

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4154853号
(P4154853)

(45) 発行日 平成20年9月24日(2008.9.24)

(24) 登録日 平成20年7月18日(2008.7.18)

(51) Int.Cl.	F I	
G05B 19/05 (2006.01)	G05B 19/05	S
G05B 9/03 (2006.01)	G05B 9/03	
G06F 11/20 (2006.01)	G06F 11/20	310C
G06F 13/00 (2006.01)	G06F 13/00	301P
G06F 13/38 (2006.01)	G06F 13/38	340A
請求項の数 13 (全 19 頁) 最終頁に続く		

(21) 出願番号 特願2000-345765 (P2000-345765)
 (22) 出願日 平成12年11月13日(2000.11.13)
 (65) 公開番号 特開2002-149212 (P2002-149212A)
 (43) 公開日 平成14年5月24日(2002.5.24)
 審査請求日 平成17年9月14日(2005.9.14)

(73) 特許権者 503361927
 富士電機機器制御株式会社
 東京都品川区大崎一丁目11番2号
 (74) 代理人 100074099
 弁理士 大菅 義之
 (74) 代理人 100088339
 弁理士 篠部 正治
 (72) 発明者 湯川 真規子
 神奈川県川崎市川崎区田辺新田1番1号
 富士電機株式会社内
 (72) 発明者 鈴木 元治
 神奈川県川崎市川崎区田辺新田1番1号
 富士電機株式会社内

審査官 二階堂 恭弘

最終頁に続く

(54) 【発明の名称】 制御データを等値化する冗長化プログラマブルコントローラ及び等値化方法。

(57) 【特許請求の範囲】

【請求項1】

2重化して冗長性を持たせた1乃至複数組のCPUを持ち、稼働側CPUと待機側CPUとの間でデータの等値化を行うプログラマブルコントローラにおいて、前記稼働側CPUは、等値化対象としているデータを収集する等値化データ収集手段と、前記等値化データ収集手段が収集したデータを等値化フレームによって前記待機側CPUに送信する等値化データ送信手段と、前記等値化データ収集手段による収集処理中に定周期タスクを開始するタイミングになった時、該収集処理完了後前記等値化データ送信手段によるデータの送信開始前に該定周期タスクを実行する定周期タスク実行手段とを備えることを特徴とするプログラマブルコントローラ。

【請求項2】

前記等値化データ送信手段は、前記等値化データ収集手段が収集したデータを複数の等値化フレームに分けて送信することを特徴とする請求項1に記載のプログラマブルコントローラ。

【請求項3】

前記等値化データ送信手段は、プログラマブルコントローラの構成に基づいて前記等値化フレームの大きさを調整することを特徴とする請求項1又は2に記載のプログラマブルコントローラ。

【請求項 4】

前記等値化データ送信手段は、前記等値化フレームのシリアルナンバー及びフレームナンバーを格納し、前記待機側 CPU は、受信した等値化フレームの前記シリアルナンバー及びフレームナンバーを調べて、整合性が確認された時のみ等値化データを分配する等値化データ分配手段を備えることを特徴とする請求項 2 又は 3 に記載のプログラマブルコントローラ。

【請求項 5】

前記整合性をチェックする情報は、どの収集処理による等値化データかを表すフレームナンバーと複数に分割したうちの何番目の等値化フレームかを示すシリアルナンバーであること特徴とする請求項 4 に記載のプログラマブルコントローラ。

10

【請求項 6】

前記待機側 CPU は、前記等値化データ分配手段が分配している間に受信した等値化フレーム内のデータを該分配しているデータの格納領域とは異なるダミー領域に格納するフレーム受信手段を更に備えることを特徴とする請求項 4 又は 5 に記載のプログラマブルコントローラ。

【請求項 7】

前記等値化データ送信手段は、前記等値化データ収集手段が収集したデータを全て送信後、該前記等値化データ収集手段が次の収集処理を完了するまでの間、ダミーフレームを送信することを特徴とする請求項 1 乃至 6 のいずれか 1 に記載のプログラマブルコントローラ。

20

【請求項 8】

前記等値化データ送信手段は、等値化フレームの送信中に送信異常が発生した時、残りの等値化フレームの送信を止めて、前記等値化データ収集手段に次の収集処理の開始を指示することを特徴とする請求項 2 乃至 7 のいずれか 1 に記載のプログラマブルコントローラ。

【請求項 9】

前記等値化データ収集手段は、アプリケーションデータ領域からフレーム送受信作業領域へ等値化データを収集するとき、前記アプリケーションデータ領域のどの位置のデータを前記フレーム送受信作業領域のどの位置に収集するかを記録している転送テーブルを参照して行ない、前記等値化データ分配手段は、前記転送テーブルと同じ転送テーブルを用いてデータの分配を行うことを特徴とする請求項 4 乃至 8 のいずれかに記載のプログラマブルコントローラ。

30

【請求項 10】

前記等値化データ収集手段及び等値化データ分配手段は、等値化対象となるデータの転送元若しくは転送先の領域での配置のされ方に基づいて、データの収集・分配時のデータ転送効率が最適な転送方式の選択することを特徴とする請求項 4 乃至 9 のいずれかに記載のプログラマブルコントローラ。

【請求項 11】

前記等値化データ収集手段は、デフォルトタスクの終了のタイミングに基づいて前記収集処理の開始のタイミングを決定することを特徴とする請求項 1 乃至 10 のいずれかに記載のプログラマブルコントローラ。

40

【請求項 12】

前記等値化データ収集手段は、等値化データ送信手段が等値化フレームを送信中に前記定周期タスクを開始するタイミングになった時、その等値化フレームの送信処理完了後、次の等値化フレームの送信開始前に該定周期タスクを実行することを特徴とする請求項 1 乃至 11 に記載のプログラマブルコントローラ。

【請求項 13】

2 重化して冗長性を持たせた 1 乃至複数組の CPU を持つプログラマブルコントローラにおける、稼働側 CPU と待機側 CPU との間でのデータの等値化方法であって、前記稼働側 CPU は、等値化対象としているデータを収集し、前記収集したデータを格納した等

50

値化フレームを前記待機側CPUに送信し、前記収集中に定周期タスクを開始するタイミングになった時、該収集が完了した後、前記等値化フレームの送信前に該定周期タスクを実行することを特徴とする等値化方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、産業用制御装置であるプログラマブルコントローラにおける入出力制御方式に関し、更に詳しくは制御データの等値化に関する。

【0002】

【従来の技術】

現在、プラント、工場などでは、機械や装置のシーケンス制御に多くのプログラマブルコントローラ（以下PCという）を用い、FA（ファクトリーオートメーション）化を実現している。

【0003】

図16に一般的なマルチCPU構成のPCの例を示す。

1つのPCは、プログラムの実行やデータの演算処理等を行うCPUモジュールや、制御機器とプロセスデータの入出力を行うI/Oモジュール及び不図示の通信モジュールが複数接続されて構成されている。これら各モジュールは、システムバス101を介して互いにデータのやりと合りを行い、接続機器の制御処理を行う。

【0004】

PCが複数のCPUモジュール102を備えるマルチCPU構成の場合、各CPUモジュール102はそれぞれ個別に動作する。よって従来から行われている入出力制御方式では、CPUモジュール102によるI/Oモジュール103などからのプロセスデータの取込みや、I/Oモジュールなどへのプロセスデータの出力を行う入出力処理は、各CPUモジュール102間で非同期に行われる。

【0005】

このため各CPUモジュール102間では同期したデータを使用した処理は行っていない。さらにI/Oモジュール103へのプロセスデータの出力についても、各CPUモジュール102は独自のタイミングで出力しているので、I/Oモジュール103へのプロセスデータの出力は各CPUモジュール102間で同期していない。

【0006】

各I/Oモジュール103は、接続機器からのプロセスデータの出力をリアルタイムで行い、またCPUモジュール102からのデータをリアルタイムで必要とする。よって、複数のCPUモジュール102が連携して処理を行う場合には、CPUモジュール102間やタスク間で同期が行われていないと異なるデータを同じデータと認識して処理してしまうなどの問題が生じる場合がある。

【0007】

上記問題点を考慮し、CPUモジュール間やマルチタスク処理時のタスク間で同期してデータのやり取りを行うことを可能としたものとして、本出願と同一出願人による特開平11-338523号「プログラマブルコントローラにおける入出力制御方式」（以下先願の方式という）がある。

【0008】

先願の方式では、各CPUモジュール102内で、システムバス101の転送速度と同じ速度で実行プロセッサにタクト割込みを一定周期で発生させ、各CPUモジュール102内の実行プロセッサが、このタクト割込みの発生周期（タクト周期）に同期させてI/Oモジュール103に対する入出力処理を行うことによってCPUモジュール102間の同期処理を実現している。

【0009】

【発明の解決しようとする課題】

ところで、システムの信頼性を向上させるためCPUモジュールをデュプレックス構成に

10

20

30

40

50

して冗長性を持たせる場合がある。例えば、図16の構成において、CPUモジュール102-1, 102-3を稼働側とし、CPUモジュール102-2, 102-4をその待機側とする。そして稼働側のCPUモジュール102-1や101-3にエラーが生じると対応する待機側のCPUモジュール102-2や101-4が稼働状態に切り替ってエラーが生じたCPUモジュール102の代わりに処理を務める。

【0010】

このようなデュプレックス構成の場合、切り替り時に処理を引き継げるように、稼働側CPUモジュール102-1, 102-3から待機側のCPUモジュール102-2, 102-4に対して制御データの等値化が行われる。先願の方式の様にPCが特定周期(タクト周期)によりCPUモジュール間やIOMジュールなどとの間で制御データの受け渡しを行い、タクト周期に同期してタスクを実行する構成においては、上記データの等値化の処理を行う場合、以下の問題点がある。

1) 図17は等値化処理のタイミングを示す図である。同図に示すように、先願の方式で等値化処理を行った場合、デフォルトタスクの終了(タスクエンド)のタイミングでシステムの等値化処理を行うことになる。この等値化処理を行っている間、定周期タスクは処理を行えず、CPUが空くのを待つことになる。よって等値化処理にCPUが取られる時間が長いと、定周期タスクはその分CPUが解放されるのを待つこととなり、処理の定周期性が崩される。

2) 1フレーム内に格納されている等値化処理の為のデータが1タクト周期内に処理しきれない大きさのものであった場合、タクト周期が伸びてしまう。従ってタクト周期に同期して実行されている定周期タスクの周期も大きくなってしまう。

3) 等値化処理の為のデータを格納しているフレーム内に、その等値化データがどのタイミングで収集されたデータなのかを示す情報が無いので収集したデータの同期保証が出来ない。

4) 全フレームのデータ収集の同期を保証しなければならないのに、送り出すデータを収集している間に、通常のフレームを送信してしまうと、不定なデータが送られてしまい、データの同期保証ができない。

5) 稼働側CPU用と待機側CPU用とで、等値化データの収集、及びアプリケーションデータ領域への分配にそれぞれ転送テーブルを必要とするが、この転送テーブルを用意するためにメモリ資源を多く消費してしまう。

【0011】

上記問題点を鑑み、本発明は定周期タスクの定周期性を崩さずに、等値化処理を実現する仕組みを持つプログラマブルコントローラ及び等値化方法を提供することを課題とする。

【0012】

また不定な等値化データを待機側に送信したり、無駄なフレームを送信しないようにする仕組みを持つプログラマブルコントローラを提供することを課題とする。

【0013】

更に待機側では、受信データで分配処理が完了する前のデータを破壊したり、不定なデータをアプリケーションデータ領域に分配しない仕組みを持つプログラマブルコントローラを提供することを課題とする。

【0014】

また資源を有効利用し、更に等値化データの収集、分配のパフォーマンスを向上させる仕組みを持つプログラマブルコントローラを提供することを課題とする。更に高速に制御データの等値化処理を行うことが出来るプログラマブルコントローラを提供することを課題とする。

【0015】

【課題を解決するための手段】

上記問題点を解決するため、本発明によるプログラマブルコントローラは、2重化して冗長性を持たせた1乃至複数組のCPUを持ち、稼働側CPUと待機側CPUとの間でデータの等値化を行うことを前提とし、等値化データ収集手段、定周期タスク実行手段及び等

10

20

30

40

50

値化データ送信手段を備える。

【0016】

等値化データ収集手段は、上記稼働側CPUは、等値化対象としているデータを収集する。
等値化データ送信手段は、上記等値化データ収集手段が収集したデータを等値化フレームによって上記待機側CPUに送信する。

【0017】

定周期タスク実行手段は、上記上記等値化データ収集手段による収集処理中に定周期タスクを開始するタイミングになった時、該収集処理完了後上記等値化データ送信手段によるデータの送信開始前に該定周期タスクを実行する。

【0018】

この構成により、定周期タスクは等値化処理が完了するのを待たずに実行することが出来る。

また上記等値化データ送信手段は、上記等値化データ収集手段が収集したデータを複数の等値化フレームに分けて送信する構成とすることが出来る。この構成により等値化データの量が多くてもタクト周期の周期性を壊すことが無い。またこの時各等値化フレームに等値化フレーム内のデータの整合性をチェックする情報を格納し、これを待機側CPUで受信した等値化フレームの上記整合性をチェックする情報を調べて、整合性が確認された時のみ等値化データを分配する構成にすることにより、整合性が取れた等値化データのみが等値化される。

【0019】

更に、上記等値化データ収集手段及び等値化データ分配手段は、等値化対象となるデータの配置に基づいて、データの収集・分配時のデータ転送に最適な転送方式を組合わせて構成することにより、高速な等値化処理を実現できる。

【0020】

【発明の実施の形態】

図1は本実施形態のPCの構成を示す図である。

本実施形態のPCでは、複数のCPUモジュール2、複数のI/Oモジュール3及び通信モジュール4がリング型の伝送トポロジーをもつシステムバス1によって接続されるネットワーク構成をもつ。このシステムバス1に接続されている各モジュールは、そのネットワーク上での通信局としてそれぞれを一意に識別する局番が設定されている。また各CPUモジュール2は、1対1のデュプレックス構成にして冗長性を持たせてあり、図1ではCPU1～CPU4が稼働側、CPU5～CPU8が稼働側それぞれに1対1対応する待機側となっている。

【0021】

モジュール間の通信は、データ送信を行うモジュールが、送信先を局番で指定したフレームをシステムバス1上に送信する。このフレームは、システムバス1上を一定方向に巡回し、システムバス1上の各局は、上流局から送信されてきたフレームが自局宛のものでなければそのまま下流局に送り、自局宛のものあればそれを取込む。これによってシステムバス1上をフレームが一定方向に巡回する。

【0022】

本実施形態では、稼働側CPUから待機側CPUへ送信される等値化データは、マルチキャスト方式によるデータ通信によって転送される。

以下にマルチキャスト方式によるデータ転送について説明する。

【0023】

マルチキャスト方式は、システムバス1上にマルチキャスト通信のフリートークンを巡回させ、データ伝送を要求する局(出力装置に出力するための送信データが用意されており、また当該送信帯域の送信許可設定がされている局)はそのフリートークンを確保して代わりにマルチキャストフレームの送信を行う。マルチキャストフレームはシステムバス1上を巡回し、このマルチキャストフレームを受信したネットワーク上の局は、それを下

10

20

30

40

50

流へ中継して巡回させると共にフレーム内の出力データの設定内容に基づいてデータの取込みを行う。

【0024】

図2は、マルチキャスト方式による通信の説明図である。

本実施形態では、システムバス1の局の内の1つをリングマスター局とし、このリングマスター局によってマルチキャスト方式によるデータ転送を管理する。また図2中のDiはI/Oモジュール3の中で入力装置、Doは出力装置の局を示す。

【0025】

マルチキャスト方式は、稼働側CPUモジュールから待機側CPUモジュールに等値化データを送信する他に、CPUモジュール2から出力機器が接続されているI/Oモジュール3に対してデータの出力を行う時用いられるデータ転送方式である。

10

【0026】

マルチキャスト方式による通信では、まず、システムバス1上に1つだけ存在するネットワーク管理を行うリングマスター局が、データ転送の権限を制御するフリートークンを生成し、システムバス1上にリリースする。このフリートークンのリリースによって、マルチキャスト通信は開始される。

【0027】

リリースされたフリートークンは、各局が順次下流局へ中継してゆき、システムバス1上を巡回する。そしてデータの出力を行いたいCPUモジュール2は、このフリートークンを受信すると、これをシステムバス1から確保し、マスター局となる。

20

【0028】

マスター局は、用意してある送信データを、SD部にこのフレームがマルチキャスト通信のフレームであることを示すデータを設定したマルチキャストフレームとしてシステムバス1上に流す。図2では、マスター局は、保持したフリートークンの代わりにDo1局に対する出力データ(M->Do1)とDo2局に対する出力データ(M->Do2)の2つの局に対する出力データがマルチキャストフレームとして、下流局Do2へ送信される。

【0029】

このマルチキャストフレームを受取った局は、そのフレーム内の出力データに設定されている出力データの格納先の局番を参照し、この出力データが自局に対するものであれば、それを取込むと共にそのフレームをそのまま下流局へ送信する。図2では、マルチキャストフレームを受信したDo2局及びDo1局はそれぞれ自局にあててマスタ局が出力した出力データ(M->Do2、M->Do1)を取込むと共に、その受信フレームをそのまま隣の下流局であるリングマスター局及びマスター局へ送信する。

30

【0030】

フレームが1巡すると、そのフレームを最初に送信したマスタ局は、これをシステムバス1から除去し、代わりにフリートークンを生成して1上にリリースする。

【0031】

そしてフリートークンがシステムバス1を1巡して戻ってくると、リングマスター局はデータ伝送要求の有無をチェックし、要求があれば当該送信データの送信を行う。その後、自局が送信したデータフレームがシステムバス1を1巡して自局に戻ってきたことでリングマスター局はフリートークンをシステムバス1上から除去し、これによってマルチキャスト通信は終了する。

40

【0032】

尚、マルチキャスト通信に限らず、システムバス1上では、リングマスター局のデータフレームは常に最後に送信される。

次に本実施形態におけるシステムでの等値化処理について説明する。

【0033】

図3は、本システムにおける稼働側CPU11と待機側CPU12との間の等値化データの流れを示す図である。同図は、稼働側CPU11とその稼働側CPU11に対応する待

50

機側CPU12の等値化データの送受信に関連する部分のみを抽出して示している。本実施形態では、等値化データはシステムバス1を介してマルチキャスト方式によって稼働側CPU11から待機側CPU12へ送信される。

【0034】

稼働側CPU11では、収集処理として、アプリケーションデータ領域14から等値化が必要なデータのみスキャンしてフレーム送受信作業領域15に収集する。そしてこの収集した等値化データをマルチキャスト方式で待機側CPU12に送信する（以下の説明では等値化データを送信するフレームを等値化フレームという）。この際、等値化データ数が多く、1タクト周期で送信出来ない量の場合、後述する様に複数のフレームに分けて送信することになる。フレーム送受信領域15は、複数のフレーム領域に分かれており、各フレーム領域が1つの送信フレームで送るデータを格納する場所となっている。稼働側CPU11及び待機側CPU12では、イニシャライズ時にシステムを構成するモジュール数等から1タクト周期で送信できる等値化フレームサイズを計算することにより、等値化データをいくつの等値化フレームで送信するかを判断し、その分の領域及びダミー領域（ダミー領域については後述する）をフレーム送受信作業領域15に用意する。

10

【0035】

フレーム送受信領域15内の等値化データは、バスコントローラ16内の送信バッファ17を介してマルチキャストフレームに格納されて、システムバス1上に送信される。送信ポインタ18は、次にフレーム送信するデータが格納されているフレーム領域を指すポインタでデータを送信する度に、次のフレーム領域に動く。

20

【0036】

待機側CPU12では、このマルチキャストフレームを受信し、バスコントローラ16内の受信バッファ19を介してフレーム送受信作業領域15に各受信フレーム毎に格納する。尚この時フレーム送受信領域15の格納位置を受信ポインタ20によってポイントし、受信ポインタ20はフレームを受信する度に次のフレーム領域に動く。フレーム送受信作業領域15に格納された等値化データは、アプリケーションデータ領域14へ古い等値化データを上書きする形で分配され、格納されてゆく。そして待機側CPU12が稼働側に切り替わった際には、アプリケーションプログラムはこのアプリケーションデータ領域14内のデータを用いて稼働側CPU11上のアプリケーションプログラムの処理を引き継ぐ。

30

【0037】

本システムでは、等値化処理を行うタスク（以下等値化タスクという）を、等値化データの収集処理（等値化準備）と等値化データを送信する処理とに分ける。このうち収集処理は、デフォルトタスクのタスク実行処理終了のタイミングで行う。また等値化データ送信処理は、タスク実行とは独立にし、定周期タスクと並列に行わせるようにする。また、等値化データの量が多い場合、等値化データを複数のフレームに分けて送信し、これら各等値化フレームの送信処理はそれぞれ独立させてある。この様に本実施形態では、等値化処理を複数の細かい処理に分けてそれぞれを独立させてある。従って等値化処理が開始された後に定周期タスクを実行させなければならなくなっても、等値化処理が完了するのを待たずに、細かく分けた処理と処理の間に定周期タスクを割込ませ、定周期タスクが完了したら等値化処理に戻る事が出来る。この構成により、等値化データの量が多く等値化処理に多くの時間がかかる場合においても、定周期タスクの定周期性を崩さないで、等値化処理を行う事が出来る。

40

【0038】

図4は、定周期タスク、デフォルトタスク及び等値化タスクによる等値化データの収集・送信のタイミングを示すタイミング図である。同図中の縦線はタクト割込みを示し、このタクト割込みの間が1つのタクト周期となる。また各タスク処理は、I/Oデータの入力処理（網かけ部分）、I/Oデータの出力処理（斜線部分）、計算処理等入出力処理以外のタスク実行処理（無地部分）に分けて示してある。

【0039】

50

図4での「デフォルトタスクエンド有り」、「デフォルトタスク起動禁止」、「デフォルトタスクO禁止」及び「収集要求」は、稼働側CPU11上に設けられた状態フラグの変化を示している。また「送信ポインタ」は、送信ポインタがどのフレーム領域を指しているかを示している。「等値化タスクのステート」は、等値化処理を行うタスクの処理状態を示している。

【0040】

「デフォルトタスクエンド有り」フラグは、等値化対象のデータの収集タイミングを作るフラグで、デフォルトタスクのタスク実行処理が終了するとON(1)になり、等値化対象のデータの収集処理が終了するとOFF(0)となる。

【0041】

「デフォルトタスク起動禁止」フラグは、等値化処理が行われている期間を示すフラグで、デフォルトタスクの実行処理が終了して「デフォルトタスクエンド有り」フラグがONとなった状態で新たなタクト周期に入るとONとなり、全ての等値化データの送信が完了して等値化処理が完了するとOFFとなる。このフラグがONの間、定周期タスクの起動は許可されるが、デフォルトタスクの起動は禁止される。

【0042】

「デフォルトタスクO禁止」フラグは、デフォルトタスクによる出力処理の開始を等値化処理が完了するまで待たせるフラグで、デフォルトタスクのタスク実行処理が終了するとONとなり、最後の等値化フレームの送信を開始するとOFFとなる。このフラグがOFFになってからデフォルトタスクはIOデータ出力処理を開始する。

【0043】

「収集要求」は、等値化処理を行うタスクが収集処理を可能な期間を示すフラグで、「デフォルトタスク起動禁止」フラグと共にONとなり、収集処理が完了するとOFFとなる。

【0044】

図4の41部分に示す様に、デフォルトタスクのタスク実行処理の終了に対応して「デフォルトタスクエンド有り」フラグがONになる。これにより等値化データの収集、送信処理(システムデータ処理)を開始できるようになる。「デフォルトタスクエンド有り」フラグがONになると次のタクト割込みで「収集要求フラグ」がONとなる。定周期タスクのIO入力処理が完了した時、この「収集要求フラグ」の状態を見て、ONとなっていれば等値化データの収集処理を開始する。そして収集処理が完了すると、収集した等値化データの待機側CPUへの送信処理を行う。

【0045】

この等値化データの収集、送信処理を行っている間、「デフォルトタスクO禁止」フラグをONにして、デフォルトタスクの出力処理がこの収集、送信処理に割り込むことを禁止する。しかしこの間、定周期タスクに対しては処理に割込んで起動することを許可する。これにより定周期タスクは定周期性を崩さずに処理を実行することが出来る。

【0046】

また本システムでは、稼働側CPUと待機側CPUとの間で等値化を行うデータの数が多く、待機側CPUに送信する量が多い場合、システムで取り決めた1タクト周期内で送信できる大きさに等値化データを分割し、これらを固定サイズの複数の等値化フレームに分けて格納し、タクト周期に同期して1フレームずつ送信する。図4の42部分ではN個のフレームに分割して等値化データを送信した場合を示している。

【0047】

等値化データを待機側CPUへ送信する等値化フレームの構成例を図5に示す。図5の等値化フレームは、マルチキャスト方式で送信されるフレームで、CPU間マルチキャストフレームヘッダ部とデータ部に分けられる。このうちCPU間マルチキャストフレームヘッダ部には、フレームの種類を知らせるFC、フレームの大きさを示すTS、自CPUモジュールのシステムバス1上のアドレスSA、送信先である待機側CPUモジュールのアドレスDA、等値化フレーム部分の大きさを示すサイズが格納されている。また

10

20

30

40

50

、データ部には等値化データの他に、受信したフレーム内の等値化データの整合性をチェックする為のシリアルNo. 及びフレームNo. が格納されている。このシリアルNo. とフレームNo. については後述する。

【 0 0 4 8 】

この等値化フレームは固定長で、1タクト周期内で送信できる最大のデータサイズからシステムが求めた大きさになっている。

図6は、1タクト周期内に稼働側CPUから送信されるデータを示す図である。1タクト周期内には等値化データの他に、I/Oモジュール3や通信モジュール4等のシステムを構成している他のモジュールへの送信データであるシステム制御データが送信される。このシステム制御データを送信するフレームには、I/Oモジュール3への出力データを含む制御I/O入出力フレーム、モジュールの各種ステータスを含む状態フレーム及び通信モジュールへのメッセージを含むメッセージフレームがある。

10

【 0 0 4 9 】

稼働側CPU及び待機側CPUでは、1タクト周期内に送信できる最大データサイズとこのシステム制御データの大きさを考慮して、1フレームに格納する等値化データの数を可変制御する。稼働側CPU及び待機側CPUは、システムを構成するモジュールの種類や数から、等値化フレームの大きさを決定するが、基本的には、システム構成が大きくなればなるほど1フレームで送信できる等値化データサイズは小さくなり、システム構成が小さければ小さいほど大きくなる。

【 0 0 5 0 】

この等値化フレームに格納できる等値化データの大きさ(等値化データ数)は、システム制御データ数と等値化データ数から式(1)の様に求まる。

1タクト周期内で送信できる最大データサイズ = システム制御データサイズ + 等値化データサイズ · · (1)

式(1)でシステム制御データサイズは、図6の制御I/O入出力フレーム、状態フレーム及びメッセージフレームで送信される制御用のI/Oモジュール3に対する入出力データ、通信モジュール4へのメッセージデータ及び各種ステータスの大きさを加えたものである。よって、等値化データサイズは、式(2)に示すように1タクト周期で転送できる最大のフレームでのデータサイズから、システム制御データサイズを引いたものである。

等値化データサイズ = 1タクト周期内で送信できる最大データサイズ - システム制御データサイズ · · (2)

30

本実施形態では等値化データを複数のフレームに分割して送信するため、これを受信する待機側CPUで受信フレームの整合性をチェック出来るように、等値化データを送信する等値化フレームには、フレームNo. 及びシリアルNo. を挿入してある。

【 0 0 5 1 】

図7は、等値化データを3つのフレームに分けて送信する場合を例とした、稼働側CPUでのフレーム送信時、及び待機側CPUがフレーム受信時のフレーム送受信作業領域15と送信ポインタ18及び受信ポインタ20の状態を示す図である。

【 0 0 5 2 】

稼働側CPUでは、収集処理により収集された等値化データは、フレーム送受信作業領域15の3つのフレーム領域に分けて格納される。そして送信ポインタ18の指すフレーム領域のデータから順次等値化フレームに格納されて送信される。これらの等値化フレームにはフレームNo. 及びシリアルNo. が格納されており、図7のシステムバス領域部分に記載されているx/yが各フレーム内のフレームNo.、Ser.zがシリアルNo. を表している。

40

【 0 0 5 3 】

フレームNo. は、複数に分割された等値化データのうちの何番目かを示す番号で同図ではx/yで表されているもののうちのy部分が分割した数、x部分がそのうちの何番目のデータにあたるかを示しx部分がそのうちの何番目のデータにあたるかを示す。同図の場合3つのフレームに分割されているので、フレームNo. は、1/3 ~ 3/3の値を取る。このフレームNo. は待機側CPUで、等値化フレームが飛ばされずに到達しているかをチェ

50

ックするのに用いる。

【 0 0 5 4 】

シリアルNo. は、そのフレームがどの等値化処理に対する等値化データを含むフレームかを示すもので、稼働側CPUで等値化データの収集処理を行う毎に1, 2, ...と番号がインクリメントされて付せられる。シリアルNo. は、収集したデータの同期を保証するために用いられ、待機側CPUは、同一のシリアルNo. が付せられているフレームは同じ処理処理によって集められた等値化データの一部と認識して処理する。

【 0 0 5 5 】

稼働側CPUでは、収集処理が完了するとフレーム送受信作業領域15内の送信ポイント18が指すフレーム領域内の等値化データをマルチキャスト送信により待機側CPUに送信する。この時、各等値化フレームには、同じシリアルNo. を格納し、また送信順に例えば1, 2, ...とその順番が分るようにフレームNo. を格納する。

10

【 0 0 5 6 】

またアプリケーションデータ領域14から等値化データを収集している間等、全等値化データの送信完了後次の等値化データを送信するまでの間、稼働側CPUは等値化データを含まないダミーフレームを待機側CPUに送信する。このダミーフレームには、シリアルNo. として予め決められた特別な番号が設定されており、フレームを受信した待機側CPUでは、受信フレームのシリアルNo. にこの番号が格納されているとこのフレームがダミーフレームであるとを認識する。そして待機側CPUは、ダミーフレームを受信することにより、稼働側CPUでは1つの収集処理によって集められた等値化データを送信する全ての等値化フレームの送信を完了したと認識する。図7の場合、ダミーフレームにはシリアルNo. として0が格納されている。

20

【 0 0 5 7 】

待機側CPUは、稼働側CPUが複数のファイルに分割して送った等値化データを全て正常に受信した場合にのみ、アプリケーションデータ領域14に等値化データを分配する。この受信した等値化データの整合性のチェックは、フレームNo. 及びシリアルNo. によって行われる。

【 0 0 5 8 】

待機側CPUは、ダミーフレームを受信すると全ての等値化データを受信したと判断して、全等値化フレームのフレームNo. とシリアルNo. から受信した等値化データの整合性を調べる。そして問題なければ、フレーム送受信作業領域15からアプリケーションデータ領域14に等値化データを分配する。

30

【 0 0 5 9 】

待機側CPUは、稼働側CPUが送り出すことになっている等値化フレームのフレームNo. 及びシリアルNo. と、実際に受信した等値化フレーム内のフレームNo. 及びシリアルNo. とを比較する。そして両者が一致すれば、これをアプリケーションデータ領域14に分配し、一致しなければ伝送異常等によりフレームが欠落したなど正常にフレームが届かなかったので、受信データを破棄する。

【 0 0 6 0 】

図7の71部分には両者が一致し、受信データを等値化データとしてアプリケーション領域14に分配する場合を、又同図72部分には両者が一致せず、送られてきた等値化データを破棄してアプリケーションデータ領域14への等値化データの分配は行わない場合を示す。同図71部分の場合、受信した等値化フレームのシリアルNo. が全て1で、フレームNo. が1/3, 2/3, 3/3の順だったので正常の受信として、フレーム送受信作業領域15からアプリケーションデータ領域14に受信した等値化データを分配している。また72部分では受信した等値化フレームにシリアルNo. が1のものと2のものがあるので、これらの受信フレーム内の等値化データはアプリケーションデータ領域14へ分配せずに破棄している。

40

【 0 0 6 1 】

また稼働側CPUは、全フレームのデータ収集の同期を保証するため、等値化データを全

50

て送信してから新たな等値化データの収集処理が完了するまで、ダミーフレームを送信する。収集処理が完了する前に等値化データを送信してしまうと不定なデータが送られてしまうので、稼働側CPUでは、最後の等値化フレームを送信してから次の収集処理が完了するまでの間、送信ポインタ18をダミー領域のアドレスであるダミーアドレスに固定しておき、ダミーフレームを送信する。

【0062】

この時待機側CPUでは、アプリケーションデータ領域14への等値化データの分配処理を行っているが、フレーム送受信作業領域15内の等値化データがダミーフレームのデータで上書きされないように、この分配処理が終るまで、受信ポインタ20をフレーム送受信作業領域15内のダミーアドレスに固定しておく。これにより、受信したダミーフレームは、ダミーアドレスの指す等値化データが未格納な領域に格納され、等値化データが格納されているフレーム送受信作業領域15への上書きがされないようになっている。尚等値化データの分配処理が終了したら、受信フレームがまたダミーフレームであっても受信ポインタ20を通常の等値化フレームの受信位置に戻す。ただこの場合ダミーフレームを受信している間は、受信ポインタ20は移動させない。

10

【0063】

図7の73部分に示すように、稼働側CPUは、アプリケーションデータ領域14からの等値化データの収集処理が完了するまでの間は、ダミーフレームを待機側CPUに送信する。ダミーフレームには予め決めておいた特別な番号である0がシリアルNo.として格納されているので、等値化フレームを受信した待機側CPUでは、シリアルNo.を調べることによってこのフレームがダミーフレームであることを認識する。

20

【0064】

稼働側CPUは、ダミーフレームを送信することにより、待機側CPUに稼働側CPUが等値化データの送信を完了して次の等値化データの送信準備中であることを認識させる。また受信したフレーム内のデータはフレーム送受信作業領域15に上書きして格納するが、ダミーフレームを受信中は、受信ポインタ15を移動させないことにより、受信したダミーフレーム内のデータを同じ領域に上書きして格納する。

【0065】

また稼働CPUは、複数の等値化フレームに分けて等値化データを送信する際、途中の等値化フレーム送信に中送信異常が起きた場合、以降残っているフレームの送信をやめる。そして直ちにアプリケーションデータ領域14をスキャンをして等値化データの再収集処理を行い、等値化処理を継続をする。この様に送信異常が発生してしまった等値化データの送信処理を中止することにより、新たな等値化データの収集処理、及びその等値化データの送信処理に直に取り掛かれる。図7の74部分では、最初の等値化フレーム(フレームNo.が1/3のフレーム)の送信中に送信異常が発生したので、以降のフレーム(フレームNo.が2/3, 3/3フレーム)の送信を取り止め、次の等値化データの収集処理を開始している。尚この等値化データの再収集処理が完了するまでの間、稼働側CPUはダミーフレームを送信する。

30

【0066】

またこの74の部分では、収集処理を開始するタイミングと定周期タスクによる処理が同時期に重なってしまっているが、この様な場合は、定周期タスクのIO入力処理を先に行い、これが完了した後等値化データの収集処理を行い、その後定周期タスクの実行に移る。

40

【0067】

次に稼働側CPUで行われる等値化データの収集処理及び待機側CPUで行われる等値化データの分配処理について説明する。

稼働側CPUで行われるアプリケーションデータ領域14からフレーム送受信作業領域15への等値化データの収集処理及び待機側CPUで行われるフレーム送受信作業領域15からアプリケーションデータ領域14への等値化データの分配処理では、稼働側CPU、待機側CPU共に同じ内容の転送テーブルを用いて行われる。

50

【 0 0 6 8 】

図 8 に稼働側 CPU で行われる等値化データの収集処理、図 9 に待機側 CPU で行われる等値化データの分配処理を示す。図 8 及び図 9 は、等値化データを 2 つの等値化フレームに分けて送信する場合を例としており、フレーム送受信作業領域 1 5 には、送受信する 2 つのフレームの CPU 間マルチキャストフレームヘッダ部内のデータを格納しているヘッダ部及びデータ部のデータを格納しているデータ部が示されている。

【 0 0 6 9 】

図 8 に示す稼働側 CPU で行われる等値化データの収集処理の場合、アプリケーションデータ領域 1 4 からフレーム送受信作業領域 1 5 へ等値化データを収集するが、この処理は、アプリケーションデータ領域 1 4 のどの位置のデータをフレーム送受信作業領域 1 5 のどの位置に収集するかを記録している転送テーブル 3 0 を参照して行われる。

10

【 0 0 7 0 】

この転送テーブル 3 0 は、稼働側 CPU 及び待機側 CPU において、CPU のイニシャライズ時に、CPU モジュール上で稼働されるユーザが作成したアプリケーションプログラムから等値化対象のデータを認識して作成される。稼働側 CPU とそれに対応する待機側 CPU では、同じアプリケーションプログラムがインストールされているため、稼働側 CPU と待機側 CPU では等値化対象データも同じ、更にその転送方式も同じになる。よって稼働側 CPU 及び待機側 CPU では、転送テーブル 3 0 として全く同じものが作成されることになる。稼働側 CPU 及び待機側 CPU では、稼働 / 待機の切替えに備え、等値化データの収集用の転送テーブルとアプリケーションデータ領域 1 4 への分配用転送テーブルの両方を実装するが、本実施形態では同じ転送テーブルを用いて収集処理と分配処理を行えるので、メモリ資源を有効に用いることが出来る。

20

【 0 0 7 1 】

転送テーブル 3 0 にはいくつかのフレームに分けて等値化データを送信するかを示すブロック数 N、後述するアプリケーションデータ領域 1 4 とフレーム送受信作業領域 1 5 の間のデータの転送方式、及びブロック数 N 分のフレーム送受信作業領域 1 5 側のアドレスを示すフレーム側アドレス、データの個数及びアプリケーションデータ領域側アドレスが記録されており、その構成は転送方式によって異なる。また複数の転送方式を組合わせて行う場合（例えば図 8 に於て、D 1 ~ D 5 をブロック転送、D 1 0 0 ~ D 1 0 4 を構造体転送で行う場合等）には、転送テーブル 3 0 には、各転送方式毎にこれらの情報が記録されている。稼働側 CPU で行われる等値化データの収集処理では、この転送テーブル 3 0 のアプリケーションデータ領域側アドレスを転送元アドレス、フレーム側アドレスが転送先アドレスとして等値化データを収集する。

30

【 0 0 7 2 】

稼働側 CPU で収集された等値化データは、等値化フレームに格納されて待機側 CPU に送信され、待機側 CPU では、受信した等値化フレームをフレーム送受信作業領域 1 5 に格納し、この中のデータ部内の等値化データを図 9 に示すように転送テーブル 3 0 を用いてアプリケーションデータ領域 1 4 に分配する。

【 0 0 7 3 】

この分配処理では稼働側 CPU でこの等値化データの収集処理に用いられた転送テーブル 3 0 と同じ転送テーブル 3 0 を、図 8 の収集処理の時とは逆に、フレーム側アドレスを送信元、アプリケーションデータ領域側アドレスを送信先としてデータの転送を行う。このように同じ転送テーブル 3 0 を、稼働側 CPU と待機側 CPU とで転送元アドレスと転送先アドレスを逆に用いることによりメモリ資源の節約をすることが出来る。

40

【 0 0 7 4 】

本実施形態では、アプリケーションデータ領域 1 4 とフレーム送受信作業領域 1 5 との間で行われる等値化データの転送方式として、ブロック転送、ワード転送及び構造体転送の 3 つの転送方式を持つ。また各転送方式によって用いられる転送テーブル 3 0 は、転送方式毎にアプリケーションデータ領域側アドレス部分の構成が異なっている。稼働側 CPU 及び待機側 CPU では、イニシャライズ時に等値化対象とするデータのアプリケーション

50

データ領域 14 の配置位置からデータの収集・分配時のデータ転送に最適な転送方式を選択し、或いは必要に応じて複数の転送方式を組合わせて対応する転送テーブル 30 を作成する。

【 0075 】

図 10 に、ブロック転送方式によって用いられる転送テーブルの構成例を示す。尚同図及び図 11 及び図 12 は、ブロック数 $N = 1$ の時のものでブロック数 N が 2 以上の時は、転送テーブル 30 はフレーム側アドレス、個数及びアプリケーションデータ領域側アドレスに対応する部分をブロック数分持つことになる。

【 0076 】

ブロック転送方式は、アプリケーションデータ領域 14 の一定の領域の全てのデータを等値化する時に行う転送方式で、ブロック単位でデータ転送を行う。

図 10 の転送テーブルには、アプリケーションデータ領域側アドレスとして個数部分に記録されている組分のワード数とアドレスが組となって記録されている。

【 0077 】

ブロック転送の際は、転送テーブルに格納されているアドレスからワード数分のデータをブロック単位で転送する。例えば図 8 の等値化データの収集処理が、ブロック転送によるものである時、図 10 のアドレス 1 にはアプリケーションデータ領域 14 の D1 の先頭アドレス、ワード数 1 には D1 の総ワード数が格納され、アドレス 2 にはアプリケーションデータ領域 14 の D2 の先頭アドレス、ワード数 2 には D2 の総ワード数が格納されている。等値化データの収集処理時には、D1 としてアドレス 1 からデータをワード数 1 分読み出し、フレーム送受信作業領域 15 のフレーム側アドレスに格納されているアドレス部分に格納する。次に D2 としてアドレス 2 からデータをワード数 2 分読み出し、フレーム送受信作業領域 15 の D1 を格納した続きに格納する。以降、転送テーブル 30 のアプリケーションデータ領域側アドレス部分に記録されている全アドレス及びワード数に対して同様の処理を行う。

【 0078 】

また等値化データの分配処理では、D1 としてフレーム送受信作業領域 15 のフレーム側アドレスに記憶されている位置からワード数 1 分データを読み出し、これをアプリケーションデータ領域 14 のアドレス 1 部分に格納し、次に D2 としてフレーム送受信作業領域 15 の D1 を読み出した続きからワード数 2 分のデータを読み出し、これをアプリケーションデータ領域 14 のアドレス 2 部分に格納する。以降全データに対して同様の処理を行うことで、転送テーブル 30 を用いた等値化データのアプリケーションデータ領域 14 への分配が行える。

【 0079 】

次にワード転送方式について説明する。

ワード転送方式は、1ワード単位で独立に制御して等値化する時に使用する転送方式で、転送テーブル 30 で転送するデータを 1ワード単位で指定する。

【 0080 】

図 11 に、ワード転送方式によって用いられる転送テーブルの構成例を示す。図 11 の転送テーブルには、アプリケーションデータ領域側アドレス部分に個数部分に記録されている数のアドレスが記録されている。

【 0081 】

等値化データの収集処理をワード転送で行う際は、アプリケーションデータ領域 14 のアプリケーションデータ領域側アドレス部分に格納されている各アドレスから 1ワードずつ読み出し、フレーム送受信作業領域 15 のフレーム側アドレスに格納されている部分に順次転送する。また等値化データのアプリケーションデータ領域 14 への分配処理時には、フレーム送受信作業領域 15 のフレーム側アドレス部分に格納されたアドレスから順次 1ワードずつ読み出し、アプリケーションデータ領域 14 のアプリケーションデータ領域側アドレス部分に格納されているアドレスの位置に転送して行く。

【 0082 】

10

20

30

40

50

次に構造体転送について説明する。

構造体転送は、変則的な位置のデータを指定して転送する場合に用いる転送方式で3種類のモードが有る。

【0083】

図12は、構造体転送に用いられる転送テーブルの構成例である。

図12の転送テーブルには、アプリケーションデータ領域側アドレス部分に個数部分にアドレスが1つ記録されている。また転送方式部分には、どのモードによる構造体転送かを示すモード番号0~2が記録されている。

【0084】

図13、図14及び図15はモード1, 2および3での構造体の例を示す図である。各図中の(R)の印のワードの所を等値化対象としてアプリケーションデータ領域14から読み出し、(M)部分を等値対象としないことを表している。

10

【0085】

モード番号0のモード1の場合は、図13に示すように2ワード構造のうち2ワード共等値化対象とする。このモード1の構造体転送において、例えば個数に3が設定されていたならば、収集処理においてアプリケーションデータ領域側アドレスのアドレス部分から連なる $2 \times 3 = 6$ ワードのデータを読み出して収集する。又分配処理の時は、上記アドレスに記録されているアドレス値の部分にフレーム送受信作業領域15からのフレーム側アドレス部分から読み出した6ワードのデータを格納する。

【0086】

20

モード番号1のモード2の場合は、図14に示す様に4ワード構造のうち3・4ワード目を等値化対象としてデータ転送を行う。このモード2の構造体転送において、例えば個数に3が設定されていたならば、収集処理においてアプリケーションデータ領域14の指定アドレス部分から3・4ワード目、7・8ワード目及び11・12ワード目のデータを等値化データとして読み出して収集する。また分配処理時には、フレーム送受信作業領域15の転送テーブルによって指定された位置から読み出した6ワードのデータをアプリケーションデータ領域14の指定アドレスから3・4ワード目、7・8ワード目及び11・12ワード目の位置に格納して行く。

【0087】

モード番号2のモード3場合は、図15に示すように8ワード構造のうち、1・2・5・6ワード目を等値化する。例えば、個数に2が設定されていたならば、収集処理においてアプリケーションデータ領域14の指定アドレス部分から1・2・5・6ワード目及び7・8・11・12ワード目のデータを等値化データとして読み出して収集し、分配処理時には、フレーム送受信作業領域15の転送テーブルによって指定された位置から読み出した6ワードのデータをアプリケーションデータ領域14の指定アドレスから1・2・5・6ワード目及び7・8・11・12ワード目の位置に格納して行く。

30

【0088】

この様に、本実施形態では、等値化対象のデータのアプリケーションデータ領域14での配置によって最適な転送方式を選択したり、或いは複数の転送方式を組み合わせることが出来るので、スピーディに収集・分配処理を行うことが可能となり、高速に制御データの等値化処理を行うことが出来る。

40

【0089】

【発明の効果】

以上説明のように、本発明によれば、稼働側CPUで行われている定周期タスクの定周期性を崩すことなく、等値化処理を実行することが出来る。

【0090】

また、複数の等値化フレームに分けて等値化データを送信することにより、等値化データの量が多くてもタクト周期の周期性を壊すことが無い。

更に稼働側CPUでは、収集処理中には、ダミーフレームを送信することにより収集が完了する前のデータが送信されることが無い。

50

【 0 0 9 1 】

また複数の等値化フレームの送信中に送信異常が発生してしまった時、以降の等値化データの送信処理を中止することにより、新たな等値化データの収集処理、及びその等値化データの送信処理に直ちに取り掛かれる。

【 0 0 9 2 】

更に待機側CPUでは、受信した等値化データの整合性のチェックを行うことが出来る。また、等値化データの分配処理中に受信したフレームのデータで分配処理中のデータが上書きされることが無い。

【 0 0 9 3 】

更に収集処理と分配処理で転送テーブルを共用することにより、メモリ資源の有効化を図れる。

10

また等値化対象となるデータの配置に基づいて、データの収集・分配時のデータ転送に最適な転送方式を選択し、必要に応じて組み合わせることにより、高速な等値化処理を実現できる。

【 図面の簡単な説明 】

【 図 1 】 本実施形態のPCの構成を示す図である。

【 図 2 】 マルチキャスト方式による通信の説明図である。

【 図 3 】 稼働側CPUと待機側CPUとの間の等値化データの流れを示す図である。

【 図 4 】 定周期タスク、デフォルトタスク及び等値化タスクによる等値化データの収集・送信のタイミングを示すタイミング図である。

20

【 図 5 】 等値化フレームの構成例を示す図である。

【 図 6 】 1タクト周期内に稼働側CPUから送信されるデータを示す図である。

【 図 7 】 稼働側CPUでのフレーム送信時及び待機側CPUがフレーム受信時のフレーム送受信作業領域と送信/受信ポイントの状態を示す図である。

【 図 8 】 稼働側CPUで行われる等値化データの収集処理を示す図である。

【 図 9 】 待機側CPUで行われる等値化データの分配処理を示す図である。

【 図 1 0 】 ブロック転送方式によって用いられる転送テーブルの構成例を示す図である。

【 図 1 1 】 ワード転送方式によって用いられる転送テーブルの構成例を示す図である。

【 図 1 2 】 構造体転送方式によって用いられる転送テーブルの構成例を示す図である。

【 図 1 3 】 モード1の構造体例を示す図である。

30

【 図 1 4 】 モード2の構造体例を示す図である。

【 図 1 5 】 モード3の構造体例を示す図である。

【 図 1 6 】 一般的なマルチCPU構成のPCの例を示す図である。

【 図 1 7 】 等値化処理のタイミングを示す図である。

【 符号の説明 】

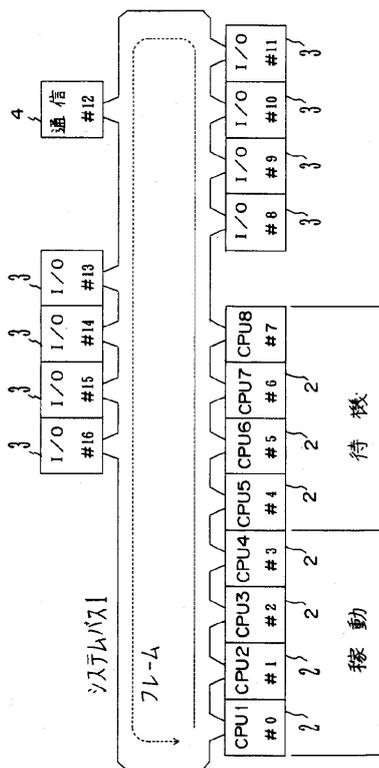
- 1、101 システムバス
- 2、102 CPUモジュール
- 3、103 I/Oモジュール
- 4 通信モジュール
 - 11 稼働側CPU
 - 12 待機側CPU
 - 13 実行プロセッサ
 - 14 アプリケーションデータ領域
 - 15 フレーム送受信作業領域
 - 16 バスコントローラ
 - 17 送信バッファ
 - 18 送信ポイント
 - 19 受信バッファ
 - 20 受信ポイント
 - 30 転送テーブル

40

50

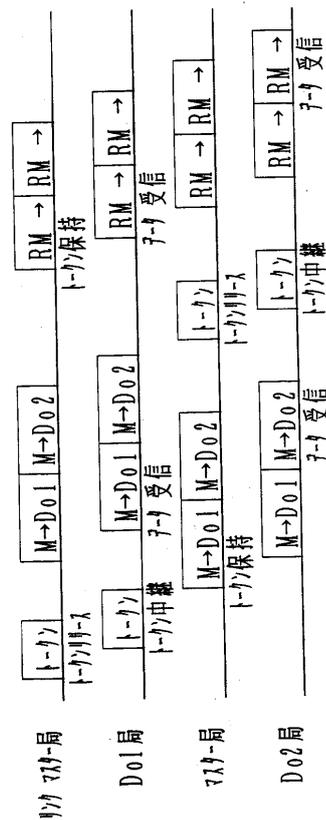
【図1】

本実施形態のPCの構成を示す図



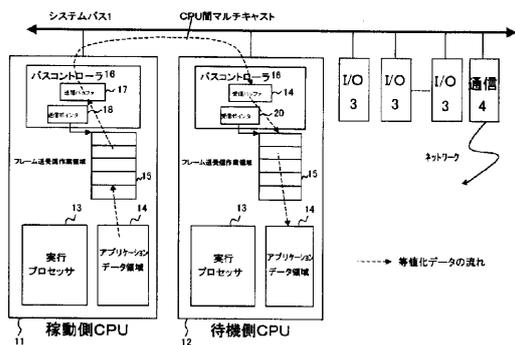
【図2】

マルチキャスト方式による通信の説明図



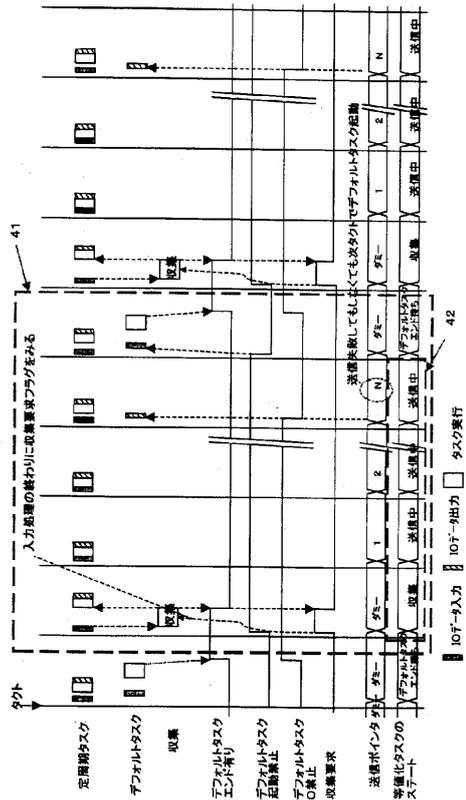
【図3】

稼動側CPUと待機側CPUとの間の等値化データの流れを示す図



【図4】

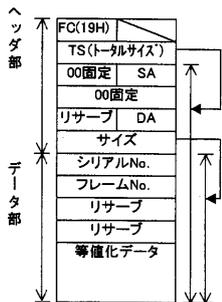
等値化データの収集・送信のタイミングを示す図



10千-入力 10千-出力 □ タスク実行

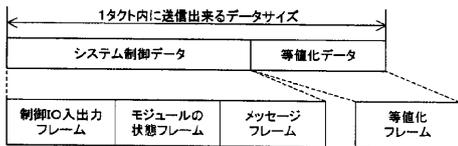
【図5】

等値化フレームの構成例を示す図



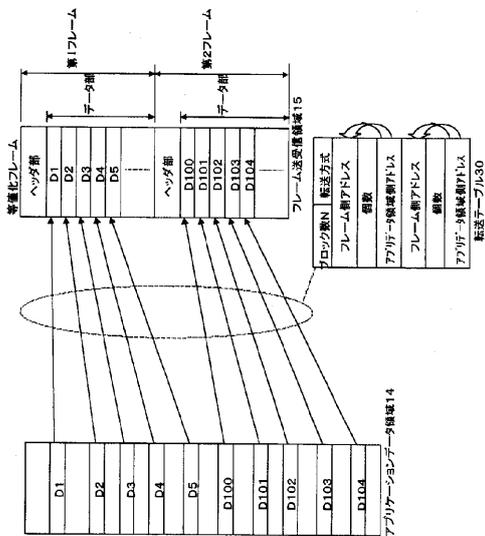
【図6】

1タクト周期内に稼働側CPUから送信されるデータを示す図



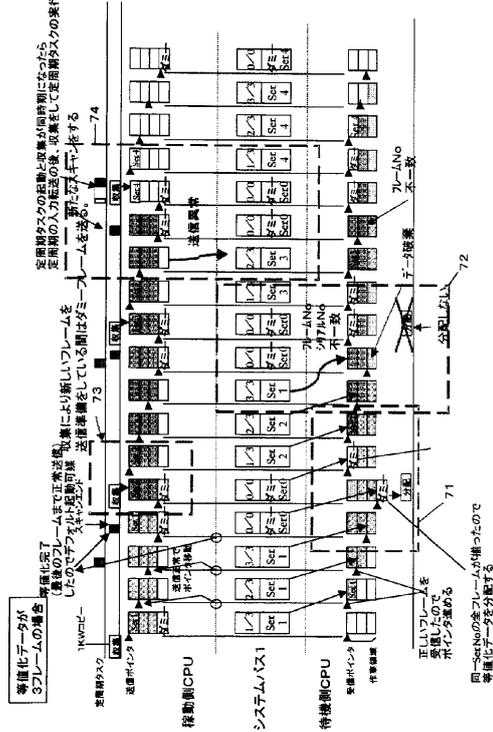
【図8】

稼働側CPUで行われる等値化データの収集処理を示す図



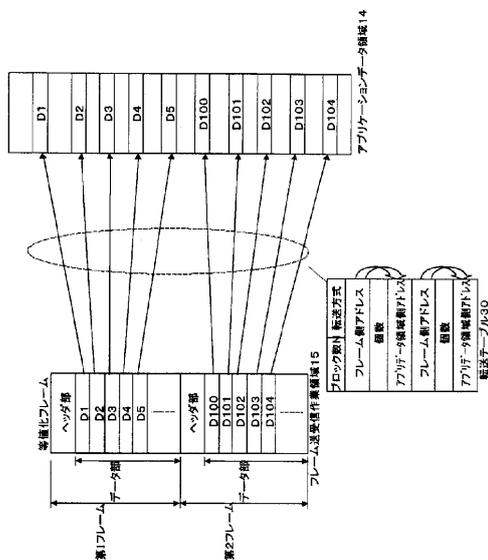
【図7】

稼働側CPUでのフレーム送信時及び待機側CPUがフレーム受信時の状態を示す図



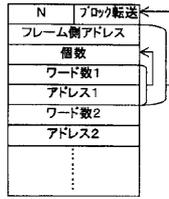
【図9】

待機側CPUで行われる等値化データの分配処理を示す図



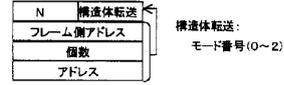
【図10】

ブロック転送方式によって用いられる転送テーブルの構成例を示す図



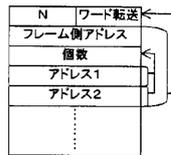
【図12】

構造体転送方式によって用いられる転送テーブルの構成例を示す図



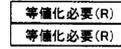
【図11】

ワード転送方式によって用いられる転送テーブルの構成例を示す図



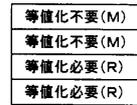
【図13】

モード1の構造体例を示す図



【図14】

モード2の構造体例を示す図



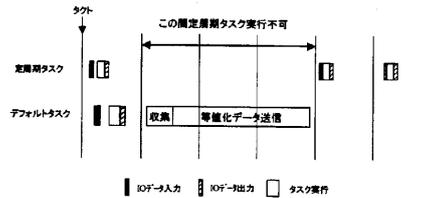
【図15】

モード3の構造体例を示す図



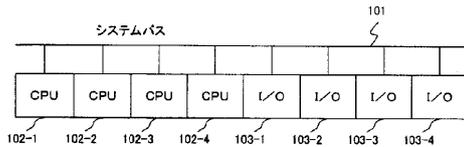
【図17】

等値化処理のタイミングを示す図



【図16】

一般的なマルチCPU構成のPCの例を示す図



フロントページの続き

(51) Int.Cl. F I
G 0 6 F 13/42 (2006.01) G 0 6 F 13/42 3 5 0 A

(56) 参考文献 特開平 0 9 - 0 6 2 3 0 4 (J P , A)
特開平 1 1 - 3 3 8 5 2 3 (J P , A)
特開平 0 8 - 0 0 6 6 1 9 (J P , A)
特開昭 5 6 - 0 3 8 5 4 2 (J P , A)
特開平 0 8 - 2 7 2 6 3 6 (J P , A)

(58) 調査した分野(Int.Cl. , D B 名)

G05B 19/04-19/05
G05B 9/03- 9/05
G06F 11/16-11/20
G06F 13/00
G06F 13/38-13/42