッサコアを除く前記プロセッサコアで再起動を実行することを特徴とする請求項１から請求項６のいずれか一項に記載の二重化システム。
【請求項８】
前記プロセッサコアは、同じ前記中央処理装置に搭載されている他の前記プロセッサコアの故障を検出するエラー検出手段を有し、主として使用されている前記プロセッサコアは、他の前記プロセッサコアの前記エラー検出手段により定期的に故障診断されていることを特徴とする請求項７記載の二重化システム。
【請求項９】
前記エラー検出手段は、主として使用されている前記プロセッサコアの故障を検出した場合、該プロセッサコアに対して再起動要求を通知することを特徴とする請求項８記載の二重化システム。
【請求項１０】

前記エラー検出手段は、主として使用されている前記プロセッサコアの故障を検出した場合、前記エラー判定手段と同様のエラーコードを出力し、前記リブート設定テーブル参照手段は、前記エラー検出手段から取得したエラーコードを前記リブート設定テーブルに参照し自装置の再起動を実行するか否かを決定することを特徴とする請求項８または請求項９に記載の二重化システム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、二重化システに関し、特に、稼動系と待機系を切り替える際に片系運転状態となる時間を短縮した二重化システムに関する。

【背景技術】

【０００２】

電気、ガス等の社会インフラに適用されるプラント監視制御システムとして、制御装置を二重化した冗長構成が採用されている。この方式では、同じ機能を有する制御装置を二台用意し、一台を稼動系（主系）、他を待機系（従系）とし、稼動系の制御装置に故障等の異常が発生した場合には、待機系の制御装置を稼動系に切り替える。これにより、システムの稼動中に稼動系の制御装置に故障が発生した場合でも、制御を継続させることができ、システムの信頼性が向上する。

【０００３】

このような二重化システムにおいて、稼動系の制御装置が故障等により停止し、保守員が故障した部品を交換する等の復旧作業を行い、待機系として再起動するまでの間は、片系運転状態となり冗長構成ではなくなる。すなわち、この片系運転時に稼動している一台の制御装置が停止すると、システム停止状態となり、システムの信頼性を著しく損なうことになる。特にインフラのプラント監視制御システムの停止は社会的な混乱を招くことになるため、片系運転となる時間をできるだけ短くする必要がある。

【０００４】

二重化システムにおける片系運転状態を短縮するための従来の対策としては、故障発生時を想定した保守部品の確保を行い、保守員が常駐して緊急事態に備え、迅速な復旧作業を行う体制をとっていた。しかし、復旧作業の改善による作業時間の短縮には限界があり、故障部位の特定や故障部品の交換に長時間を要することがあった。

【０００５】

また、特許文献１には、コンピュータシステムの外部に接続された障害監視装置を用いて、複数のコンピュータシステムで発生する障害を監視する方法が提示されている。この先行技術では、障害監視装置は、コンピュータシステムにおいて障害が発生した場合の復旧動作を定義する障害復旧情報を記憶しており、コンピュータシステムに障害が発生した場合、その障害に対応する障害復旧動作を行うようにコンピュータシステムに指示するものである。

【先行技術文献】

【特許文献】

【０００６】

【特許文献１】特開２００３－１１４８１１号公報

【発明の概要】

【発明が解決しようとする課題】

【０００７】

しかしながら、特許文献１の方法では、コンピュータシステムの他にシステム監視専用の障害監視装置を用意する必要があり、コストが高くなるという問題があった。また、障害監視装置に障害復旧情報を記憶させる必要があり、多大な労力を必要としていた。

【０００８】

本発明は、上記のような課題を解決するためになされたものであり、二重化システムにおける片系運転状態の時間を短縮し、システムの信頼性の向上を図ることを目的とする。

【課題を解決するための手段】
【０００９】
本発明に係る二重化システムは、同一の機能を有する二台の制御装置が通信可能に接続され、一方は稼動系、他方は待機系として運用される冗長構成の二重化システムであって、制御装置は、自装置に故障が発生した際に原因を特定しエラーコードを出力するエラー判定手段と、各エラーコードに対応する故障内容と再起動を実行するか否かを定義したリブート設定テーブルを格納する記憶手段と、エラー判定手段から取得したエラーコードをリブート設定テーブルに参照し自装置の再起動を実行するか否かを決定するリブート設定テーブル参照手段と、リブート設定テーブル参照手段による参照結果に基づいて自装置の再起動を実行するリブート実行手段と、自装置の運転状態を稼動系から待機系または待機系から稼動系に移行する稼動状態切り替え手段を備えたものである。
【発明の効果】
【００１０】
本発明に係る二重化システムによれば、各制御装置がリブート設定テーブルの参照結果に基づいて自装置の再起動を実行するようにしたので、保守員による復旧作業が必要な故障を除く故障を、自動的且つ迅速に復旧することが可能となり、保守員の負担が軽減されると共に、片系運転状態の時間を短縮することができ、システムの信頼性が向上する。
【図面の簡単な説明】
【００１１】
【図１】本発明の実施の形態１に係る二重化システムおよび自動リブート手段の構成を示す図である。
【図２】本発明の実施の形態１に係るリブート設定テーブルの一例を示す図である。
【図３】本発明の実施の形態１に係る自動リブート手段の処理の流れを示す図である。
【図４】本発明の実施の形態２に係る二重化システムの構成を示す図である。
【図５】本発明の実施の形態２に係る二重化システムにおける自動リブート手段の処理の流れを示す図である。
【図６】本発明の実施の形態３に係る二重化システムにおける自動リブート手段の処理の流れを示す図である。
【図７】本発明の実施の形態４に係る二重化システムにおける自動リブート手段の処理の流れを示す図である。
【図８】本発明の実施の形態４に係る二重化システムにおける自動リブート手段の処理の流れを示す図である
【図９】本発明の実施の形態５に係る二重化システムにおけるプロセッサコアのエラー検出手段の処理の流れを示す図である。
【図１０】本発明の実施の形態５に係る二重化システムにおけるプロセッサコアの自動リブート手段の処理の流れを示す図である。
【図１１】本発明の実施の形態６に係る二重化システムにおけるプロセッサコアの自動リブート手段の処理の流れを示す図である。
【図１２】本発明の実施の形態７に係る二重化システムにおいてリブート発生後に待機系となった制御装置のＣＰＵの処理の流れを示す図である。
【発明を実施するための形態】
【００１２】
実施の形態１.
以下に、本発明の実施の形態１に係る二重化システムについて、図面に基づいて説明する。図１（ａ）は、本実施の形態１に係る二重化システムの構成を示し、図１（ｂ）は、本実施の形態１に係る二重化システムにおける自動リブート手段の構成を示している。
【００１３】
本実施の形態１に係る二重化システムは、同一の機能を有する二台の制御装置１ａ、１ｂが互いに通信可能に接続され、一方は稼動系（主系）、他方は待機系（従系）として運用される冗長構成である。本システムをプラント監視制御システムに適用した場合、制御

装置１ａ、１ｂは、プラント監視制御装置として用いられる。
【００１４】
図１（ａ）に示すように、制御装置１ａ、１ｂは、ＣＰＵカードから構成される中央処理装置であるＣＰＵ２ａ、２ｂを備えている。これらのＣＰＵ２ａ、２ｂは、制御装置１ａ、１ｂによる主な制御を実行するためのプログラム（図示省略）を有すると共に、エラー判定手段３ａ、３ｂ、自動リブート手段４ａ、４ｂ、記憶手段５ａ、５ｂを有している。
【００１５】
エラー判定手段３ａ、３ｂは、自装置に故障が発生した際にその原因を特定し、該当するエラーコードを出力する。自動リブート手段４ａ、４ｂについては、後に詳細に説明する。記憶手段５ａ、５ｂは、例えば不揮発性のバックアップＳＲＡＭやＦＬＡＳＨである。なお、記憶手段５ａ、５ｂは、複数個備えられていても良い。エラー判定手段３ａ、３ｂから出力されたエラーコードは、記憶手段５ａ、５ｂに保存される。
【００１６】
ＣＰＵ２ａ、２ｂは、記憶手段５ａ、５ｂに保存されたデータを、トラッキングバス６を介して互いにやりとりする。トラッキングバス６には、イーサネット（Ｅｔｈｅｒｎｅｔ）（登録商標）等のＬＡＮや専用の制御線を用いることができる。
【００１７】
計算機８ａ、８ｂは、例えば下位計算機である。制御装置１ａ、１ｂと計算機８ａ、８ｂは、例えばＬＡＮ７を介して通信可能に接続されている。なお、通信手段としてＬＡＮ７以外のネットワーク、例えばＷＡＮ（Ｗｉｄｅ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）を用いても良い。なお、図１では、制御装置１ａ、１ｂと接続される装置として二台の計算機８ａ、８ｂを示したが、これに限定されるものではなく、計測器等であっても良い。
【００１８】
また、ＣＰＵ２ａ、２ｂは、自装置の運転状態を稼動系から待機系、または待機系から稼動系に移行する稼動状態切り替え手段（図示省略）を備えている。ただし、稼動状態切り替え手段は、ＣＰＵ２ａ、２ｂ内に配置されていなくても良い。また、制御装置１ａ、１ｂに運転状態の切り替えを指示する切り替え装置を、制御装置１ａ、１ｂの外部に備えた構成であっても良い。
【００１９】
図１に示す二重化システムの基本的な動作について簡単に説明する。なお、ここでは、制御装置１ａを稼動系で運用中であると仮定し、制御装置１ａのＣＰＵ２ａを稼動系ＣＰＵ２ａと呼ぶ。また、待機系の制御装置１ｂのＣＰＵ２ｂを待機系ＣＰＵ２ｂと呼ぶ。
【００２０】
計算機８ａ、８ｂは、稼動系の制御装置１ａに対し、プラントデータ等のデータを周期的に送信する。稼動系ＣＰＵ２ａは、計算機８ａ、８ｂから送信されたデータを取得し、記憶手段５ａに保存する。また、必要に応じ取得したデータに対し処理を実行する。待機系ＣＰＵ２ｂは、トラッキングバス６を介して稼動系ＣＰＵ２ａの内部データを取得し、内部データを等値化している。これにより、待機系ＣＰＵ２ｂが稼動系に移行した際に即座に動作を継続することができる。
【００２１】
制御装置１ａに故障が発生した場合、稼動系ＣＰＵ２ａは稼動状態切り替え手段により自装置の運転状態を稼動系から待機系に移行する。制御装置１ａが稼動系から待機系に移行したことは、制御装置１ａから送信される信号により制御装置１ｂに通知される。これを受けた制御装置１ｂのＣＰＵ２ｂは、稼動状態切り替え手段により自装置の運転状態を待機系から稼動系に移行する。
【００２２】
次に、自動リブート手段４ａ、４ｂ（総称して自動リブート手段４）について説明する。本システムにおける制御装置１ａ、１ｂは、自動リブート手段４ａ、４ｂを備えることにより、自装置のリブートすなわち再起動を自動的に実行し、自動復旧することが可能な

ものである。自動リブート手段４は、図１（ｂ）に示すように、エラーコード取得手段４１、リブート設定テーブル４２、リブート設定テーブル参照手段４３、およびリブート実行手段４４を含んで構成される。

【００２３】

エラーコード取得手段４１は、エラー判定手段３ａ、３ｂから出力されたエラーコードを取得し、該エラーコードをリブート設定テーブル参照手段４３に送る。リブート設定テーブル４２は、各エラーコードに対応する故障内容と再起動を実行するか否かを定義したテーブルである。なお、リブート設定テーブル４２は、機能的には自動リブート手段４に含まれるが、実際には記憶手段５ａ、５ｂに格納されている。

【００２４】

図２は、リブート設定テーブル４２の一例を示している。リブート設定テーブル４２の各欄は、エラーコード、故障内容、リブートの有無、故障の軽重、統計情報対象か否か、備考の項目で構成されている。リブートの有無の欄は、自動リブートを実行する場合は「１」が記載され、自動リブートを実行しない場合は「０」が記載されている。

【００２５】

次の欄では、各エラーコードに対応する故障を、その内容に関連して軽度の故障と重度の故障に分類している。図２に示す例では、重度の故障には「１」が記載され、軽度の故障には「０」が記載されている。また、次の欄では、統計情報対象の故障の場合には「１」が記載され、対象外の故障の場合には「０」が記載されている。統計情報対象の故障が発生した場合の処理については、実施の形態５で詳細に説明する。

【００２６】

図２に示す例では、エラーコード「０ｘ２００１」に対応する故障内容は、ゼロ割（除数を０として除算するエラー。その後の処理が続行不能に陥りプログラムの異常終了となる）等のフォールトエラーであり、リブート有りで重度故障である。エラーコード「０ｘ２００２」に対応する故障内容は、ＷＤＴエラー（コンピュータが正常に稼動しているかどうかを定期的に監視するウォッチドックタイマのエラー）であり、リブート有りで重度故障である。

【００２７】

また、エラーコード「０ｘ３００１」に対応する故障内容は、ＦＰＧＡ故障であり、リブート無しで重度故障である。ＦＰＧＡ故障は、ゲートアレイの故障であり、このようなハードウェア故障では部品交換が必要なため、自動リブートを実行することはできない。また、エラーコード「０ｘ４００１」に対応する故障内容は、ＬＡＮ通信リトライエラーであり、リブート有りで軽度故障であり、統計情報対象である。

【００２８】

リブート設定テーブル参照手段４３は、エラー判定手段３ａ、３ｂから取得したエラーコードをリブート設定テーブル４２に参照し、自装置のリブートを実行するか否かを決定する。図２に示す例では、エラーコード「０ｘ２００１」、「０ｘ２００２」の場合にはリブートを実行、「０ｘ４００１」の場合には、エラーカウンタを１つ上げ、規定回数以上となったらリブートを実行、「０ｘ３００１」の場合にはリブートを実行しない、と決定する。

【００２９】

リブート実行手段４４は、リブート設定テーブル参照手段４３による参照結果に基づいて、リブート有りの場合には、自装置のリブートを実行する。なお、リブート実行手段４４によるリブートは、稼動系として運用中の制御装置１ａに故障が発生した場合、稼動状態切り替え手段により自装置を稼動系から待機系に移行した後、実行される。すなわち、リブート実行時には、制御装置１ｂが稼動系として運用されており、片系運転状態となっている。

【００３０】

本実施の形態１に係る二重化システムにおける自動リブート手段４の処理の流れについて、図３のフローチャートを用いて説明する。なお、ここでは制御装置１ａを稼動系、制

御装置１ｂを待機系として運用している場合を例に挙げて説明するが、逆の場合も同様の処理が行われる。

【００３１】

図３のステップ１（Ｓ１）において、制御装置１ａに故障が発生した場合（ＹＥＳ）、ステップ２（Ｓ２）において、エラーコード取得手段４１はエラー判定手段３ａが出力したエラーコードを取得する。Ｓ１で故障が発生していない場合（ＮＯ）は、処理は行われない。

【００３２】

続いて、ステップ３（Ｓ３）において、制御装置１ａは、稼動状態切り替え手段により自装置を稼動系から待機系に移行する。この通知を受けた制御装置１ｂは、自装置を待機系から稼動系に移行し、これまでの制御装置１ａの動作を継続する。次に、ステップ４（Ｓ４）において、リブート設定テーブル参照手段４３は、Ｓ２で取得したエラーコードをリブート設定テーブル４２に参照し、ステップ５（Ｓ５）において自装置のリブートを実行するか否かを決定する。

【００３３】

Ｓ５において、参照結果がリブート有りであった場合（ＹＥＳ）、ステップ６（Ｓ６）に進み、リブート実行手段４４はリブートを実行する。その後、初期化処理を経て、制御装置１ａは待機系として運用される。Ｓ５において、参照結果がリブート無しであった場合（ＮＯ）、処理を終了する。

【００３４】

本実施の形態１に係る二重化システムによれば、各制御装置１ａ、１ｂが自動リブート手段４ａ、４ｂを備え、自装置のリブートを自動的に実行するようにしたので、保守員による部品交換等の作業が必要な故障を除く故障を、自動的且つ迅速に復旧することが可能である。これにより、保守員の負担が軽減されると共に、片系運転状態の時間を短縮することができ、システムの信頼性が向上する。

【００３５】

実施の形態２.

図４は、本発明の実施の形態２に係る二重化システムの構成を示している。なお、図４において、図１と同一または相当部分には同一符号を付している。本実施の形態２に係る二重化システムは、制御装置１ａ、１ｂと通信可能に接続され、制御装置１ａ、１ｂの故障を監視する監視装置９を備えている。さらに、各制御装置１ａ、１ｂは、自装置の運転状態および故障を監視装置９に通知する通知手段（図示省略）を備えている。それ以外の構成については、上記実施の形態１（図１）と同様であるので説明を省略する。

【００３６】

各制御装置１ａ、１ｂと監視装置９は、ＬＡＮ７を介して接続されている。各制御装置１ａ、１ｂの通知手段は、自装置に発生した故障が、リブート実行手段４４によるリブートを実行できない故障であった場合に、その旨を監視装置９に通知する。通知を受けた監視装置９は、その表示手段または警報手段等により故障の発生を保守員に報知し、保守員は必要な復旧作業を行う。

【００３７】

本実施の形態２に係る二重化システムにおける自動リブート手段４の処理の流れについて、図５のフローチャートを用いて説明する。ただし、図５において、Ｓ１～Ｓ６は、上記実施の形態１で説明した図３のフローチャートと同じ処理であるので、説明を省略する。

【００３８】

図５のＳ５において、制御装置１ａのリブート設定テーブル参照手段４３は、Ｓ２で取得したエラーコードをリブート設定テーブル４２に参照し、リブート無しであった場合（ＮＯ）、上記実施の形態１では処理を終了したが、本実施の形態２では、ステップ５１（Ｓ５１）に進み、制御装置１ａの通知手段は、自動復旧不可であることを監視装置９に通知する。その後、自装置を停止し処理を終了する。

【００３９】
本実施の形態２に係る二重化システムによれば、上記実施の形態１と同様の効果に加え、自動リブートを実行できない故障が発生した場合に、自動復旧不可であることを監視装置９に通知し、保守員が迅速に部品交換等の復旧作業を行えるようにしたので、片系運転状態の時間が短縮され、さらにシステムの信頼性が向上する。
【００４０】
実施の形態３．
本発明の実施の形態３に係る二重化システムの構成は、上記実施の形態２と同様であるので図４を流用して説明する。本実施の形態３に係る二重化システムは、制御装置１ａ、１ｂと通信可能に接続され、制御装置１ａ、１ｂの故障を監視する監視装置９を備えている。また、各制御装置１ａ、１ｂは、自装置の運転状態および故障を監視装置９に通知する通知手段（図示省略）を備えている。
【００４１】
各制御装置１ａ、１ｂの通知手段は、上記実施の形態２と同様に、自装置に発生した故障がリブート実行手段４４によるリブートを実行できない故障であった場合に、その旨を監視装置９に通知する。さらに、本実施の形態３では、通知手段は、リブート実行手段４４によるリブートの原因が重度の故障であった場合に、その旨を監視装置９に通知するようにしている。発生した故障が重度の故障であるか否かは、リブート設定テーブル４２に定義されている。
【００４２】
本実施の形態３に係る二重化システムにおける自動リブート手段４の処理の流れについて、図６のフローチャートを用いて説明する。ただし、図６において、Ｓ１～Ｓ６、およびＳ５１は、上記実施の形態２で説明した図５のフローチャートと同じ処理であるので、説明を省略する。
【００４３】
図６のＳ５において、制御装置１ａのリブート設定テーブル参照手段４３は、Ｓ２で取得したエラーコードをリブート設定テーブル４２に参照し、リブート有りであった場合（ＹＥＳ）、ステップ５２（Ｓ５２）に進む。Ｓ５２において、リブート設定テーブル参照手段４３は、現在発生している故障が軽度の故障であるか否かを、リブート設定テーブル４２を参照して確認する。
【００４４】
Ｓ５２において、現在発生している故障が軽度の故障であった場合（ＹＥＳ）、Ｓ６に進み、リブート実行手段４４はリブートを実行する。Ｓ５２で重度の故障であった場合（ＮＯ）、ステップ５３（Ｓ５３）に進み、制御装置１ａの通知手段は、現在発生している故障が重度の故障であることを監視装置９に通知する。その後、Ｓ６に進みリブートを実行する。
【００４５】
本実施の形態３に係る二重化システムによれば、上記実施の形態１および実施の形態２と同様の効果が得られる。さらに、軽度の故障の場合には、監視装置９に通知せずに自動でリブートが実行されるので、あたかも故障が発生していないように運転が継続され、保守員の負担がさらに軽減される。また、重度の故障の場合には、監視装置９に通知した後リブートが実行されるため、保守員が重度の故障の発生を把握することができ、システムの信頼性がさらに向上する。
【００４６】
実施の形態４．
本発明の実施の形態４に係る二重化システムの主な構成は、上記実施の形態２と同様であるので図４を流用して説明する。本実施の形態４では、制御装置１ａ、１ｂのＣＰＵ２ａ、２ｂは、それぞれ二つのプロセッサコア（コア１、コア２；図示省略）を搭載している。
【００４７】

本実施の形態４において、リブート実行手段４４は、リブートを実行する前に、故障時に主として使用されていたプロセッサコアを記憶手段５ａ（または５ｂ）に記憶しており、該プロセッサコアを除くプロセッサコアで再起動を実行する。その他の構成および動作については、上記実施の形態１～実施の形態３と同様であるので説明を省略する。

【００４８】

本実施の形態４に係る二重化システムにおける自動リブート手段４の処理の流れについて、図７および図８のフローチャートを用いて説明する。ただし、図７において、Ｓ１～Ｓ６、およびＳ５１～Ｓ５３は、上記実施の形態３で説明した図６のフローチャートと同じ処理であるので、説明を省略する。

【００４９】

図７のフローチャートのＳ５２において、制御装置１ａのリブート設定テーブル参照手段４３は、現在発生している故障が軽度の故障であるか否かをリブート設定テーブル４２で確認し、軽度の故障であった場合（ＹＥＳ）、ステップ５４（Ｓ５４）に進む。また、Ｓ５２で重度の故障であった場合（ＮＯ）、Ｓ５３で現在発生している故障が重度の故障であることを監視装置９に通知した後、Ｓ５４に進む。

【００５０】

Ｓ５４では、制御装置１ａのＣＰＵ２ａにおいて、現在、すなわち故障時に主として使用されていたプロセッサコアを記憶手段５ａに記憶する。続いてステップ５５（Ｓ５５）において、自動リブートの履歴（エラーコード、実行時刻等）を記憶手段５ａに記憶した後、Ｓ６でリブートを実行する。

【００５１】

図８は、Ｓ６のリブート実行後の処理の流れを示している。図８のステップ６１（Ｓ６１）において、制御装置１ａにリブートが発生すると、ステップ６２（Ｓ６２）において、該リブートが自動リブート（自動リブート手段４によるリブート）であるか否かを、記憶手段５ａのリブート履歴を参照して確認する。自動リブート手段４によるリブートの場合には、図７のＳ５５で記憶手段５ａに記憶されている。

【００５２】

なお、リブートには、自動リブート手段４によるリブートの他に、保守員によるリブートもある。Ｓ６２において、自動リブート手段４によるリブートではない場合（ＮＯ）、ステップ６６（Ｓ６６）に進み、通常の初期化処理を実行する。

【００５３】

また、Ｓ６２において、自動リブート手段４によるリブートであった場合（ＹＥＳ）、ステップ６３（Ｓ６３）に進み、リブート実行前に主として動作していたプロセッサコアがどちらであったかを確認する。Ｓ６３において、主として動作していたのがコア１であった場合（ＹＥＳ）、ステップ６４（Ｓ６４）に進み、コア２で起動する。Ｓ６３において、主として動作していたのがコア２であった場合（ＮＯ）、ステップ６５（Ｓ６５）に進み、コア１で起動する。その後、Ｓ６６に進み、通常の初期化処理を実行する。

【００５４】

本実施の形態４に係る二重化システムによれば、上記実施の形態１～実施の形態３と同様の効果が得られる。さらに、自動リブート手段４のリブート実行手段４４によるリブートを実行する前に、主として使用されているプロセッサコアを記憶しておき、リブート後は別のプロセッサコアで起動するようにしたので、片方のプロセッサコアが故障した場合にも、待機系としての動作を継続して実行することができる。これにより、片系運転状態の時間を短縮することができ、システムの信頼性がさらに向上する。

【００５５】

実施の形態５．

本発明の実施の形態５に係る二重化システムの主な構成は、上記実施の形態２と同様であるので図４を流用して説明する。本実施の形態５では、上記実施の形態４と同様に、制御装置１ａ、１ｂのＣＰＵ２ａ、２ｂは、それぞれ二つのプロセッサコア（コア１、コア２）を搭載している。また、プロセッサコアは、同じＣＰＵ２ａ（または２ｂ）に搭載さ

れた他のプロセッサコアの故障を検出するエラー検出手段（図示省略）を有している。その他の構成および動作については、上記実施の形態４と同様であるので説明を省略する。

【００５６】

例えば稼動系として運用されている制御装置１ａのＣＰＵ２ａにおいて、主として使用されているコア１は、同じＣＰＵ２ａに搭載されたコア２のエラー検出手段により定期的に故障診断されている。コア２のエラー検出手段は、コア１の故障を検出した場合、その故障内容に応じたエラーコードを出力し、コア１に対してリブート要求を通知する。なお、エラー検出手段により出力されるエラーコードは、エラー判定手段３ａと同様のものである。

【００５７】

すなわち、本実施の形態５では、制御装置１ａにおいて、リブート実行手段４４がリブートを実行するパターンとして、以下の二つがある。一つは、制御装置１ａに故障が発生し、エラー判定手段３ａによりエラーコードが出力され、このエラーコードをリブート設定テーブルに参照した結果「リブート有」であった場合である。もう一つは、制御装置１ａのＣＰＵ２ａにおいて主として動作しているコア１の故障をコア２のエラー検出手段が検出し、その故障内容によりコア１に対しリブート要求を通知した場合である。

【００５８】

稼動系として運用されている制御装置１ａのＣＰＵ２ａにおいて、コア１が主として使用されている場合の、コア２のエラー検出手段の処理の流れについて、図９のフローチャートを用いて説明する。なお、図９では、統計情報対象のエラーが発生した場合を例に挙げて説明する。

【００５９】

統計情報対象のエラーとは、例えば図２に示すリブート設定テーブルのエラーコード「０ｘ４００１」に対応するＬＡＮ通信リトライエラーのような、比較的軽度の故障である。頻繁に発生し易い軽度な故障は、１回の発生でリブートを実行せず、発生回数が予め設定された規定回数（例えば１００回）以上となった時にリブートを実行する。

【００６０】

図９のステップ７（Ｓ７）において、統計情報対象のエラーが発生すると、コア２のエラー検出手段は、ステップ７１（Ｓ７１）において、エラーカウンタのカウントを１つ上げる。続いてステップ７２（Ｓ７２）において、エラーカウンタのカウントが規定回数以上であるか否かを判定する。規定回数以上ではない場合（ＮＯ）は、リブート不要であるため処理を終了する。

【００６１】

また、Ｓ７２において、規定回数以上の場合（ＹＥＳ）は、ステップ７３（Ｓ７３）に進み、カウンタをクリアする。続いて、ステップ７４（Ｓ７４）において、コア１に対しリブート要求を通知する。

【００６２】

次に、コア２からリブート要求の通知を受けたコア１の自動リブート手段４の処理の流れについて、図１０のフローチャートを用いて説明する。図１０のステップ８（Ｓ８）において、コア１は、コア２からの通知待ち（無限待ち）状態であり、ステップ８１（Ｓ８１）においてコア２からリブート要求の通知を受けた場合（ＹＥＳ）、ステップ８２（Ｓ８２）に進む。

【００６３】

Ｓ８２では、自装置の運転状態を稼動系から待機系に移行し、ステップ８３（Ｓ８３）において、リブート実行手段４４によりリブートを実行する。また、Ｓ８１でコア２からリブート要求の通知を受けていない場合（ＮＯ）は、処理を終了し、再度Ｓ８に戻りコア２からの通知待ち状態となる。

【００６４】

本実施の形態５に係る二重化システムによれば、上記実施の形態１～実施の形態４と同様の効果が得られる。さらに、同じＣＰＵ２ａ（または２ｂ）内に二つのプロセッサコア

を搭載し、主として使用されているプロセッサコア（例えばコア１）を、他のプロセッサコア（例えばコア２）のエラー検出手段で定期的に診断しているので、制御装置１ａ（または１ｂ）が重度の故障になる前に自動的にリブートを実行することができ、システムの信頼性がさらに向上する。

【００６５】

実施の形態６．

本発明の実施の形態６に係る二重化システムの主な構成は、上記実施の形態２と同様であるので図４を流用して説明する。本実施の形態６では、上記実施の形態５と同様に、例えば稼動系として運用されている制御装置１ａのＣＰＵ２ａにおいて、主として使用されているコア１を、同じＣＰＵ２ａに搭載されているコア２のエラー検出手段が定期的に故障診断し、故障を検出した場合、コア１に対しリブート要求を通知する。

【００６６】

上記実施の形態５では、コア２からリブート要求の通知を受けたコア１は、図１０に示すように、稼動系から待機系に移行した後、すぐにリブートを実行しているが、本実施の形態６では、コア２からリブート要求の通知を受けたコア１は、上記実施の形態４と同様の処理（図７に示すフローチャート）を経てリブートを実行するようにしている。

【００６７】

図１１は、本実施の形態６に係る二重化システムにおいて、コア２からリブート要求の通知を受けたコア１の自動リブート手段４の処理の流れを示すフローチャートである。なお、図１１において、Ｓ３～Ｓ６、およびＳ５１～Ｓ５５は、上記実施の形態４で説明した図７のフローチャートと同じ処理であるので説明を省略する。

【００６８】

図１１のステップ９（Ｓ９）において、コア１は、コア２からの通知待ち（有限待ち）状態であり、ステップ９１（Ｓ９１）において、コア２からリブート要求の通知を受けた場合（ＹＥＳ）、ステップ９２（Ｓ９２）に進む。また、Ｓ９１においてリブート要求の通知を受けていない場合は、処理を終了する。Ｓ９２において、コア１のエラーコード取得手段４１は、コア２のエラー検出手段が出力したエラーコードを取得し、Ｓ３に進む。

【００６９】

また、本実施の形態６では、リブートを実行した後、上記実施の形態４で説明した図８のフローチャートと同様の処理を行う。すなわち、リブート実行手段４４によるリブートを実行する前に、主として使用されているコアを記憶し、リブート後は別のコアで起動するようにしている。

【００７０】

本実施の形態６に係る二重化システムによれば、上記実施の形態１～実施の形態５と同様の効果が得られる。さらに、コア２からのリブート要求時においてもリブート設定テーブル４２を参照するようにしたので、リブートの有無をリブート設定テーブル４２により決定することができ、汎用性が向上する。

【００７１】

実施の形態７．

本発明の実施の形態７に係る二重化システムの構成は、上記実施の形態２と同様であるので図４を流用して説明する。本実施の形態７では、稼動系として運用されている制御装置１ａ（または１ｂ）は、自装置に故障が発生した場合、稼動状態切り替え手段により自装置を稼動系から待機系に移行した後、リブート実行手段４４によるリブートを実行し、該リブートの原因が軽度の故障であった場合には、待機系から稼動系に再度移行するようにしたものである。

【００７２】

図１２は、本実施の形態７に係る二重化システムにおいて、リブート発生後、待機系となった制御装置１ａのＣＰＵ２ａの処理の流れを示すフローチャートである。Ｓ６１において、制御装置１ａにリブートが発生すると、Ｓ６２において、該リブートが自動リブート（リブート実行手段４４によるリブート）であるか否かを、記憶手段５ａを参照して確

認する。

【００７３】

Ｓ６２において、リブート実行手段４４によるリブートではない場合（ＮＯ）、処理を終了する。Ｓ６２においてリブート実行手段４４によるリブートであった場合（ＹＥＳ）、ステップ６６（Ｓ６６）において通常初期化処理を行う。続いてステップ６７（Ｓ６７）において、Ｓ６１で発生したリブートが軽度の故障によるものか否かを、記憶手段５ａを参照して確認する。リブート実行手段４４によるリブートの場合、リブートの履歴は記憶手段５ａに記憶されている。

【００７４】

Ｓ６７において、Ｓ６１で発生したリブートが重度の故障によるものであった場合（ＮＯ）、処理を終了する。また、Ｓ６７において、Ｓ６１で発生したリブートが軽度の故障によるものであった場合（ＹＥＳ）、待機系から稼動系に移行する。ただし、この場合には、後に再度リブートする必要がある。

【００７５】

本実施の形態７に係る二重化システムによれば、上記実施の形態１～実施の形態６と同様の効果が得られる。さらに、自動リブート手段４によるリブートが軽度の故障による場合には、待機系に移行した後、再度稼動系に移行して元の状態に戻るため、あたかも故障が発生していないように運転が継続され、システムの信頼性がさらに向上する。

【００７６】

なお、本実施の形態７による処理の流れは、上記実施の形態４～実施の形態６に係る二重化システム、すなわちＣＰＵ２ａ、２ｂに複数のプロセッサコアを搭載した場合にも、適用することができる。本発明は、その発明の範囲内において、各実施の形態を自由に組み合わせたり、各実施の形態を適宜、変形、省略したりすることが可能である。

【産業上の利用可能性】

【００７７】

本発明は、制御装置を二重化したプラント監視制御システムとして利用することができる。

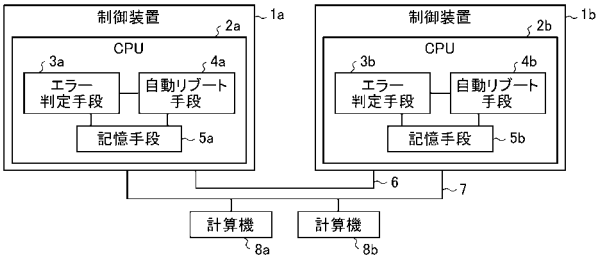
【符号の説明】

【００７８】

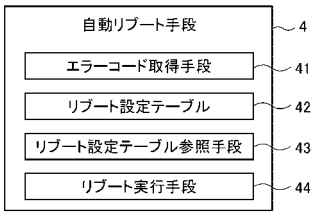
１ａ、１ｂ　制御装置、２ａ、２ｂ　ＣＰＵ、３ａ、３ｂ　エラー判定手段、４、４ａ、４ｂ　自動リブート手段、５ａ、５ｂ　記憶手段、６　トラッキングバス、７　ＬＡＮ、８ａ、８ｂ　計算機、９　監視装置、４１　エラーコード取得手段、４２　リブート設定テーブル、４３　リブート設定テーブル参照手段、４４　リブート実行手段。

【図１】

図１



(a)



(b)

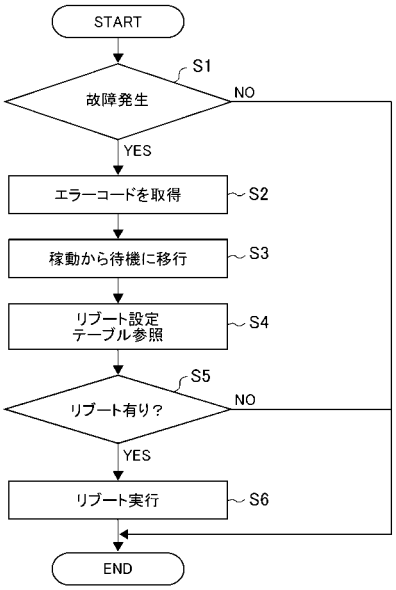
【図２】

図２

エラーコード	故障内容	リブート有=1 無=0	重度故障=1 軽度故障=0	統計情報対象	備考
0x2001	フォールト	1	1	0	0割などのエラー
0x2002	WDTエラー	1	1	0	
0x3001	FPGA故障	0	1	0	ゲートアレイ故障
0x4001	LAN通信リトライエラー	1	0	1	

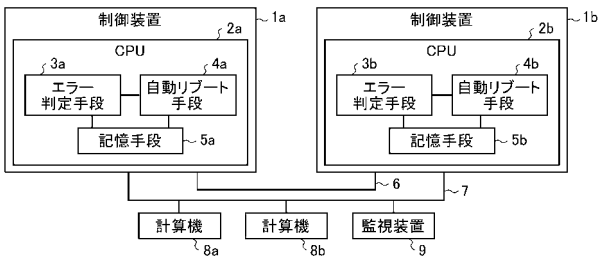
【図３】

図３



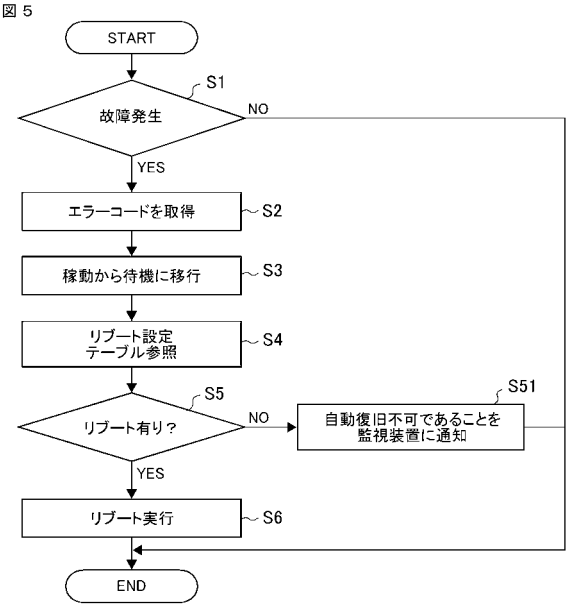
【図４】

図４



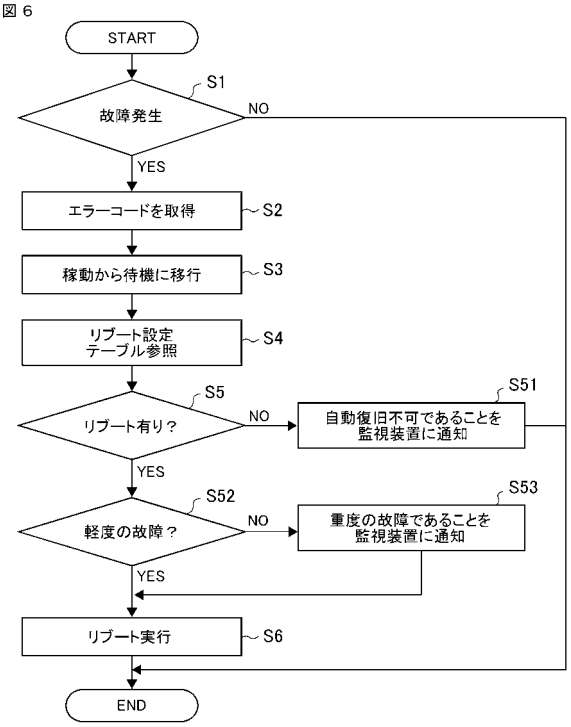
【図５】

図５



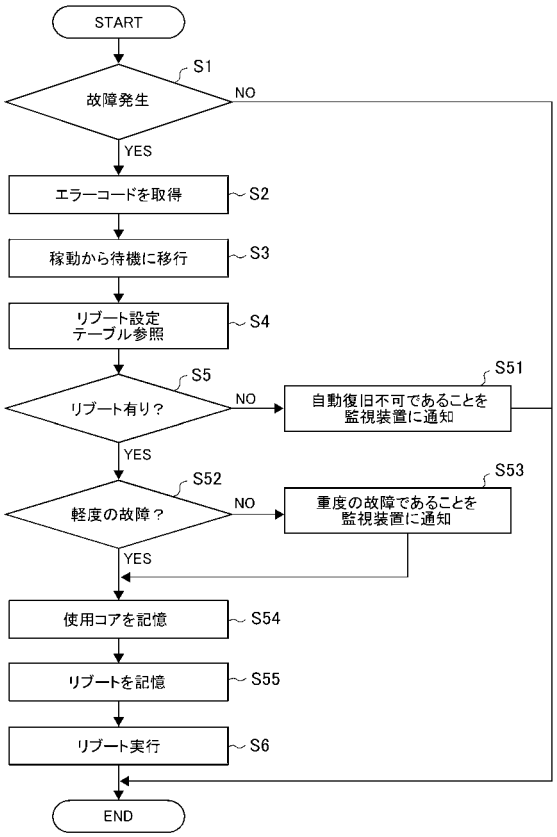
【図６】

図６



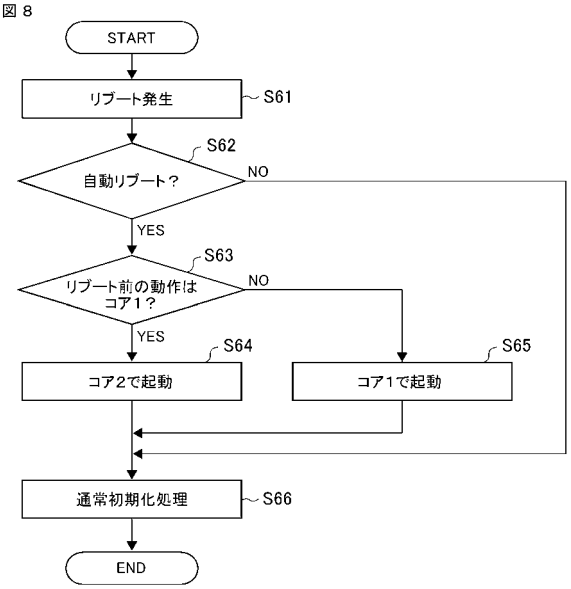
【図７】

図７



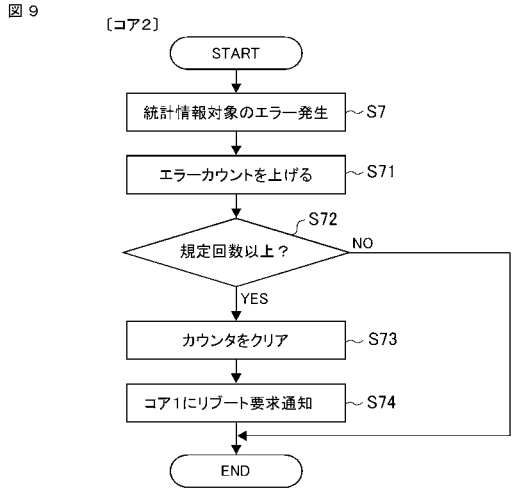
【図８】

図８



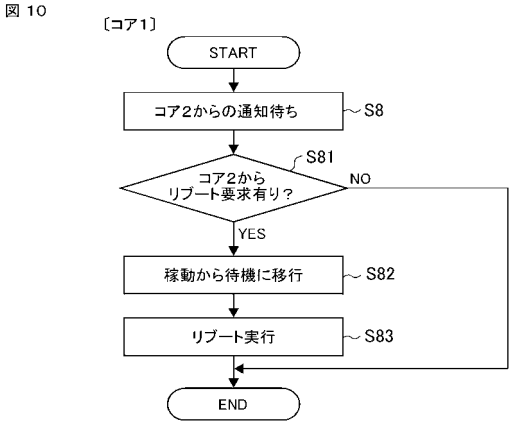
【図９】

図９



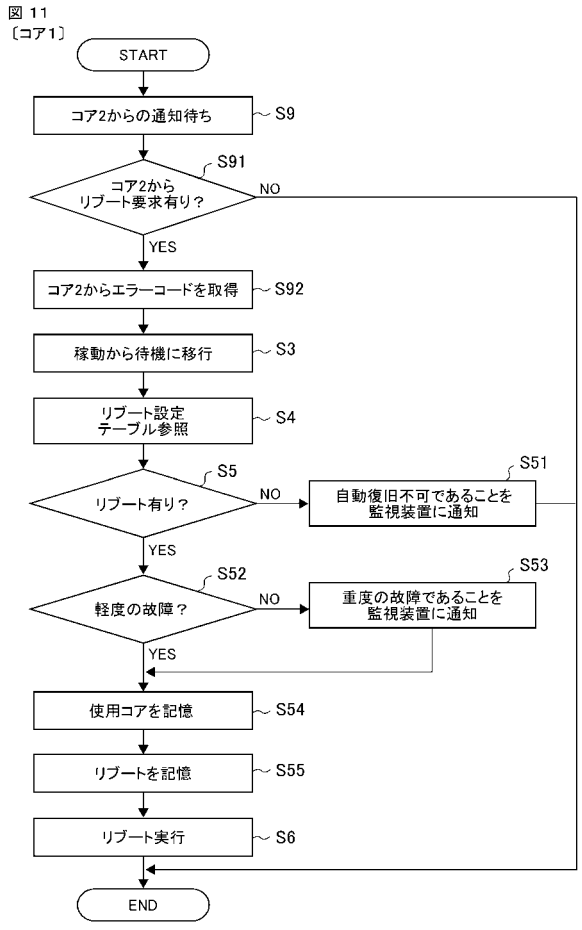
【図１０】

図１０



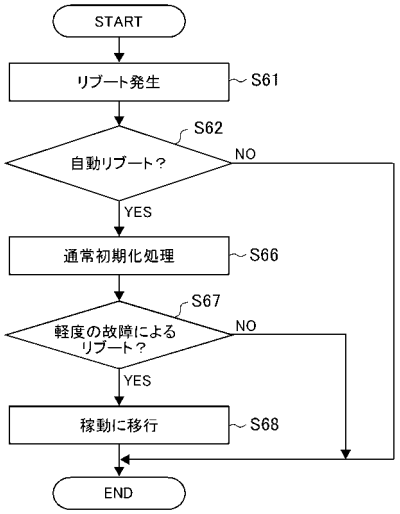
【図１１】

図１１



【図１２】

図１２



フロントページの続き

(72)発明者　梶田　智之
東京都千代田区丸の内二丁目７番３号　三菱電機株式会社内

Ｆターム(参考) 5B027 AA05 CC03
5B034 BB02 BB16 CC01 DD05
5B042 GA11 GA12 GB06 JJ04 JJ08 KK02 KK04 KK11 KK20 MC16
5B048 AA01 CC14