(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2010-3081
(P2010-3081A)
(43)公開日　平成22年1月7日(2010.1.7)

(51)Int.Cl.		F I		テーマコード（参考）
G06F 11/20	(2006.01)	G06F 11/20	310A	5B034

審査請求 未請求　請求項の数 5　ＯＬ　（全 11 頁）

(21)出願番号　特願2008-160877 (P2008-160877)
(22)出願日　平成20年6月19日 (2008.6.19)

(71)出願人　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(71)出願人　000153443
株式会社日立情報制御ソリューションズ
茨城県日立市大みか町５丁目２番１号
(74)代理人　110000442
特許業務法人　武和国際特許事務所
(72)発明者　清水　俊樹
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内

最終頁に続く

(54)【発明の名称】演算処理装置多重化制御システム

(57)【要約】
【課題】従来は、主系のＣＰＵのメモリデータ信号をスヌープ（Snoop）し、当該メモリデータ信号をパラレルバスを用いた一致化バスを介し従系のＣＰＵが取得していたので、ノイズに弱いパラレルバスを用いていることから長距離伝送には不向きである点、スヌープ（Snoop）するデータが主系にメモリされるデータであるために従系が主系と同時に処理できずに遅れることによるシステムの応答速度が犠牲になってしまう点、さらに、プログラムの更新に際してはプラントを止めることなく行うための機能を別途備える必要があり、機能が複雑になる点等の問題点を有していた。
【解決手段】本発明では、2重化制御システムにおいて、待機系である従系のＣＰＵが実行系である主系のＣＰＵからスヌープ（Snoop）するデータを、主系のＣＰＵが取得する制御対象からの制御情報とすること、さらに、有利的には従系の制御周期の位相を主系の制御周期の位相より進めること。
【選択図】図１

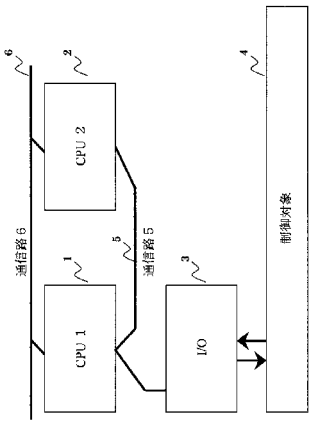

図1

【特許請求の範囲】
【請求項１】
複数の演算処理装置、制御対象に接続された入出力機器、前記演算処理装置と入出力機器を接続する通信路、前記演算処理装置同士を接続する通信路等から構成され、前記複数の演算処理装置は実行系の主系と待機系の従系に切替設定可能な演算処理装置２重化制御システムにおいて、主系演算処理装置が取得する制御対象からの制御情報を従系演算処理装置が同時にスヌープ（Snoop）する演算処理装置２重化制御システム。
【請求項２】
請求項１における従系演算処理装置は、その制御周期の位相が主系演算装置の制御周期の位相より相対的に進んでいる演算処理装置２重化制御システム。
【請求項３】
請求項２における相対的に進んだ位相は、主系演算処理装置からの同期要求に応じて従系演算処理装置がトリガをかけて制御周期を開始し、当該同期要求の応答として従系演算処理装置から主系演算処理装置に対して出力する同期要求確認応答に応じて、主系演算処理装置がトリガをかけて制御周期を開始することにより設定された演算処理装置２重化制御システム。
【請求項４】
請求項１における主系と従系の切替は、主系演算処理装置が行う自己診断において異常が検出された場合、または、主系演算処理装置から定期的に送信されるべき出力信号が停止したことを従系演算処理装置が確認したときに行う演算処理装置２重化制御システム。
【請求項５】
請求項１における主系と従系の切替が、プログラム更新に際しては、予めプログラム更新された従系演算処理装置の動作を、主系演算処理装置からスヌープ（Snoop）した制御対象からの制御情報を用いて確認したときに行われる、演算処理装置２重化制御システム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、プラント制御等に用いる演算処理装置（以降ＣＰＵと略する）の多重化システムに関する技術である。
【背景技術】
【０００２】
従来、制御用プラント等に用いる制御用ＣＰＵを2重あるいは多重構成にして、主系のＣＰＵに障害が発生した場合にも停止することなく制御を行ったり、制御用プログラムの更新を行う際にはプラントを停止することなく待機系のＣＰＵにプログラム更新を行うことによりシステムの可用性を高めることが行われており、その場合には、前記更新時、障害発生時に速やかに切り替わることが必要となる。
【０００３】
当該速やかな切り替えのためには、複数台のＣＰＵは制御対象からの制御情報を同一態様で取得している必要があり、当該複数のＣＰＵが同一態様で取得するための手段としては、各ＣＰＵが独立して制御対象からの制御情報を取得することや、２重化を構成するＣＰＵの1方のＣＰＵから他方のＣＰＵがメモリデータをスヌープ（Snoop）することにより取得すること（特許文献１、特許文献２参照）等が従来より知られていた。
【特許文献１】特開平９－３０５４２４号公報
【特許文献２】特開平９－２４５００８号公報
【発明の開示】
【発明が解決しようとする課題】
【０００４】
しかしながら、前記従来例においては、主系のＣＰＵとそのメモリを結ぶラインからメモリデータ信号をスヌープ（Snoop）し、当該メモリデータ信号を、パラレルバスを用い

た一致化バスを介して従系のＣＰＵが取得するもので、ノイズに弱いパラレルバスを用いていることから長距離伝送には不向きである点、スヌープ（Snoop）するデータが主系にメモリされるデータであるために従系が主系と同時に処理できずに遅れることによるシステムの応答速度が犠牲になってしまう点、さらに、プログラムの更新に際してはプラントを止めることなく行うための機能を別途備える必要があり、機能が複雑になる点等の問題点を有していた。

【０００５】

本発明は、上記の課題を解決するためになされたものであり、主系と従系間が長距離であってもノイズに強く、しかも、システムの応答速度を犠牲にすることなく、プログラム更新時、障害発生時にスムーズに主系、従系が切り替わるＣＰＵ２重化制御システムの提供を目的とする。

【０００６】

さらに、プログラム更新を行うための機能を別途備えずに、プラントシステムを停止させることなくプラント制御用プログラムを更新できるＣＰＵ２重化制御システムの提供を目的とする。

【課題を解決するための手段】

【０００７】

本発明では、2重化制御システムにおいて、待機系である従系のＣＰＵが実行系である主系のＣＰＵからスヌープ（Snoop）するデータを、主系のＣＰＵが取得する制御対象からの制御情報とすること、さらに、有利的には従系の制御周期の位相を主系の制御周期の位相より進めることを特徴とする。

【発明の効果】

【０００８】

本発明は、制御を実行している主系のＣＰＵが取得する制御対象からの制御情報を、待機状態である従系のＣＰＵがスヌープ（Snoop）することにより、ＣＰＵ２重化制御システムにおいて必要な同期化（ホットスタンバイ）に際し、従系のＣＰＵが新たに制御対象からの制御情報取得の為の処理を行うことなく、同期化（ホットスタンバイ）することができ、さらに、応答速度を犠牲にすることなくＣＰＵ２重化制御システムが可能であり、障害発生時に速やかに主系のＣＰＵ、従系のＣＰＵ切り替え処理が可能となる効果を有する。

【０００９】

また、従系のＣＰＵを主系のＣＰＵに対してその位相を進めておくことにより、従系のＣＰＵは入力待機状態になっているのでより早い応答が可能となる効果を奏する。

【００１０】

また、主系のＣＰＵと従系のＣＰＵとの一致化バスがシリアル回線であるので、パラレルバスに比べてノイズに強く、長距離伝送が可能になる効果を有する。

【００１１】

また、主系のＣＰＵが取得する制御対象からの制御情報を、待機状態である従系のＣＰＵが同時にスヌープ（Snoop）しているので、プラントシステムを停止せずにプラント制御用プログラムの更新に際しては、プログラム更新を行うための別途機能を不要にする効果を有する。

【発明を実施するための最良の形態】

【００１２】

次に、本発明を実施するための最良の形態（以下、「実施形態」という）について図面を参照して説明する。

【００１３】

図１はＣＰＵ２重化制御システムのブロック図、図２乃至図３はＣＰＵ２重化制御システムの通常時におけるタイムチャート、図４は主系のＣＰＵ障害発生における従系のＣＰＵへの切替処理を示すタイムチャート、図５は従系のＣＰＵが主系のＣＰＵからスヌープ（Snoop）確認要求を受信できないときに従系のＣＰＵへの切替処理を示すタイムチャー

トである。
【００１４】
図1における当該２重化制御システムは、ＣＰＵ１、ＣＰＵ１と同一のシステムであるＣＰＵ２、ＣＰＵ１とＣＰＵ２に接続される入出力機器であるＩ／Ｏ３、Ｉ／Ｏ３につながる制御対象４、ＣＰＵ１とＣＰＵ２及びＩ／Ｏ３を接続する通信路５、ＣＰＵ１とＣＰＵ2を接続する通信路６等から構成される。
【００１５】
前記通信路５は、RS-485等に代表されるバスマスタが存在するバスＩ／O3に対してＣＰＵ１とＣＰＵ２とが通信可能であり、しかも2重化制御を行うＣＰＵ１とＣＰＵ２とが同期化する際に使用可能であり、さらに、前記バスマスタ権は、主系となるＣＰＵが保持し、同期が要請されるＩ／０３との通信に専ら使用される。
【００１６】
また、前記通信路６は、ＣＰＵ1とＣＰＵ２間で通信可能な通信路であり、2重化制御を行うＣＰＵ１とＣＰＵ２で同期をあまり要請されない通信に専ら使用される、例えば、イーサネット（登録商標）に代表される通信路である。
【００１７】
次に、本発明におけるＣＰＵ2重化制御システムの処理について、ＣＰＵ１を主系、ＣＰＵ２を従系として説明する。
【００１８】
主系であるＣＰＵ１は実際に制御対象に対して制御指示を行うＣＰＵであり、従系であるＣＰＵ２は制御対象への制御を行わず、主系のＣＰＵに障害が発生した際に切り替わるために待機状態にあるＣＰＵである。
【００１９】
なお、いずれのＣＰＵを最初に主系とするかの決定は、本発明におけるＣＰＵが同一の機能を有してＣＰＵ間において主系／従系の切り替えが可能であるから、予め何れかが主系のＣＰＵ、従系のＣＰＵであると決定しても良いし、あるいは、いずれかのＣＰＵが主系／従系であるかを決定せずに、例えば早く起動が完了したＣＰＵを主系としても良い。
【００２０】
ＣＰＵ２重化制御における通常制御の例を図２のタイムチャートを用いて説明する。
【００２１】
当該図において、点線で示した矢印は通信路６を介した通信を示し、実線で示した矢印は通信路５を介した通信を示す。
【００２２】
最初に主系となったＣＰＵ１は、ＣＰＵ２に対し制御開始のためのコマンドである制御開始信号１１を、通信路６を介して送信する。
【００２３】
制御開始信号１１を受信した従系のＣＰＵ２は、制御開始確認応答１２を主系のＣＰＵ１に対して送信すると、主系のＣＰＵ１、従系のＣＰＵ２はそれぞれ制御を開始する為の設定及び初期化の処理１３を行う。
【００２４】
初期化処理１３終了後、主系のＣＰＵ１及び従系のＣＰＵ２はそれぞれ自己が正常に動作しているかの自己診断１４を行った結果を主系のＣＰＵ１から従系のＣＰＵ２に対し、自己診断結果１５として通信路６を介して送信する。
【００２５】
ＣＰＵ１からの診断結果１５を受信したＣＰＵ２は、自己診断結果１６を主系のＣＰＵ１に対して通信路６を介して送信する。
【００２６】
主系のＣＰＵ１の自己診断の結果、異常がある場合は従系のＣＰＵ２と主系、従系切り替え動作を行うこととなるが、この処理については後述する。
【００２７】

従系であるＣＰＵ２の自己診断１４の結果、異常がある場合には従系のＣＰＵ２への切り替え動作を禁止し、また、各々の自己診断１４の結果、お互いに問題がなければ、主系のＣＰＵ１は、従系のＣＰＵ２に対し同期化要求１７を通信路５を介して送出する。

【００２８】

当該同期化要求１７を受信した時点において従系のＣＰＵ２は、トリガをかけて同期化処理１８を行い、従系のＣＰＵ２制御周期２１を開始させると共に同期化確認応答１９を主系のＣＰＵ１に対して、通信路５を用いて送出する。

【００２９】

当該同期化確認応答１９を受信した時点で、主系のＣＰＵ１は同期処理２０を行うと共にＣＰＵ1制御周期２２を開始する。この処理により、主系のＣＰＵ１は自己の制御周期の位相が、従系のＣＰＵ２に対して相対的に遅れることとなる。

【００３０】

すなわち、従系のＣＰＵ２は、主系のＣＰＵ１からの同期化要求１７を受信した時点でＣＰＵ２制御周期２１が開始されるのに対し、主系のＣＰＵ１は従系のＣＰＵ２からの同期化確認応答１９を受けてからＣＰＵ１制御周期２２が開始されるので、当該ＣＰＵ１制御周期２２の開始は、従系のＣＰＵ２が同期化要求１７を受信した時点と同期化確認応答１９を主系のＣＰＵ１が受信した時点との時間だけＣＰＵ１制御周期２２の開始が遅れることとなる。

【００３１】

したがって、従系のＣＰＵ2は主系のＣＰＵ１に対して、相対的に位相が進んだ状態となるので、従系のＣＰＵ２は主系のＣＰＵ１に対して同期処理の後に行われる通信及び制御に必要な処理等を主系のＣＰＵ１に対して先行して行うことが出来、ＣＰＵ１より先行して入力待機状態になる。

【００３２】

その後、主系であるＣＰＵ1は、従系であるＣＰＵ2に対しデータのスヌープ（Snoop）可能な状態であるかを確認するスヌープ（Snoop）可否確認３１を、通信路５を介して送出し、当該スヌープ（Snoop）可否確認３１を受信した従系のＣＰＵ２は、主系のＣＰＵ１に対してスヌープ（Snoop）可能確認応答３２を通信路５を介して主系のＣＰＵ１に対し送信することとなるが、従系のＣＰＵ２は主系のＣＰＵ１に対して位相が相対的に先行しているので、スヌープ（Snoop）可能状態にてスタンバイしていることとなる。

【００３３】

したがって、スヌープ（Snoop）可否確認３１を受信したら速やかにスヌープ（Snoop）可能確認応答３２を、通信路５を介して主系であるＣＰＵ１に対し送信することができる。

【００３４】

すなわち、従系のＣＰＵ2がスヌープ（Snoop）でスタンバイしているので、主系のＣＰＵ１は従系のＣＰＵ2がスヌープ（Snoop）状態になるまで待機する必要がなくなり、直ちに従系のＣＰＵ2からスヌープ（Snoop）可能確認応答３２を受信することができるので、主系のＣＰＵ１は応答速度を犠牲にすることがなくなる。

【００３５】

次に、主系のＣＰＵ１はＩ／Ｏ3に対し、制御上必要となる制御対象の入力情報の送出を要求する入力情報送出要求３３を送出すると、Ｉ／Ｏ３は、主系のＣＰＵ１からの入力情報送出要求３３を受けて制御対象４から取得した制御対象情報３４を主系のＣＰＵ1に対して通信路５を用いて送出する。

【００３６】

このとき従系のＣＰＵ２は、Ｉ／Ｏ3から主系のＣＰＵ１に対して通信路５を介して送出される制御対象情報３４を同時にスヌープ（Snoop）する。

【００３７】

主系のＣＰＵ１はＩ／Ｏ3から得た制御対象４の入力情報をもとにＩ／Ｏ3に対し制御命令３５を行うが、従系のＣＰＵ２はＩ／Ｏ3に対しては制御命令を行わない。

【００３８】
当該制御命令３５を受信したＩ／Ｏ３は、制御命令確認応答３６を、主系のＣＰＵ1に対して通信路５を用いて送信する。
【００３９】
ここで、主系のＣＰＵ１は、従系のＣＰＵ２に対して当該制御命令３５に対応するＩ／Ｏ３から送信された制御対象の情報をスヌープ（Snoop）できたか否かを確認するスヌープ（Snoop）確認要求３７を通信路５を介して出力し、当該スヌープ（Snoop）確認要求３７を受信した従系のＣＰＵ2は、スヌープ（Snoop）に成功した場合はスヌープ（Snoop）確認応答３８を主系のＣＰＵ１に対して送出する。
【００４０】
主系のＣＰＵ１は、従系のＣＰＵ２からのスヌープ（Snoop）確認応答３８に基づいて従系のＣＰＵ２がスヌープ（Snoop）に成功した否かを確認し、従系のＣＰＵ２が連続してスヌープ（Snoop）に複数回失敗した場合には、従系のＣＰＵ2には何らかの障害が発生している状態であると判断して、主系のＣＰＵ１から従系のＣＰＵ２への主系、従系切り替え動作を以後禁止する。
【００４１】
一方、従系のＣＰＵ２がスヌープ（Snoop）に成功したことを確認した主系のＣＰＵ１は、先に示したそれぞれ自己が正常に動作しているかの自己健全情報のやりとりを行い、自己診断結果１５及び１６、及び同期化要求処理である同期化処理２０を行う。
【００４２】
以上までの処理を１周期として行うことにより、主系のＣＰＵ１に対して、従系のＣＰＵ２の制御周期の位相を相対的に進め、かつ、主系のＣＰＵ１が制御対象から取得した情報を従系のＣＰＵがスヌープ（Snoop）することにより、制御対象に対する主系のＣＰＵ１の応答速度を犠牲にすることなくＣＰＵ２重化制御システムにおけるホットスタンバイが可能となる。
【００４３】
ここで、主系のＣＰＵ１に対して進める従系のＣＰＵ２の制御周期の位相は、ＣＰＵ１制御周期２２の全期間に亘って主系のＣＰＵ１が制御動作を行うわけではないので、主系のＣＰＵ１の制御動作終了時点と主系のＣＰＵ１制御周期２２の終了時点との範囲の時間内で適宜位相を遅らせて良い。
【００４４】
また、ここでいうスヌープ（Snoop）とは、特許文献２に記載（参照）するように、データを分岐信号線を介して盗み読むことである。
【００４５】
次に主系のＣＰＵ１が何らかの理由にて障害が発生してシステムダウンした場合の切り替えについてＣＰＵ１を主系、ＣＰＵ２を従系として図４のタイムチャートに基づいて説明する。
【００４６】
前記のように、主系のＣＰＵ１及び従系のＣＰＵ２はお互いに通信路６を介して自己が健全であるか否かの情報である自己診断結果１５，１６をやり取りするが、主系のＣＰＵ１において、自己診断によってエラーが検出され自己健全情報内にエラー情報が含まれた際の切り替え処理に関して説明する。
【００４７】
主系のＣＰＵ１の自己診断１４においてエラーが発生すると、主系のＣＰＵ１は自己に障害が発生したと判断して主系のＣＰＵ1から従系のＣＰＵ２に対して送出される自己診断結果１５内にエラー発生の情報を含めて通信路６を介して送出する。
【００４８】
当該、自己診断結果１５を受信した従系のＣＰＵ2は、自己診断結果１６を、通信路６を介して送信し、当該自己診断結果１６を受信した主系のＣＰＵ1は、その診断結果が健全であるという内容である場合は、従系のＣＰＵ２に対して主系／従系切替要求４１を通

信路６を用いて送出し、当該主系／従系切替要求４１を受信した従系のＣＰＵ2は、主系／従系切替要求確認応答４２を主系のＣＰＵ１に対して通信路６を用いて送出する。

【００４９】

次いで、主系のＣＰＵ１は、従系のＣＰＵ２からの主系／従系切替要求確認応答４２を受信した後、Ｉ／O3に対してＣＰＵ切替要求４３を送出すると、I／０３はＣＰＵ切替が行われることを確認するＣＰＵ切替要求確認応答４４を主系のＣＰＵ１へ通信路５を用いて出力する。

【００５０】

次に、主系のＣＰＵ１は、従系のＣＰＵ２に対して通信路５切替要求４５を送出すると、従系のＣＰＵ２は、当該通信路５切替要求４５を受信してその旨を確認した後に、主系のＣＰＵ１に対して通信路５切替要求確認応答４６を送出すると、当該通信路５切替要求確認応答４６を受信した主系のＣＰＵ１は通信路５のマスタ権移行の処理を行って通信路５切替実行４７の一連の処理により、通信路５のマスタ権が主系のＣＰＵ１から従系のＣＰＵ２に切り替わり、従系のＣＰＵ２は、主系のＣＰＵ１の通信路５切替実行４７を以て従系から主系へと切り替わる。

【００５１】

この切替は、主系のＣＰＵ１が取得していた制御対象４の制御情報等をスヌープ（Snoop）処理によりあらかじめ従系のＣＰＵ２が取得している為、主系としてシステムを制御するにあたり新たに制御情報等を取得することがなく速やかに主系のＣＰＵとして動作することが可能となる。

【００５２】

ここでは、主系のＣＰＵ１の自己診断結果１５により主系／従系切替要求４１を主系のＣＰＵ１が出力したが、主系のＣＰＵ１に対し外部からのコマンドにより主系／従系切替要求４１を発行し、マニュアルにて主系／従系切替を実行することも可能である。

【００５３】

一方、このような自己診断結果にエラーがない場合においては、前記通常状態の時と同様に、従系となったＣＰＵ1の制御周期の位相を主系となったＣＰＵ2の位相に対し進めることが可能である。

【００５４】

すなわち、主系と従系が入れ替わって主系となったＣＰＵ２は、前記通常状態の主系のＣＰＵ１と同様の作用、すなわち、同期化要求１７を従系のＣＰＵとなったＣＰＵ１に対し通信路５を介して送信すると、従系となったＣＰＵ１は、同期化要求１７を受信した時点においてトリガをかけて同期化処理１８を行い、ＣＰＵ１制御周期２２を開始させると共に、同期化確認応答１９を主系となったＣＰＵ２に対して、通信路５介して送出すると、当該同期化確認応答１９を受けて主系となったＣＰＵ２は、同期化処理２０を行い、ＣＰＵ２制御周期２２を開始することにより、主系となったＣＰＵ２は自己の制御周期の位相を、従系となったＣＰＵ１に対して相対的に遅らせることは、前記通常状態の時の動作と同様である。

【００５５】

次にＣＰＵの主系／従系が切り替わるもう一つの条件となる、従系のＣＰＵが主系のＣＰＵからの制御情報のスヌープ（Snoop）要求を連続して受信できない場合の処理について図5を用いて説明する。

【００５６】

制御周期の中において、従系のＣＰＵ２は主系のＣＰＵ１が正常動作しているときには、一定周期にて送られる情報である、スヌープ（Snoop）可否確認３１、スヌープ（Snoop）確認要求３７及び、自己診断結果１５の処理が複数回連続して確認できない場合、例えば、通信路５及び通信路６においてあらかじめ設定されている通信路のタイムアウト検出により確認することができない場合に、従系のＣＰＵ2は、主系のＣＰＵ１が何らかの異常状態にあると判断する。

【００５７】

すると、従系のＣＰＵ２は、主系のＣＰＵに自ら移行し、通信路５のマスタ権を自ら取得し、通信路５を介してI／０３に対しＣＰＵ主系／従系切り替えの指示となるＣＰＵ切替要求５１を出力し、当該ＣＰＵ切替要求５１を受信したI／０３は、ＣＰＵ切替確認応答５２をＣＰＵ２に対して通信路５を介して出力すると、ＣＰＵ２は主系へ切り替わって制御を行い、これ以降において、異常状態と判断されたＣＰＵ１への切り替え処理は行われない。

【００５８】

以上のように、主系のＣＰＵの応答が無くなった状態においても、ＣＰＵの主系／従系のＣＰＵ切替が可能となる。

【００５９】

本ＣＰＵ２重化制御システムを用いることにより、ＣＰＵが制御しているプラントを止めることなく、ＣＰＵ制御用プログラムの入れ替え処理について、主系をＣＰＵ１、従系をＣＰＵ２として説明する。

【００６０】

システムを止めることなくプログラムを更新するときは、従系のＣＰＵ２に対してプログラムの更新を行うこととなるが、従系のＣＰＵ２は、通常前記のように（参照）直接システムの制御は行っていないので、システムを止めることなく、プログラムの入れ替ができ、さらに、従系のＣＰＵ２は、主系のＣＰＵ１がＩ／O3より得た制御情報をスヌープ（Snoop）により取得しているので、システムを制御している主系のＣＰＵ１と同一態様で動作することができる。

【００６１】

そこで、

プログラムを更新した従系のＣＰＵ２は、主系のＣＰＵ１からスヌープ（Snoop）して得た制御情報を元に、更新したプログラムを用いて作成したシステムへの制御命令と、主系のＣＰＵ１が出力したシステムへの制御命令とを比較し、当該比較結果である、制御対象への制御命令が一致、もしくは、想定される結果であれば、更新したプログラムはシステムを制御する上において問題ないと判断して、主系のＣＰＵ１に対し、外部コマンドにて主系／従系切替要求４１を発行させて前記したように主系／従系を切り替える。

【００６２】

すると、プラントなどのシステムを止めることなく導入当初に発生すると想定されるプログラムの入れ替え作業を、従系のＣＰＵにおいてあらかじめ評価したうえで行うことが出来、プログラム入れ替えに伴うシステム停止等の事故を回避することが可能となる。

【図面の簡単な説明】

【００６３】

【図１】ＣＰＵ２重化制御システムのブロック図

【図２】ＣＰＵ２重化制御システムの通常時のタイムチャート

【図３】ＣＰＵ２重化制御システムの通常時のタイムチャート

【図４】主系のＣＰＵ障害発生による従系のＣＰＵへの切替処理を示すタイムチャート

【図５】従系のＣＰＵが主系のＣＰＵからスヌープ（Snoop）確認要求を受信できないときに、従系のＣＰＵを主系のＣＰＵに切替える処理を示すタイムチャート

【符号の説明】

【００６４】

１　ＣＰＵ1

２　ＣＰＵ２

３　Ｉ／Ｏ

４　制御対象

１１　制御開始信号

１２　制御開始確認応答

１３　設定及び初期化処理

１４　自己診断

１５、１６　自己診断結果
１７　同期化要求
１８　ＣＰＵ２同期化処理
１９　同期化確認応答
２０　ＣＰＵ1同期化処理
２１　ＣＰＵ２制御周期
２２　ＣＰＵ１制御周期
３１　Snoop開始要求
３２　Snoop開始確認応答
３３　制御対象入力情報要求
３４　制御対象情報出力
３５　制御命令
３６　制御命令確認応答
３７　スヌープ（Snoop）確認要求
３８　スヌープ（Snoop）確認応答
４１　主系/従系切替要求
４２　主系/従系切替要求確認応答
４３　ＣＰＵ切替要求
４４　ＣＰＵ切替確認応答
４５　通信路５切替要求
４６　通信路５切替要求確認応答
４７　通信路５切替
５１　ＣＰＵ切替要求
５２　ＣＰＵ切替確認応答

【図１】

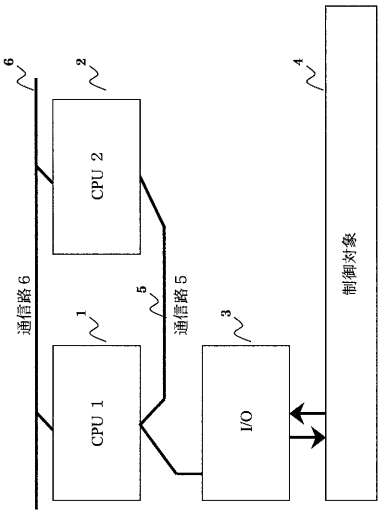

図1

【図２】

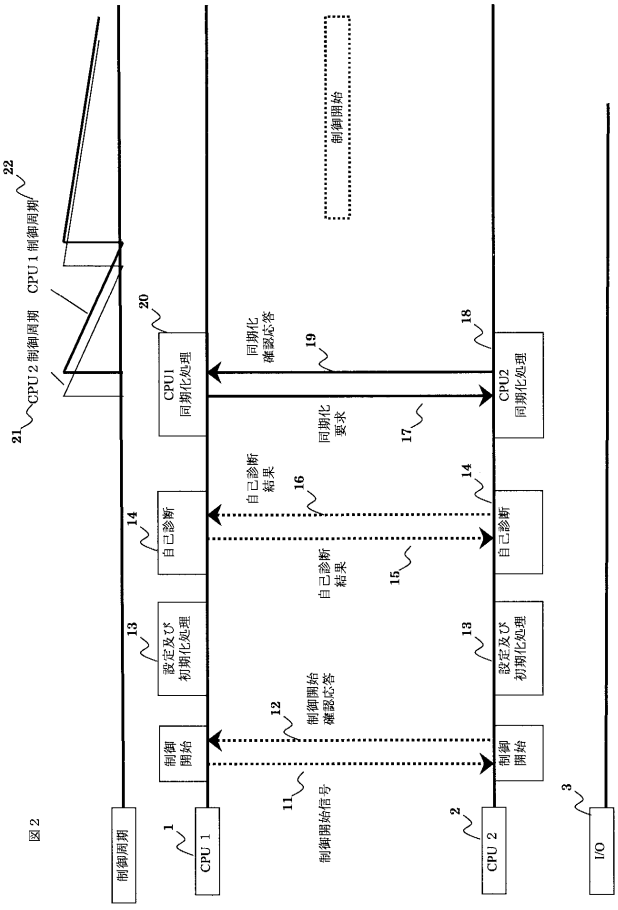

図2

【図３】

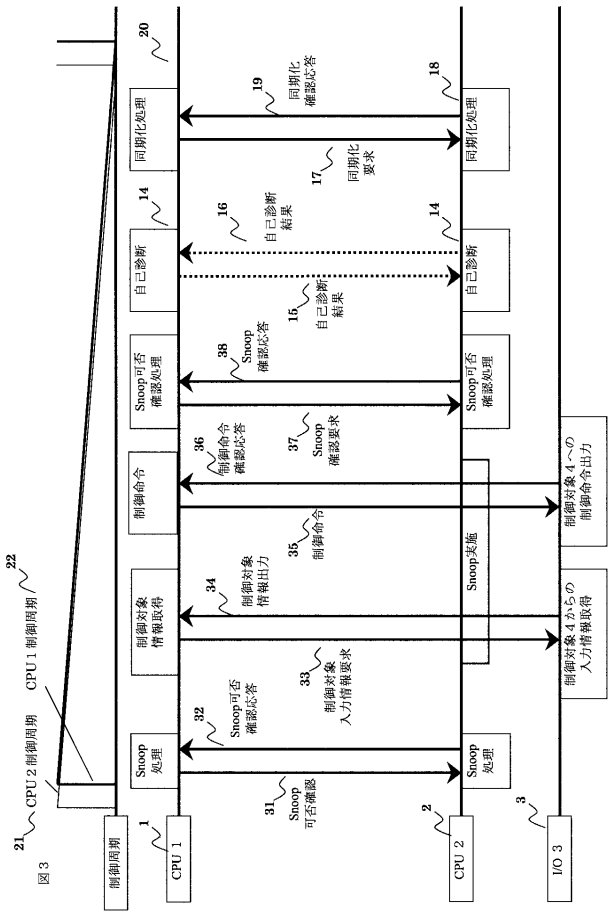

図３

【図４】

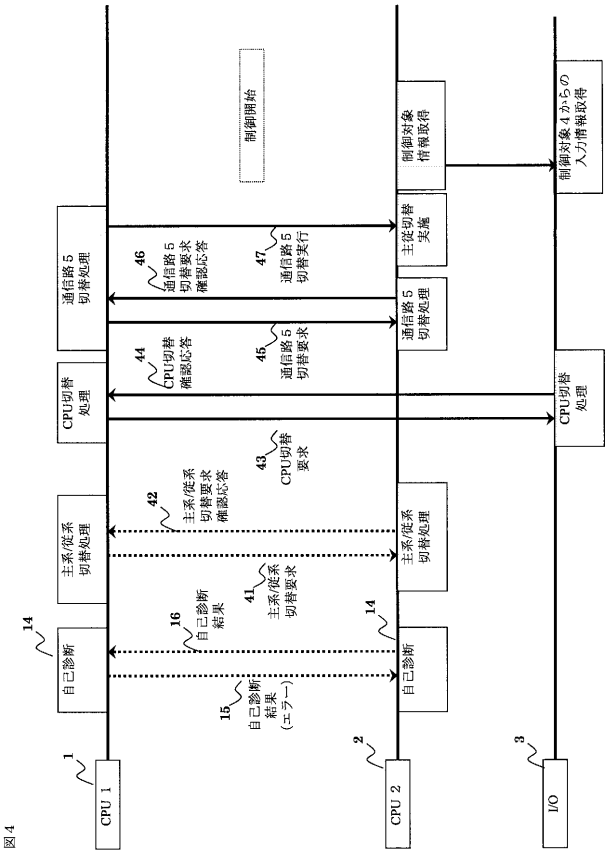

図４

【図５】

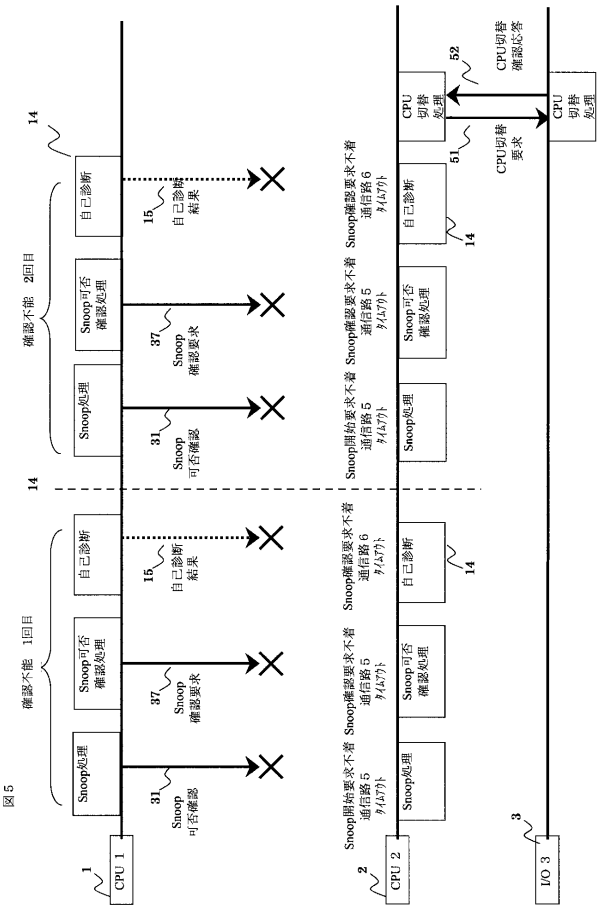

図５

フロントページの続き

(72)発明者　阪東　明
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　大塚　祐策
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　清藤　康弘
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　小林　英二
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　小野塚　明弘
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　船木　覚
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　石川　雅一
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　益子　英昭
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　関　裕介
茨城県日立市大みか町五丁目１番２６号　株式会社日立情報制御ソリューションズ内
(72)発明者　笹木　亘
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　益子　直也
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　中野　晃博
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　小倉　真
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　小澤　彰一
茨城県日立市大みか町五丁目２番１号　株式会社日立製作所情報制御システム事業部内
(72)発明者　岩崎　遊
茨城県日立市大みか町五丁目１番２６号　株式会社日立情報制御ソリューションズ内

Ｆターム(参考) 5B034 BB02　BB17　CC01　DD02　DD06