

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3884643号

(P3884643)

(45) 発行日 平成19年2月21日(2007.2.21)

(24) 登録日 平成18年11月24日(2006.11.24)

(51) Int. Cl.

F I

G05B 9/03 (2006.01)

G05B 9/03

G06F 11/18 (2006.01)

G06F 11/18 310E

請求項の数 2 (全 20 頁)

(21) 出願番号	特願2001-341027 (P2001-341027)	(73) 特許権者	000006666
(22) 出願日	平成13年11月6日 (2001.11.6)		株式会社山武
(65) 公開番号	特開2003-140707 (P2003-140707A)		東京都千代田区丸の内2丁目7番3号
(43) 公開日	平成15年5月16日 (2003.5.16)	(74) 代理人	100064621
審査請求日	平成16年6月1日 (2004.6.1)		弁理士 山川 政樹
		(72) 発明者	大塚 尚樹
			神奈川県横浜市神奈川区新浦島町1丁目1番地32 山武産業システム株式会社内
		審査官	佐々木 一浩
		(56) 参考文献	特開平08-202402 (JP, A)
			特開平03-071201 (JP, A)
			特開平08-123520 (JP, A)
			特開昭60-229107 (JP, A)
			最終頁に続く

(54) 【発明の名称】 プロセス制御装置

(57) 【特許請求の範囲】

【請求項1】

プロセス制御のための制御演算を行うコントローラと、ネットワークを介して上位装置から受信したリクエストを前記コントローラへ通知するとともに、当該リクエストに対する前記コントローラからの応答を前記上位装置へ送信するネットワークインタフェースと、前記コントローラでの制御演算に基づいてI/Oバスに接続されているプロセス機器へ制御データを送信するとともに、前記プロセス機器から計測データを受信するプロセス機器インタフェースとを有し、前記ネットワークインタフェースを介して受信された前記リクエストに基づき前記コントローラで制御演算を行い、得られた制御データを前記プロセス機器インタフェースを介して前記プロセス制御機器へ送信することにより前記プロセス機器を制御し、前記プロセス制御機器から得られた計測データを前記コントローラから応答として前記ネットワークインタフェースへ出力し、前記ネットワークインタフェースから前記上位装置へ応答として送信するプロセス制御装置において、

前記コントローラが、前記ネットワークインタフェースと前記プロセス機器インタフェースとの間に並列的に設けられ、それぞれ個別のクロックに基づき所定の制御周期で繰り返し動作し、当該プロセス制御装置内で生成されて前記コントローラへ共通に供給された複数bitの平行信号を参照して、前記制御周期を分割して設けられた複数のタイミングのうち前記bitのコード値に対応するタイミングに同期させて各種処理を実行するN個（Nは3以上の自然数）のコントローラから構成されていることを特徴とするプロセス制御装置。

10

20

【請求項 2】

請求項 1 記載のプロセス制御装置において、

前記 I O バスの多重化に応じて、前記パラレル信号を生成して前記各コントローラへ供給する処理タイミング生成部をそれぞれ有する複数のプロセス機器インタフェースをさらに備え、

前記各コントローラは、前記各プロセス機器インタフェースのうち現用系として動作しているプロセス機器インタフェースからのパラレル信号を参照し、その b i t のコード値に対応するタイミングに同期させて各種処理を実行する

ことを特徴とするプロセス制御装置。

【発明の詳細な説明】

10

【0001】**【発明の属する技術分野】**

本発明は、プロセス制御装置に関し、特に冗長構成を有するプロセス制御装置に関する。

【0002】**【従来の技術】**

従来、プラントなどのプロセス制御に用いるプロセス制御装置において、高信頼性を要求される場合には制御演算を行うコントローラを複数設えた冗長構造を持つものが用いられている。

このような冗長構造をもつ従来のプロセス制御装置は、主系と従系の 2 つのコントローラを備え、主系コントローラに異常が発生すると待機中の従系コントローラに切り替わる二重化待機冗長方式が一般的である。

20

【0003】

しかしながら、このような二重化待機冗長方式のプロセス制御装置は、コントローラが主系から従系に切り替わる切り替わる際、主系の停止を検出後、待機していた従系が処理を開始するため、制御空白時間が生じ、プロセスの制御周期が短いプロセス制御には使用できないという問題があった。

このため、稼働中のコントローラに異常が生じた場合でも制御空白時間の生じないプロセス制御装置が求められていた。

【0004】

これに対して、同一構成の 3 つ以上のコントローラを並列動作させ、これらコントローラの制御出力について多数決判定を行い、その判定結果を用いてプロセス制御を行う方式が考えられる。

30

通常、プロセス制御装置では、上位の監視装置からの処理要求や下位のプロセス機器のための制御をタイムシェアリングで処理しているため、異なる制御演算処理をわずかな時間で行っている。したがって、上記方式で所望の制御演算結果を得るためには、各コントローラから同一の制御演算処理に対する制御演算結果を得る必要がある。

そのため、外部からハードウェアによる動作クロックを入力することにより複数のコントローラ間で処理を同期させる方法が、通常使われている。

【0005】**【発明が解決しようとする課題】**

40

しかしながら、上記のような C P U などのマイクロプロセッサを用いるコントローラにおいて、複数のコントローラ間の処理をハードウェア的に完全に同期させると、ソフトウェアの動作タイミングによって発生するバグがあった場合に、各コントローラのすべてにおいてタイミングずれの不具合が発生してしまうという問題点があった。

本発明は、このような課題を解決するためのものであり、並列動作する 3 つ以上のコントローラの制御演算結果を多数決判定してプロセス制御する際、特定動作タイミングに起因して発生する不具合の影響を回避できるプロセス制御装置を提供することを目的とする。

【0006】**【課題を解決するための手段】**

50

上述した課題を解決するために、本発明にかかるプロセス制御装置は、プロセス制御のための制御演算を行うコントローラと、ネットワークを介して上位装置から受信したリクエストをコントローラへ通知するとともに、当該リクエストに対するコントローラからの応答を上位装置へ送信するネットワークインタフェースと、コントローラでの制御演算に基づいてI/Oバスに接続されているプロセス機器へ制御データを送信するとともに、プロセス機器から計測データを受信するプロセス機器インタフェースとを有し、ネットワークインタフェースを介して受信されたリクエストに基づきコントローラで制御演算を行い、得られた制御データをプロセス機器インタフェースを介してプロセス制御機器へ送信することによりプロセス機器を制御し、プロセス制御機器から得られた計測データをコントローラから応答としてネットワークインタフェースへ出力し、ネットワークインタフェースから上位装置へ応答として送信するプロセス制御装置であって、コントローラが、ネットワークインタフェースとプロセス機器インタフェースとの間に並列的に設けられ、それぞれ個別のクロックに基づき所定の制御周期で繰り返し動作し、当該プロセス制御装置内で生成されてコントローラへ共通に供給された複数bitの平行信号を参照して、制御周期を分割して設けられた複数のタイミングのうちそのbitコード値に対応するタイミングに同期させて各種処理を実行するN個（Nは3以上の自然数）のコントローラを備えるようにしたものである。

10

【0008】

また、I/Oバスの多重化に応じて、平行信号を生成して各コントローラへ供給する処理タイミング生成部をそれぞれ有する複数のプロセス機器インタフェースを備え、各コントローラで、各プロセス機器インタフェースのうち現用系として動作しているプロセス機器インタフェースからの平行信号を参照し、そのbitコード値に対応するタイミングに同期させて各種処理を実行するようにしてもよい。

20

【0009】

【発明の実施の形態】

次に、本発明の実施の形態について図面を参照して説明する。

図1は、本発明の第1の実施の形態にかかるプロセス制御装置の構成を示すブロック図である。同図において、プロセス制御装置1は、1つのネットワークインタフェース10と、3つの同一構成のコントローラ20（コントローラA、コントローラB、コントローラC）と、1つのプロセス機器インタフェース30を備えている。

30

【0010】

ネットワークインタフェース10と3つのコントローラ20は、内部LAN（Local Area Network）40に接続されており、相互通信可能に構成されている。プロセス機器インタフェース30は、3つのコントローラ20と個別にUSB（Universal Serial Bus）などのデータ通信路50で接続されており、これらとの間で1対1の通信が可能に構成されている。また、ネットワークインタフェース10と3つのコントローラ20とプロセス機器インタフェース30は、図示しないバスに接続されている。さらに、ネットワークインタフェース10は、監視装置2が接続された外部の制御LAN3（ネットワーク）に接続されており、プロセス機器インタフェース30は、プロセス機器4が接続された外部のI/Oバス5に接続されている。

40

【0011】

このような構成において、プロセス制御装置1のネットワークインタフェース10では、制御LAN3を介して受信した監視装置2からのリクエストを3つのコントローラ20の各々に与えると同時に、このリクエストに対する3つのコントローラ20の各々からの応答の中から多数決によって定まる1つの応答を制御LAN3に出力して監視装置2へ回答する。

3つのコントローラ20の各々では、ネットワークインタフェース10から与えられたリクエストの処理を行い、リクエストに対する応答をネットワークインタフェース10に出力するとともに、プロセス制御の演算を行い、演算結果を制御データとしてプロセス機器インタフェース30に出力する。

50

【0012】

プロセス機器インタフェース30では、3つのコントローラ20の各々から出力される制御データの中から多数決によって定まる1つの制御データをI/Oバス5に出力してプロセス機器4を制御する。

このため、コントローラ20の1つに異常が生じ、他の2つと異なる応答や制御データを出力した場合であっても、監視装置2やプロセス機器4へ正常な応答や制御データを途切れることなく出力することができる。

【0013】

次に、ネットワークインタフェース10について図2を参照して詳細に説明する。図2は、ネットワークインタフェース10の機能ブロック図である。

10

図2に示すように、ネットワークインタフェース10は、応答／リクエスト送受信部11、負荷チェック部12、リクエスト配信部13、応答受信部14、応答比較部15および異常通知部16を備えている。

【0014】

監視装置2が制御LAN3に出力したリクエストは、応答／リクエスト送受信部11で受信され、負荷チェック部12に送られる。負荷チェック部12は、コントローラ20ごとのリクエストの処理状況（処理負荷）をチェックし、稼働中の全てのコントローラ20が送られたリクエストを受け入れ可能な状態のときは、応答／リクエスト送受信部11から送られたリクエストをリクエスト配信部13へ転送し、稼働中で受け入れ不可能な状態のコントローラがあるときは応答／リクエスト送受信部11へ当該リクエストの受け入れ不可を通知するとともに、このリクエストを消去する。

20

【0015】

この場合、負荷チェック部12は、コントローラ20で実行中のリクエストの数が所定数以下のとき、次のリクエストを受け入れ可能と判定する。実行中のリクエストの数は、例えば転送したリクエスト数から応答数を差し引いて求める。応答数は、後述する応答受信部14から入力される。応答／リクエスト送受信部11は、リクエストの受け入れ不可が通知されると、リクエスト元へリクエスト実行不可を通知する。

【0016】

リクエスト配信部13は、負荷チェック部12から転送されたリクエストに処理すべきタイミングを示す処理タイミング情報を付加した後、内部LAN40を介して3つのコントローラ20に同報配信する。この場合、処理タイミング情報は、1制御周期中における処理タイミングを示す3bitコードであり、リクエスト配信部13は後述するプロセス機器インタフェース30が図示しないバス上にパラレル出力する3bitの処理タイミング信号103の現在値を参照し、リクエストを実行するタイミングとして将来生成される処理タイミング信号の値を設定する。この際、各コントローラへ送信したリクエストの処理状況に基づいて、新たに受信したリクエストを実行するタイミングやリクエストの応答を送信するタイミングが選択される。リクエストの処理状況については、負荷チェック部12で管理されているものを用いればよい。

30

【0017】

一方、3つのコントローラ20の各々からリクエストの応答が送信されると、応答受信部14が受信し、応答比較部15に送るとともに、応答を返したコントローラを負荷チェック部12に通知する。この場合、応答受信部14は、受信した応答を一時的に保持し、3つのコントローラ20からの応答が揃った後、応答比較部15に送る。所定時間内に3つのコントローラ20の内の1つから応答がなかったときは、当該コントローラ20を負荷チェック部12に通知するとともに、例えばヌルデータのような予め定めた特定のデータを当該コントローラ20の応答として、他の2つの応答とともに応答比較部15に送る。

40

【0018】

ここで、所定時間内に応答がなかったときに応答のないコントローラ20を負荷チェック部12に通知するのは、負荷チェック部12で応答のないコントローラ20の未処理リクエストが処理されないまま残ることにより実行中のリクエストの数が所定数を超えて、動

50

作中の２つのコントローラがリクエストを実行できなくなることを防ぐためである。

【００１９】

応答比較部１５は、応答受信部１４から送られた３つのコントローラ２０からの応答を比較し、内容の一致する応答が２つ以上ある場合のみ応答／リクエスト送受信部１１へこの内容の応答を１つ送る。ただし、一致した内容がヌルデータのような予め定めた特定のデータであった場合は、その応答を送信しない。応答／リクエスト送受信部１１へ送られた応答は、制御ＬＡＮ３へ出力され、リクエスト元の監視装置２へ送信される。また、応答比較部１５は、他と異なる内容の応答を返したコントローラを異常通知部１６へ通知する。異常通知部１６は、応答比較部１５から通知されたコントローラに内部ＬＡＮ４０を介して異常通知を送信する。

10

【００２０】

このネットワークインタフェース１０は、制御ＬＡＮ３用のＬＡＮインタフェースと、内部ＬＡＮ４０用のＬＡＮインタフェースと、処理タイミング信号１０３入力用のデジタル入力インタフェースと、ネットワークインタフェース１０の機能を実現するプログラムを格納した不揮発性半導体メモリと、この不揮発性半導体メモリに格納されたプログラムを実行する演算処理装置（ＣＰＵ）と、プログラム実行時の一時記憶に用いられるＲＡＭと、これらを接続するバスとから構成されている。

【００２１】

次に、コントローラ２０について図３を参照して詳細に説明する。図３は、コントローラ２０の機能ブロック図である。図３に示すように、コントローラ２０は、リクエスト処理部２１、データベース２２、演算処理部２３、入出力処理部２４および異常処理部２５を備えており、データベース２２にはプロセス制御条件の設定値、計測データの現在値およびプロセス機器を制御する制御データが格納されている。この場合、内部ＬＡＮ４０に出力されたリクエストは、リクエスト処理部２１で受信されてリクエスト内容が解析され、リクエストに応じた処理がリクエストに付加された処理タイミング情報に基づいて実行される。

20

【００２２】

例えば、リクエストがプロセス制御条件の変更であれば、データベース２２の指定された制御条件の設定値をリクエストで指示された設定値に書き換え、書き換えた設定値をデータベース２２から読み出して応答内容とする。また、リクエストがプロセス条件の現在値の要求であれば、データベース２２から指定されたプロセス条件の現在値を読み出して応答内容とする。このような応答は、リクエスト処理部２１により内部ＬＡＮ４０に出力され、ネットワークインタフェース１０へ送信される。この場合、リクエスト処理は、プロセス機器インタフェース３０が出力する処理タイミング信号１０３を参照し、この信号の示す情報がリクエストに付加された処理タイミング情報と一致したときに実行する。

30

【００２３】

演算処理部２３は、所定のタイミングでデータベース２２からプロセス制御条件の設定値とプロセス条件の現在値を読み出し、現在値を設定値に合わせるために必要な制御量を算出する制御演算を行い、演算結果を制御データとしてデータベース２２に書き込む。入出力処理部２４は、所定のタイミングでデータベース２２から制御データを読み出してプロセス機器インタフェース３０へ送信する。また、プロセス機器が計測したプロセスデータの送信を要求するコマンドをプロセス機器インタフェース３０へ送信するとともに、プロセス機器インタフェース３０から送信されるプロセス機器が計測したプロセスデータを受信し、現在値としてデータベース２２に書き込む。

40

【００２４】

また、リクエスト処理部２１は、リクエストと同様にしてネットワークインタフェース１０の異常通知部１６が送信する異常通知を受信し、異常処理部２５に通知する。入出力処理部２４は、プロセス機器インタフェース３０から送信される情報に異常通知が含まれているか否か確認し、異常通知が含まれているときは、異常処理部２５に通知する。異常処理部２５は、異常通知を受けるとコントローラ２０の処理を停止させる。

50

【0025】

このコントローラ20は、内部LAN40用のLANインタフェースと、データ通信路50用の通信インタフェースと、処理タイミング信号103入力用のデジタル入力インタフェースと、コントローラ20の機能を実現するプログラムを格納した不揮発性半導体メモリと、この不揮発性半導体メモリに格納されたプログラムを実行する演算処理装置（CPU）と、データベース22の格納領域やプログラム実行時の一時記憶領域として用いられるRAMと、これらを接続するバスとから構成されている。また、このコントローラ20は、活線挿抜可能にして、3つのコントローラ20のうちの1つが故障した時はプロセス制御を継続したまま交換可能に構成されている。

【0026】

次に、プロセス機器インタフェース30について図4を参照して詳細に説明する。図4は、プロセス機器インタフェース30の機能ブロック図である。図4に示すように、プロセス機器インタフェース30は、コントローラ通信処理部31、データ比較部32、IOバス入出力処理部33、異常通知部34および処理タイミング信号生成部35を備えている。

10

【0027】

各コントローラ20がそれぞれのデータ通信路50を介して送信したコマンドや制御データは、コントローラ通信処理部31で受信され、データ比較部32に送られる。この場合、コントローラ通信処理部31は、受信データを一時的に保持し、3つのコントローラ20からの受信データが揃った後、データ比較部32に送る。所定時間内に3つのコントローラ20からの受信データが揃わなかったときは、例えばヌルデータのような予め定めた特定のデータを未受信コントローラ20の受信データとし、他の受信データとともにデータ比較部32に送る。

20

【0028】

データ比較部32は、コントローラ通信処理部31から送られた3つの受信データを比較し、内容の一致する受信データが2つ以上ある場合のみ、IOバス入出力処理部33へ内容の一致した受信データを1つ入力する。ただし、一致した内容がヌルデータのような予め定めた特定のデータであった場合は、送信しない。IOバス入出力処理部33へ送られたデータは、IOバス5へ出力され、プロセス機器4へ送信される。また、データ比較部32は、他と異なるデータを送信したコントローラを異常通知部34へ通知する。異常通知部34は、コントローラ通信処理部31を介してデータ比較部32から通知されたコントローラ20に異常通知を送信する。

30

【0029】

一方、プロセス機器4がIOバス5へ出力したデータは、IOバス入出力処理部33が受信し、コントローラ通信処理部31へ送られる。コントローラ通信処理部31は、送られたデータを3つのデータ通信路50にそれぞれ出力し、各コントローラ20に送信する。処理タイミング信号生成部35は、処理タイミング信号103を生成し、ネットワークインタフェース10と3つのコントローラ20とプロセス機器インタフェース30とに接続されているバスに出力する。

【0030】

ここで、処理タイミング信号103は、例えば3bitの平行信号であり、プロセス制御装置の1制御周期中に順次3bitの組み合わせから得られる8つの状態をとることにより、8個の処理タイミングを設けるものである。なお、処理タイミング信号103は3bitに限られるものではなく、1制御周期中に実行する処理の種類やコントローラ20の処理速度に応じて他のbit数に変更してもよい。

40

【0031】

このプロセス機器インタフェース30は、3つのデータ通信路50用の通信インタフェースと、IOバス5用インタフェースと、処理タイミング信号103出力用のデジタル出力インタフェースと、プロセス機器インタフェース30の機能を実現するプログラムを格納した不揮発性半導体メモリと、この不揮発性半導体メモリに格納されたプログラムを実

50

行する演算処理装置（CPU）と、プログラム実行時の一時記憶に用いられるRAMと、これらを接続するバスとから構成されている。

【0032】

次に、このプロセス制御装置の動作について説明する。最初に、監視装置からのリクエストがないときの動作を説明する。図5は、第1の実施の形態のプロセス制御装置の動作を説明するシーケンス図であり、監視装置からのリクエストがないときの動作を示す。図5に示すように、このプロセス制御装置は、3つのコントローラ（コントローラA、コントローラB、コントローラC）が並列して同じ処理を実行している。

【0033】

ここで、周期 n （ n は自然数）の制御周期が開始されると、各々のコントローラは、処理
タイミング信号が所定のタイミングとなったときにプロセス機器インタフェースに計測
データ要求のコマンドを送信する。プロセス機器インタフェースは、各々のコントローラから
送信されたコマンドの多数決を採って多数側のコマンドをプロセス機器に送信するととも
に、このコマンドを受信したプロセス機器から送信される計測データを受信し、各々の
コントローラに送信する。この場合、多数決の判定は、各々のコントローラから送信され
たコマンドを比較し、2つ以上のコントローラからのコマンドが一致した場合に一致した
コマンドを多数のコマンドと決することにより行う。

10

【0034】

各々のコントローラは、計測データを受信するとデータベースの現在値を更新し、所定の
処理タイミングとなったらデータベースの設定値と現在値を入力とする制御演算を行い、
データベースの制御データを演算結果で更新し、所定の処理タイミングとなったときにこ
の制御データをプロセス機器インタフェースに送信する。プロセス機器インタフェースは
、各々のコントローラから送信された制御データの多数決を採って多数側の制御データを
プロセス機器に送信し、プロセス機器を制御する。この場合も多数決の判定は、各々のコ
ントローラから送信された制御データを比較し、2つ以上のコントローラからの制御デー
タが一致した場合に一致した制御データを多数の制御データと決することにより行う。

20

【0035】

次に、監視装置からリクエストがなされたときの動作を説明する。図6は、第1の実施の
形態のプロセス制御装置の動作を説明するシーケンス図であり、監視装置から計測デー
タの現在値を要求するリクエストがなされたときの動作を示す。図6に示すように、3つの
コントローラ（コントローラA、コントローラB、コントローラC）が周期 n （ n は自然
数）の制御周期において同じ処理を並列に実行しているときに、ネットワークインタフ
ェースがリクエストを受信すると、ネットワークインタフェースは、3つのコントローラの
負荷チェックを行う。

30

【0036】

負荷チェックの結果、3つのコントローラがともにリクエストを実行可能であれば、リク
エストに処理タイミング情報を付加して3つのコントローラに送信する。この場合、ネッ
トワークインタフェースは、リクエストを実行するタイミングを指定する情報を処理タイ
ミング情報として付加するが、リクエストを実行するタイミングには、処理が割り当てら
れていない処理タイミングを割り当てる。3つのコントローラは、各々リクエストを受信
すると、処理タイミング情報で指定された処理タイミングでリクエストを実行する。

40

【0037】

この場合、3つのコントローラは、それぞれデータベースの現在値更新後の処理タイミ
ングで各々のデータベースにアクセスし、現在値を読み出してネットワークインタフェ
ースへリクエストに対する応答として送信する。ネットワークインタフェースは、各々のコ
ントローラから送信された応答を受信し、これらの内容（現在値）を比較し、内容の一致す
る2つ以上の応答のうちのいずれか1つを制御LANに出力し、監視装置に送信する。こ
の場合、応答内容は3つとも一致しているので、3つのコントローラは、以後も図5で説
明したものと同じ動作を行う。ここで、3つのコントローラのリクエスト処理以外の処理
タイミングにおける処理動作と、プロセス機器インタフェースの動作は、図5で説明した

50

ものと同じであるので説明を省略する。

【0038】

次に、ネットワークインタフェースが応答の不一致を検出したときの動作を説明する。図7は、第1の実施の形態のプロセス制御装置の動作を説明するシーケンス図であり、ネットワークインタフェースが3つのコントローラからの応答に不一致を検出したときの動作を示す。図7において、3つのコントローラがリクエストに付加された処理タイミング情報で指定された処理タイミングでリクエストを実行し、応答をネットワークインタフェースに送信するまでは、図6での説明と同じであるので、説明を省略する。

【0039】

ネットワークインタフェースが各々のコントローラから送信された応答を受信した後、これらの内容（現在値）を比較し、コントローラAの応答内容が他の2つのコントローラの応答内容と一致しないことを検出すると、ネットワークインタフェースは、内容が一致したコントローラBとコントローラCの応答をリクエストに対する応答として制御LANに出力し、監視装置に送信するとともに、コントローラAに異常通知を送信する。

10

【0040】

コントローラAは、異常通知を受信すると異常処理を実行し、制御演算処理からの切り離しを行う。一方、コントローラBとコントローラCは、以後も図5で説明したのと同じ動作を行う。ネットワークインタフェースとプロセス機器インタフェースは、コントローラAが切り離された後も、コントローラBとコントローラCの応答や制御データが一致している限り以前と同じ動作を行い、プロセス制御装置全体として無瞬断のプロセス制御を

20

【0041】

次に、プロセス機器インタフェースが3つのコントローラ間でコマンドや制御データに不一致を検出したときの動作を説明する。図8は、第1の実施の形態のプロセス制御装置の動作を説明するシーケンス図であり、プロセス機器インタフェースが3つのコントローラ間でコマンドや制御データに不一致を検出したときの動作を示す。図8において、3つのコントローラが所定の処理タイミングで指定された制御演算を実行し、演算結果の制御データをプロセス機器インタフェースに送信するまでは、図5での説明と同じであるので、説明を省略する。

【0042】

プロセス機器インタフェースが各々のコントローラから送信されたコマンドや制御データを受信した後、これらの内容を比較し、コントローラCの内容が他の2つのコントローラの内容と一致しないことを検出すると、プロセス機器インタフェースは、内容が一致したコントローラAとコントローラBのコマンドや制御データをIOバスに出力し、プロセス機器に送信するとともに、コントローラCに異常通知を送信する。

30

【0043】

コントローラCは、異常通知を受信すると異常処理を実行し、制御演算処理からの切り離しを行う。一方、コントローラAとコントローラBは、以後も図5で説明したのと同じ動作を行う。ネットワークインタフェースとプロセス機器インタフェースは、コントローラCが切り離された後も、コントローラAとコントローラBの応答や制御データが一致している限り以前と同じ動作を行い、プロセス制御装置全体として無瞬断のプロセス制御を

40

【0044】

以上説明したように、この実施の形態のプロセス制御装置は、制御演算処理を行うコントローラを3つ並列動作させるとともに、これらの出力するデータを比較し、2つ以上が一致している間、動作を続けるようにしたので、コントローラの1つに異常が生じて制御演算処理から切り離されても、無瞬断でプロセス制御を続行可能である。なお、この実施の形態では、コントローラを3つ並列動作させるものとして説明したが、この数は3つに限られるものではなく、3以上であればよい。並列動作させるコントローラの数3を増やすにしたがい、無瞬断のプロセス制御を続行するのに許容されるコントローラの異常発

50

生数を増やすことができる。よって、プロセス制御装置に求められる信頼性とコントローラの信頼性とに基づいて、最適な個数を選択すればよい。

【0045】

また、制御演算処理を行うコントローラを並列実行による冗長構成としたので、ネットワークインタフェースとプロセス機器インタフェースの信頼性を確保することで、コントローラに高信頼性を求めなくともプロセス制御装置全体としての信頼性を確保することができる。このため、コントローラに低価格の市販製品を用いることができ、信頼性を確保したままコストを下げることができる。また、世代交代の早い市販製品の利用が可能となることで、半導体技術の向上による処理能力の向上がタイムリーにでき、製品開発期間の短縮も可能となる。

10

【0046】

また、ネットワークインタフェースとプロセス機器インタフェースにより、コントローラの出力が正常であることを常時確認しているため、プロセス制御装置として十分な信頼性が確保できるので、コントローラの使用するメモリシステムにエラー訂正機能を設けなくてもよく、またエラー訂正機能では訂正不可能な宇宙線などに起因すると考えられるマルチビットエラーにも対処でき、より低コスト化を図ることができる。

【0047】

次に、図9および図10を参照して、並列動作する各コントローラ20で処理タイミングを同期させるための構成について説明する。図9はコントローラへ供給される処理タイミング信号を示す説明図である。図10は処理タイミング信号の構成例を示すタイミングチャートである。

20

図9に示すように、各コントローラ20には、プロセス機器インタフェース30の処理タイミング信号生成部35（図4参照）からタイミング信号バス51を介して処理タイミング信号103が並列的に供給されている。各コントローラ20のリクエスト処理部21、演算処理部23および入出力処理部24（図3参照）では、この処理タイミング信号103の内容に基づき処理内容を決定している。

【0048】

処理タイミング信号103は、図10に示すように、3bitの平行信号すなわち処理タイミング信号103A～103Cからなり、1制御周期（例えば、100ms）ごとに巡回するコードが出力される。ここでは、1制御周期ごとに、コードの値に対応して8つの処理タイミング $t_1 \sim t_8$ が設けられており、各コントローラ20では、そのコードの値に基づき処理タイミングを識別子、各処理タイミングに対して予め設定されている処理、例えば、上位の監視装置からの処理要求や下位のプロセス機器のための制御を実行する。

30

【0049】

このように、各コントローラ20に対して処理タイミング信号103を並列的に供給し、各コントローラ20では、その処理タイミング信号103に基づき処理する内容を決定するようにしたので、各コントローラ20間でほぼ同期して同一内容の処理を実行することが可能となる。

これにより、従来のように、各コントローラ20で共通の動作クロックを用いてそのマイクログロッサでの処理を完全に同期させる必要がなくなり、それぞれ個別の動作クロックを用いることが可能となる。

40

【0050】

したがって、各コントローラが特定動作タイミングに起因して発生する不具合を内在している場合でも、複数のコントローラで同時に不具合が発生する確率が極めて低くなり、これらコントローラの出力／応答を多数決判定することにより、プロセス制御装置1全体に対する影響を回避でき、安定した制御動作を実現できる。

また、共通クロックを用いて各コントローラの動作を同期させるための高価な回路構成が不要となり、製品コストを大幅に削減できる。さらには、コントローラとして完全に同一構成の装置を用いる必要がなくなり、システム構成として柔軟性が得られる。

50

【0051】

なお、処理タイミング信号103に用いるコードとしては、一般的な2進数を用いてもよい。このとき、複数ビットが同時に変化するため、コントローラ20でのビット検出にばらつきが生じた場合は、瞬間的に異なるコードとして認識される場合も考えられる。本実施の形態では、処理タイミング信号103として、図10に示すようなグレイコードを用いている。このグレイコードとは、任意の値から次の値へ遷移する際、いずれか1ビットだけが変化するコードである。したがって、処理タイミングの変遷の際、ビット検出のばらつきによるコードの誤認を回避でき、各コントローラ20において、安定した処理タイミングの同期を実現できる。

【0052】

10

次に、図11および図12を参照して、ネットワークインタフェース10によるコントローラ管理処理について説明する。図11はネットワークインタフェース10から各コントローラ20へ送信されるリクエスト（メッセージ）の構成例である。図12は各コントローラでの処理スケジュール例である。

前述したように、ネットワークインタフェース10では、各コントローラ20へ処理を指示する際、リクエスト配信部13（図2参照）から内部LAN40を介してリクエストを送信する。

【0053】

このリクエストは、図11に示すように、イーサネット（登録商標）で用いるパケットの1つであり、パケットの送信先や送信元、パケット種別などパケットに関する情報を格納するヘッダ41と、リクエストの内容を示す情報を格納するデータ部42とから構成されている。

20

データ部42には、そのリクエストを識別するためのリクエストID43、リクエストの内容がコード化されたリクエスト内容44、そのリクエストを実行するタイミングを示す処理タイミング情報45が格納されている。例えば、各コントローラ20に対して設定値の変更を指示するリクエストの場合は、リクエスト内容44として、処理対象となるプロセス機器すなわち制御ポイントを示すポイントID44A、処理の種別を示す処理種別44Bとその処理に用いる処理データ44Cとがセットされている。また、処理タイミング情報45には、処理タイミング信号103で規定されるグレイコードがセットされている。

30

【0054】

各コントローラ20では、このようなリクエストをリクエスト処理部21で受信し、その内容を解析して実行する。このとき、処理の内容はリクエスト内容44から決定し、処理タイミングは処理タイミング情報45から決定する。

図11のリクエスト46の例では、処理種別44Bが「設定値更新」を示すことから、リクエスト処理部21は、処理データ44Cの「DB設定値」を取り出して、ポイントID44Aに対応する新たな設定値としてデータベース22へ書き込む。また、この処理は、プロセス機器インタフェース30からの処理タイミング信号103のグレイコードが「011」すなわち処理タイミングt3を示す場合に実行される。これにより、各コントローラ20において、次の処理タイミングt3に同期して一斉に新たな設定値が書き込まれることになる。

40

【0055】

また、図11のリクエスト47の例では、処理種別44Aが「現在値読み出し」を示すことから、リクエスト処理部21は、データベース22に格納されているプロセス機器から得られた現在値を読み出して、ネットワークインタフェース10へ通知する。また、この処理は、プロセス機器インタフェース30からの処理タイミング信号103のグレイコードが「101」すなわち処理タイミングt5を示す場合に実行される。これにより、各コントローラ20から、次の処理タイミングt5に同期して一斉に現在値が読み出されることになる。

なお、読み出された現在値は、応答によりコントローラ20からネットワークインタフェ

50

ース10へ通知される。このとき、対応するリクエストのリクエストIDもその応答により通知することにより、各応答がどのリクエストに対応するものか識別するようにしてもよく、複数のリクエストが多重して発行される場合にも対応できる。

【0056】

このように、各コントローラ20に対して処理タイミング信号103を並列的に供給し、各コントローラ20では、その処理タイミング信号103に基づき処理する内容を決定するものとし、ネットワークインタフェース10から各コントローラ20へのリクエストで、要求した処理を実行するタイミングを指定するようにしたので、各コントローラ20間ではほぼ同期して同一内容の処理を実行することが可能となり、コントローラの出力／応答を精度よく多数決判定することができる。

10

また、従来のように、各コントローラ20で共通の動作クロックを用いてそのマイクロプロセッサでの処理を完全に同期させる必要がなくなり、それぞれ個別の動作クロックを用いることが可能となる。

【0057】

したがって、共通クロックを用いて各コントローラの動作を同期させるための高価な回路構成が不要となり、製品コストを大幅に削減できる。さらには、コントローラとして完全に同一構成の装置を用いる必要がなくなり、システム構成として柔軟性が得られる。

【0058】

各コントローラ20では、このようにして、各処理タイミングごとに予め設定された処理が行われる。前述した図6のシーケンスでは、図12に示すように、まず処理タイミングt1で、プロセス機器への「計測データ要求」処理が行われ、処理タイミングt2で「リクエスト受付」処理が行われる。また、処理タイミングt3でプロセス機器から得られた計測データをデータベース22へ格納する「DB現在値更新」処理が行われ、処理タイミングt4でリクエストに基づく「DB現在値読み出し」処理が行われる。そして、処理が設定されていない処理タイミングt5の後、処理タイミングt6でプロセス機器への制御データを算出するための「制御演算」処理が行われ、処理タイミングt7でその演算結果をデータベース22へ格納する「DB制御データ更新」処理が行われ、処理タイミングt8でプロセス機器への「制御データ送信」処理が行われる。

20

【0059】

このような処理のスケジューリングは、ネットワークインタフェース10の負荷チェック部12で管理しており、新たな処理を要求する際には、処理が設定されていない処理タイミングを指定してリクエストを発行する。

30

このとき、監視装置2からの要求に基づきコントローラ20に対して新たな処理を要求する際、負荷チェック部12では、上記のようなスケジューリングに基づき受け入れ可否を判断する。過負荷であり受け入れできない場合あれば、監視装置2へ受け入れ不可を通知する。これにより、コントローラ20への過負荷を未然に回避でき、安定したプロセス制御を実現できる。

【0060】

なお、各処理タイミングでの処理は、すべてネットワークインタフェース10側からコントローラ20に対して指定してもよいが、各コントローラ20で行われる定常的な処理、例えば「計測データ要求」、「DB現在値更新」、「制御演算」、「DB制御データ更新」、「制御データ送信」などプロセス機器の制御に関する処理については、予めその処理タイミングを固定的に割り当てておくようにしてもよく、ネットワークインタフェース10でのリクエスト処理を省くことができる。

40

【0061】

次に、本発明の第2の実施の形態について説明する。この実施の形態のプロセス制御装置が第1の実施の形態と異なる点は、3つのコントローラのデータベースがそれぞれ正しいか否かの確認を行う自己診断機能を有することである。この自己診断機能は、図2で示したネットワークインタフェース10の応答／リクエスト送受信部11に、図3で示したデータベース22のデータを所定単位ごとに順次送信させるリクエストを所定間隔で発行す

50

るとともに、このリクエストに応じて応答／リクエスト送受信部 11 へ送られたデータを破棄する機能を設けることにより実現することができる。すなわち、ネットワークインタフェース 10 の機能を実現するプログラムを格納した不揮発性半導体メモリに、この自己診断機能を実現するプログラムを加えればよい。

【0062】

この場合、応答／リクエスト送受信部 11 が発行する自己診断機能のリクエストは、負荷チェック部 12 の働きによりコントローラ 20 の負荷が軽いときのみ 3 つのコントローラ 20 へ送信される。コントローラ 20 へ送信されたリクエストは、所定のタイミングでリクエスト処理部により実行され、データベース 22 のデータが所定単位ごとに読み出されて順次ネットワークインタフェース 10 へ送信させる。3 つのコントローラ 20 から送信されたデータベース 22 のデータは、応答比較部 15 で比較され、不一致の検出されたコントローラが異常通知部 16 に通知される。これにより、異常通知部 16 が当該コントローラに異常通知を送信し、異常通知を受信したコントローラは、異常処理を実行し制御演算処理からの切り離しを行う。

10

【0063】

この実施の形態によれば、3 つのコントローラ 20 のデータベース 22 のデータ比較をバックグラウンドで行うことで、データベース 22 のデータを格納する RAM のエラーなどを検出することができる。このため、頻繁に使用されない領域で発生したエラーを検出でき、異常の生じたコントローラ 20 を除くことができる。

【0064】

次に、図 13 を参照して、本発明の第 3 の実施の形態について説明する。図 13 は、本発明のプロセス制御装置にかかる第 3 の実施の形態を示すブロック図である。

20

この実施の形態のプロセス制御装置 6 が第 1 の実施の形態と異なる点は、ネットワークインタフェース 10、プロセス機器インタフェース 30、内部 LAN 40 およびデータ通信路 50 をそれぞれ冗長構成としたことである。図 13 において、このプロセス制御装置 6 は、2 つの同一構成のネットワークインタフェース 10（ネットワークインタフェース A、ネットワークインタフェース B）と、3 つの同一構成のコントローラ 20（コントローラ A、コントローラ B、コントローラ C）と、2 つの同一構成のプロセス機器インタフェース 30（プロセス機器インタフェース A、プロセス機器インタフェース B）を備えている。

30

【0065】

2 つのネットワークインタフェース 10 は、系統の異なる内部 LAN 40（A 系内部 LAN、B 系内部 LAN）を介して 3 つのコントローラ 20 と接続されており、それぞれ 3 つのコントローラ 20 と相互通信可能に構成されている。この場合、ネットワークインタフェース A は A 系内部 LAN を介し、ネットワークインタフェース B は B 系内部 LAN を介して、3 つのコントローラ 20 と接続されている。

【0066】

2 つのプロセス機器インタフェース 30 は、それぞれ 3 つのコントローラ 20 と個別にデータ通信路 50 で接続されており、これらとの間で 1 対 1 の通信が可能に構成されている。また、2 つのプロセス機器インタフェース 30 の間に制御信号線 60 が接続されており、相互に制御信号の伝達が可能に構成されている。さらに、2 つのプロセス機器インタフェース 30 は、各々が系統の異なる図示しないバスを介して 2 つのネットワークインタフェース 10 と 3 つのコントローラ 20 に接続されている。

40

【0067】

また、2 つのネットワークインタフェース 10 は、系統の異なる制御 LAN 3（A 系制御 LAN、B 系制御 LAN）を介して監視装置 2 と接続されており、それぞれ監視装置 2 と相互通信可能に構成されている。この場合、ネットワークインタフェース A は A 系制御 LAN を介し、ネットワークインタフェース B は B 系制御 LAN を介して、監視装置 2 と接続されている。2 つのプロセス機器インタフェース 30 は、各々がプロセス機器 4 の接続された外部の 2 つの I/O バス 5（A 系 I/O バス、B 系 I/O バス）に接続されている。

50

【0068】

このプロセス制御装置6は、2つのネットワークインタフェース10と3つのコントローラ20が並列に動作し、2つのプロセス機器インタフェース30の一方がマスター（現用系）として常時動作し、他方がスレーブ（待機系）として待機している。2つのネットワークインタフェース10は、それぞれ系統の異なる制御LAN3を介して受信した監視装置2のリクエストを別々に3つのコントローラ20の各々に与えるとともに、このリクエストに対する3つのコントローラ20の各々からの応答の中から多数決によって定まる1つの応答を各々が接続された制御LAN3に出力して監視装置2へ回答する。

【0069】

2つのネットワークインタフェース10が第1の実施の形態若しくは第2の実施の形態で説明したものと異なる点は、処理タイミング信号入力用のデジタル入力インタフェースを2系統有し、マスターとスレーブの両方のプロセス機器インタフェース30から出力される処理タイミング信号が入力され、マスター側の処理タイミング信号が入力されなくなったときに、スレーブ側の処理タイミング信号に基づいて処理を続行するようにしたことである。

10

【0070】

3つのコントローラ20は、各々が2つのネットワークインタフェース10から与えられた同じリクエストのうち、先に与えられたリクエストのみ処理を行い、このリクエストに対する応答を2つのネットワークインタフェース10に出力する。また、プロセス機器4に対するコマンド生成やプロセス制御の演算を行い、生成したコマンドや演算結果を制御データとして2つのプロセス機器インタフェース30に出力する。

20

【0071】

ここで、2つのネットワークインタフェース10から与えられた同じリクエストのうち、先に与えられたリクエストのみ処理を行うことができるのは、監視装置の送信するリクエストがA系とB系で同じメッセージヘッダを有するため、応答／リクエスト送受信部11が同じメッセージを受信した場合、後から受信したメッセージを先に受信したものと同一メッセージと判定し、無視するためである。

【0072】

3つのコントローラ20が第1の実施の形態若しくは第2の実施の形態で説明したものと異なる点は、内部LAN40用のLANインタフェースと、データ通信路50用の通信インタフェースと、処理タイミング信号103入力用のデジタル入力インタフェースとをそれぞれ2系統有し、応答／リクエスト送受信部11が2つの内部LAN40を介して送受信を行うことと、入出力処理部24が2つのデータ通信路50を介して送受信を行うことと、マスターとスレーブの両方のプロセス機器インタフェース30から出力される処理タイミング信号が入力され、マスター側の処理タイミング信号が入力されなくなったときに、スレーブ側の処理タイミング信号に基づいて処理を続行するようにしたことである。

30

【0073】

プロセス機器インタフェース30は、マスター側が3つのコントローラ20の各々から出力される制御データの中から多数決によって定まる1つの制御データを2つのIOバス5に出力してプロセス機器4を制御し、スレーブ側が制御信号線60を介してマスター側の動作を確認する待機動作を行う。スレーブ側のプロセス機器インタフェース30は、マスター側の動作停止を検出するとマスターに切り替わる。また、2つのプロセス機器インタフェース30は、マスター側かスレーブ側かにかかわらず各々が接続された図示しないバスに処理タイミング信号を出力している。

40

【0074】

2つのプロセス機器インタフェース30が第1の実施の形態若しくは第2の実施の形態で説明したものと異なる点は、IOバス5用インタフェースを2系統有し、制御データを2つのIOバス5に出力可能に構成したことと、制御信号線60用のデジタル入出力インタフェースを有し、マスター・スレーブ切替可能に構成したことである。マスター・スレーブ切替は、例えば、マスターのときは常時、動作中を示す信号をデジタル入出力イン

50

タフェースに出力し、スレーブのときは常時、ディジタル入出力インタフェースに入力する信号を監視し、信号の入力が途絶すると自身をマスターに切り替えるプログラムを不揮発性半導体メモリに格納しておき、CPUがこのプログラムを実行するようにしておくことにより行うことができる。

【0075】

このように構成したので、この実施の形態のプロセス制御装置は、ネットワークインタフェース10、制御LAN3および内部LAN40のいずれかで異常が生じた場合であっても、監視装置2との通信が途絶することがないので、正常に処理を続行することができる。また、動作中のプロセス機器インタフェース30に異常が生じた場合であっても、待機中のプロセス機器インタフェース30によって処理を続行することができる。このように、この実施の形態のプロセス制御装置は、コントローラ20の1つに異常が生じて制御演算処理から切り離されても、無瞬断でプロセス制御を続行可能であるだけでなく、3つのコントローラ20に接続されたインタフェース部分に異常が生じた場合にも処理を続行することができる。このため、第1の実施の形態に比べて、さらに信頼性を向上することができる。

10

【0076】

この実施の形態では、3つのコントローラ20に接続されるものをすべて冗長構成としたが、必ずしもすべてを冗長構成としなくともよい。プロセス制御装置として要求される信頼性と、3つのコントローラ20に接続されるものの個々の信頼性とを勘案して必要な箇所のみ冗長構成とすることにより、必要とされる信頼性を最小限のコスト上昇で得ることができる。例えば、プロセス制御装置を2つの同一構成のネットワークインタフェース10（ネットワークインタフェースA、ネットワークインタフェースB）と、3つの同一構成のコントローラ20（コントローラA、コントローラB、コントローラC）と、1つのプロセス機器インタフェース30とで構成してもよいし、1つのネットワークインタフェース10と、3つの同一構成のコントローラ20（コントローラA、コントローラB、コントローラC）と、2つの同一構成のプロセス機器インタフェース30（プロセス機器インタフェースA、プロセス機器インタフェースB）とで構成してもよい。

20

【0077】

次に、図14を参照して、並列動作する各コントローラ20で処理タイミングを同期させるための構成について説明する。図14はコントローラへ供給される処理タイミング信号を示す説明図である。

30

図14に示すように、各コントローラ20には、プロセス機器インタフェース30の処理タイミング信号生成部35（図4参照）からタイミング信号バス51を介して処理タイミング信号103が並列的に供給されている。各コントローラ20のリクエスト処理部21、演算処理部23および入出力処理部24（図3参照）では、この処理タイミング信号103の内容に基づき処理内容を決定している。

【0078】

処理タイミング信号103は、図10に示すように、3bitの平行信号すなわち処理タイミング信号103A～103Cからなり、1制御周期（例えば、100ms）ごとに巡回するコードが出力される。ここでは、1制御周期ごとに、コードの値に対応して8つの処理タイミング $t_1 \sim t_8$ が設けられており、各コントローラ20では、そのコードの値に基づき処理タイミングを識別子、各処理タイミングに対して予め設定されている処理、例えば、上位の監視装置からの処理要求や下位のプロセス機器のための制御を実行する。

40

【0079】

本実施の形態では、各コントローラ20および各ネットワークインタフェース10には、2つのプロセス機器インタフェース30からそれぞれ個別に処理タイミング信号103が供給される。各コントローラ20および各ネットワークインタフェース10では、2つのプロセス機器インタフェース30から供給された処理タイミング信号103のうち、いずれかマスター側の処理タイミング信号を選択して処理タイミングの決定に用いる。選択の

50

方法としては、プロセス機器インタフェース30との通信によりいずれがマスター側か判断してもよく、あるいはデフォルトでマスター側を予め決めておき、そのマスター側からの処理タイミング信号が停止するなど不具合が検出された場合に他方の処理タイミング信号へ切り替えるようにしてもよい。

【0080】

このように、プロセス機器インタフェース30を2重化した場合は、各プロセス機器インタフェース30からの処理タイミング信号103を各コントローラ20および各ネットワークインタフェース10を個別に供給し、各コントローラ20および各ネットワークインタフェース10では、2つのプロセス機器インタフェース30から供給された処理タイミング信号103のうち、いずれか現用系のものを選択して利用するようにしたので、各コ

10

【0081】

【発明の効果】

以上説明したように、本発明は、コントローラとして、ネットワークインタフェースとプロセス機器インタフェースとの間に並列的に設けられ、それぞれ個別のクロックに基づき所定の制御周期で繰り返し動作し、当該プロセス制御装置内で生成されてコントローラへ共通に供給された複数bitの平行信号を参照して、制御周期を分割して設けられた複数のタイミングのうちそのbitコード値に対応するタイミングに同期させて各種処理を実行するN個（Nは3以上の自然数）のコントローラを用いるようにしたので、各コントローラ間でほぼ同期して同一内容の処理を実行することが可能となる。

20

【0082】

これにより、従来のように、各コントローラで共通の動作クロックを用いてそのマイクロプロセッサでの処理を完全に同期させる必要がなくなり、それぞれ個別の動作クロックを用いることが可能となる。

したがって、各コントローラが特定動作タイミングに起因して発生する不具合を内在している場合でも、複数のコントローラで同時に不具合が発生する確率が極めて低くなり、これらコントローラの出力／応答を多数決判定することにより、プロセス制御装置全体に対する影響を回避でき、安定した制御動作を実現できる。

【図面の簡単な説明】

【図1】 本発明の第1の実施の形態にかかるプロセス制御装置の構成を示すブロック図である。

30

【図2】 図1のネットワークインタフェースの機能ブロック図である。

【図3】 図1のコントローラの機能ブロック図である。

【図4】 図1のプロセス機器インタフェースの機能ブロック図である。

【図5】 第1の実施の形態にかかるプロセス制御装置の動作を説明するシーケンス図である。

【図6】 第1の実施の形態にかかるプロセス制御装置の他の動作を説明するシーケンス図である。

【図7】 第1の実施の形態にかかるプロセス制御装置の他の動作を説明するシーケンス図である。

40

【図8】 第1の実施の形態にかかるプロセス制御装置の他の動作を説明するシーケンス図である。

【図9】 コントローラへ供給される処理タイミング信号を示す説明図である。

【図10】 処理タイミング信号の構成例を示すタイミングチャートである。

【図11】 リクエスト（メッセージ）の構成例である。

【図12】 各コントローラでの処理スケジュール例である。

【図13】 本発明の第3の実施の形態にかかるプロセス制御装置の構成を示すブロック図である。

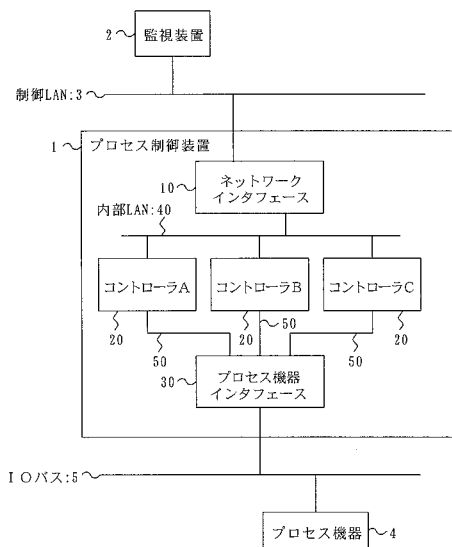
【図14】 コントローラへ供給される処理タイミング信号を示す他の説明図である。

【符号の説明】

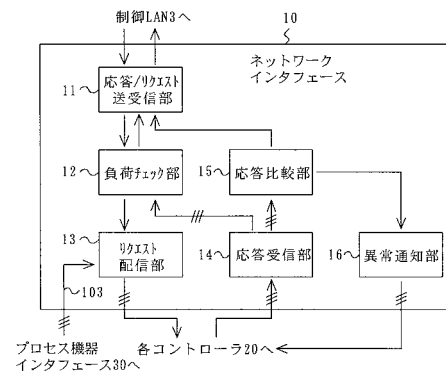
50

1, 6…プロセス制御装置、2…監視装置、3…制御LAN、4…プロセス機器、5…I/Oバス、10…ネットワークインタフェース、11…応答／リクエスト送受信部、12…負荷チェック部、13…リクエスト配信部、14…応答受信部、15…応答比較部、16…異常通知部、20…コントローラ、21…リクエスト処理部、22…データベース、23…演算処理部、24…入出力処理部、25…異常処理部、30…プロセス機器インタフェース、31…コントローラ通信処理部、32…データ比較部、33…I/Oバス入出力処理部、35…処理タイミング信号生成部、40…内部LAN、50…データ通信路、51…タイミング信号バス、60…制御信号線、103…処理タイミング信号。

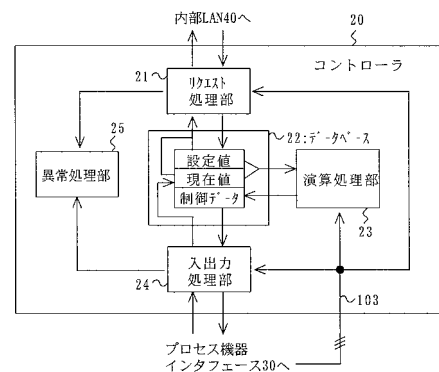
【図 1】



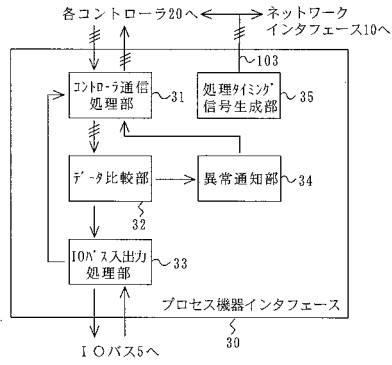
【図 2】



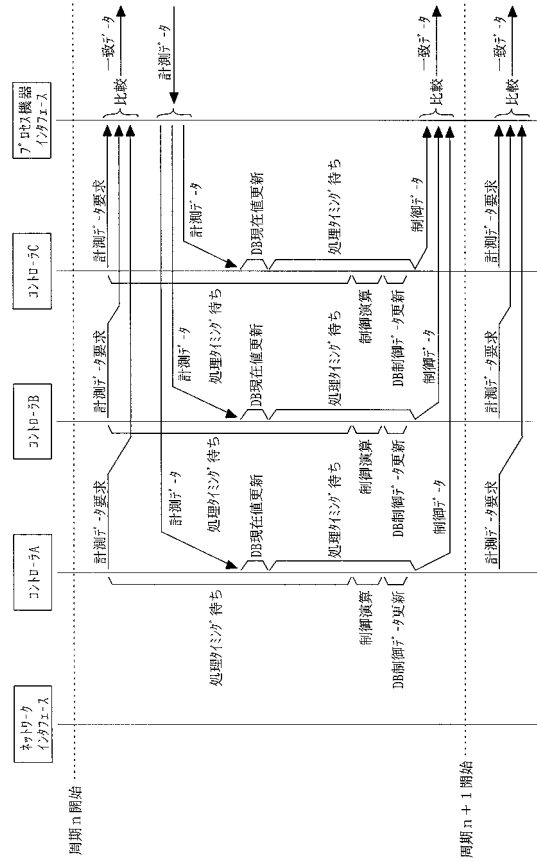
【図 3】



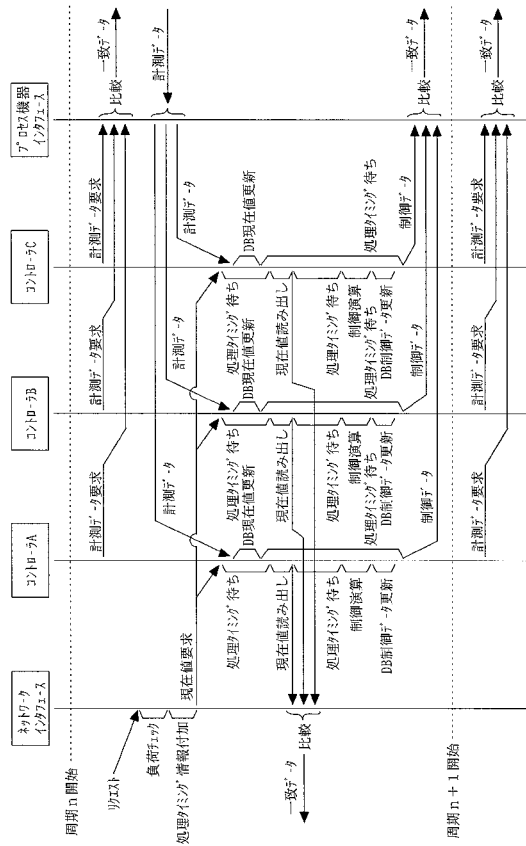
【図 4】



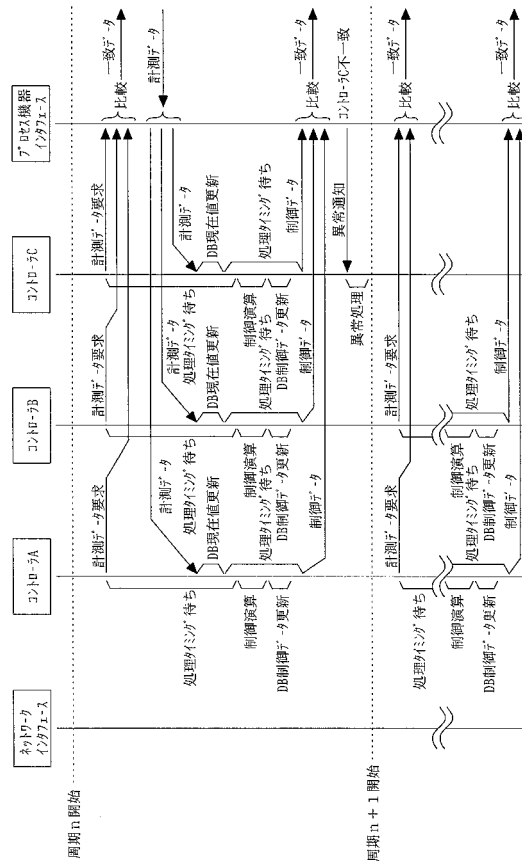
【図 5】



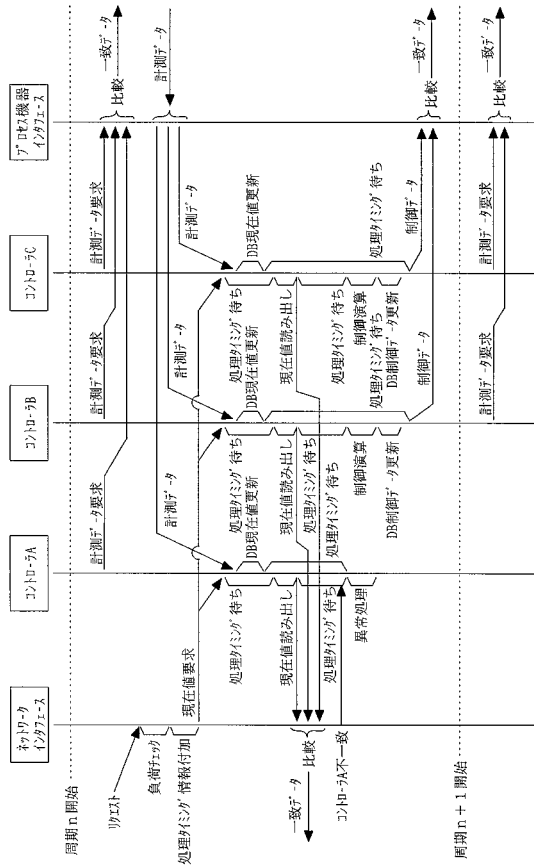
【 図 6 】



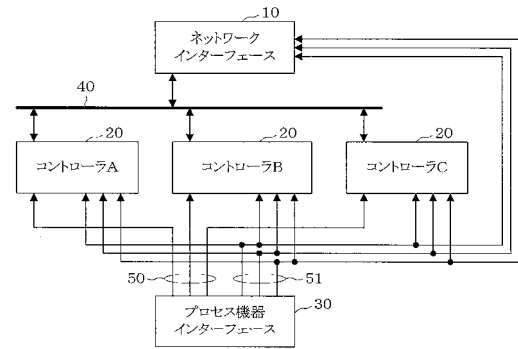
【図 7】



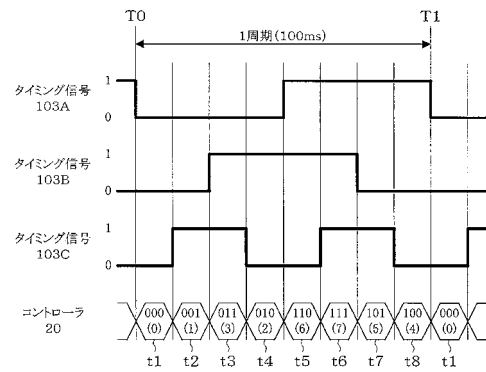
【図 8】



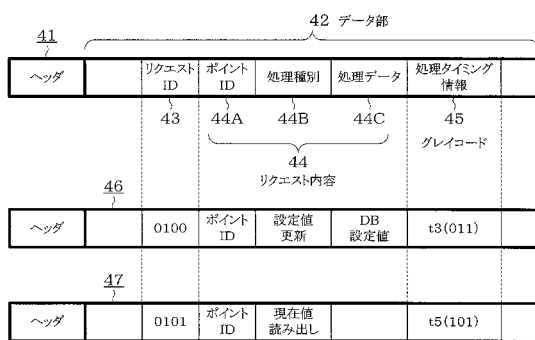
【図 9】



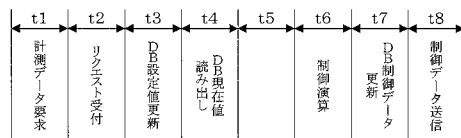
【図 10】



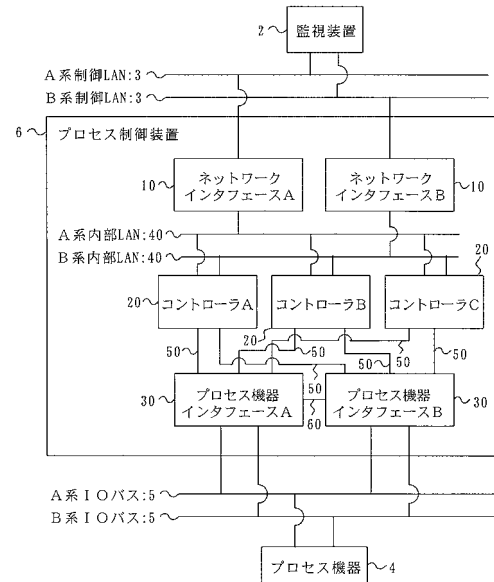
【図 11】



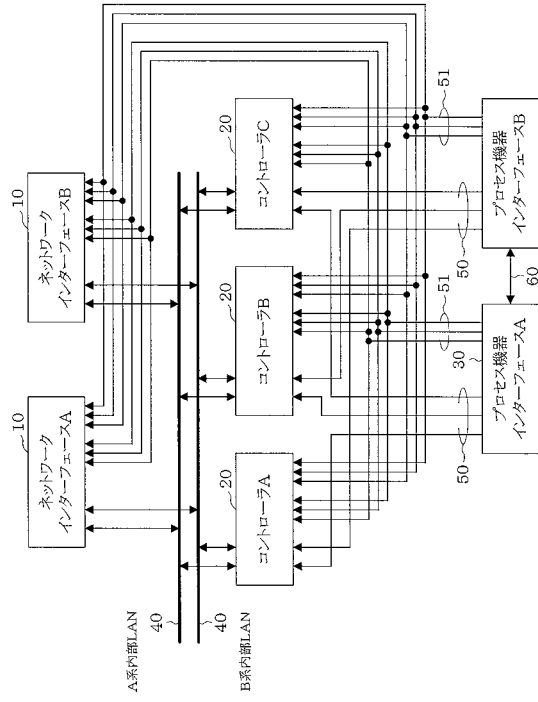
【図 12】



【図 13】



【図 14】



フロントページの続き

(58)調査した分野(Int.Cl., DB名)

G05B 9/03

G06F 11/18