

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5908068号
(P5908068)

(45) 発行日 平成28年4月26日(2016.4.26)

(24) 登録日 平成28年4月1日(2016.4.1)

(51) Int. Cl. F I
G05B 9/03 (2006.01) G05B 9/03
G06F 11/20 (2006.01) G06F 11/20 310C

請求項の数 9 (全 22 頁)

(21) 出願番号 特願2014-514305 (P2014-514305)
 (86) (22) 出願日 平成24年5月10日 (2012.5.10)
 (86) 国際出願番号 PCT/JP2012/062003
 (87) 国際公開番号 W02013/168258
 (87) 国際公開日 平成25年11月14日 (2013.11.14)
 審査請求日 平成26年9月8日 (2014.9.8)

(73) 特許権者 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 100094916
 弁理士 村上 啓吾
 (74) 代理人 100073759
 弁理士 大岩 増雄
 (74) 代理人 100127672
 弁理士 吉澤 憲治
 (74) 代理人 100088199
 弁理士 竹中 考生
 (72) 発明者 野村 明裕
 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】 待機冗長二重化装置

(57) 【特許請求の範囲】

【請求項1】

2つのコントローラで構成し、一の前記コントローラが稼働側コントローラとして動作する時は、他方のコントローラは待機側コントローラとして動作する待機冗長二重化装置において、

各前記コントローラは、それぞれ内部に複数のCPUと、

メモリと、

I/Oボードと、

複数の前記CPUと、前記メモリと、前記I/Oボードとの間でデータを送受信するI/Oバスコントローラと、

前記稼働側コントローラから等値化データバスを介して前記待機側コントローラに、前記稼働側コントローラと前記待機側コントローラの装置状態を等値化するために使用する等値化データを送信する、前記I/Oバスコントローラから独立した等値化バスコントローラとを備え、

前記等値化データは、各演算ルーチンの識別コードと版コードを有し、

前記待機側コントローラの各前記CPUは、前記識別コードと前記版コードを確認して各前記CPUに割り当てられた演算を実行する待機冗長二重化装置。

【請求項2】

各前記コントローラは、

タイマと、

前記タイマに連動して一定周期で前記CPUに処理の開始を通知をする起動制御部とを有し、

前記稼働側コントローラの前記起動制御部から通知を受けた、前記稼働側コントローラの一の前記CPUは、前記稼働側コントローラの前記CPUが演算に使用するデータを、前記I/Oボードから取得して前記メモリに、前記I/Oバスコントローラを介して入力した後、

前記稼働側コントローラの前記CPUに割り当てられた演算の実行を指示し、全ての前記CPUの演算が終了した後、全ての前記CPUの演算結果を纏めて、前記待機側コントローラに送信すべき前記等値化データを構成する請求項1に記載の待機冗長二重化装置。

10

【請求項3】

前記メモリ内には、

前記I/Oボードから入力されたデータを纏めて記録する入力データと、

いずれかの前記CPUによって実行される演算ルーチンを記録する演算ルーチンと、

個別の前記演算ルーチンで使用する、前記入力データと、前記入力データを利用して前記演算ルーチンを実行して得た演算結果と、過去の前記演算結果とを記録する演算データと

、前記I/Oボードに出力するデータを纏めて保存する出力データと、

前記コントローラが前記稼働側コントローラとして動作する時に、前記待機側コントローラに送信する前記等値化データを纏めて記録する等値化データ送信バッファと、

20

前記コントローラが前記待機側コントローラとして動作する時に、前記稼働側コントローラからの前記等値化データを交互に受信する2つの等値化データ受信バッファとを有する請求項2に記載の待機冗長二重化装置。

【請求項4】

前記メモリは、各前記演算ルーチンの属性データを有し、

前記演算ルーチンの前記属性データには、

各前記演算ルーチンを実行するCPU番号を特定するCPU情報と、

各前記演算ルーチンを前記タイマの周期の倍数の周期で起動させる起動周期情報と、

各前記演算ルーチンを前記タイマの周期の倍数に対して所定のオフセットを持たせて起動させるオフセット情報と、

30

各前記演算ルーチンを実行した回数を示す実行回数情報と、

前記等値化データを前記待機側コントローラに送信した回数を示す送信回数情報と、

各前記演算ルーチンが前記タイマの周期内に終了しなかった場合における次の処理方法を示す継続処理可否情報、の中の少なくとも一つの情報を有する請求項3に記載の待機冗長二重化装置。

【請求項5】

前記稼働側コントローラの前記メモリに記録した前記属性データには、前記実行回数情報と、前記送信回数情報を有し、前記実行回数情報と、前記送信回数情報の内容を比較して、前記等値化データのデータ構成を決定する請求項4に記載の待機冗長二重化装置。

【請求項6】

40

前記等値化データは、前記送信回数情報を有し、

前記待機側コントローラは、2つの前記等値化データ受信バッファの中の前記等値化データの内、前記送信回数情報の進んでいる方の前記等値化データを使用して各前記CPUに割り当てられた演算を実行する請求項4に記載の待機冗長二重化装置。

【請求項7】

前記メモリは、

各前記CPUがそれぞれ固有に扱うデータを記録するCPU用メモリと、

複数の前記CPUが、共通して使用するデータを記録する共通メモリとからなる、

請求項4に記載の待機冗長二重化装置。

【請求項8】

50

前記等値化バスコントローラと前記等値化データバスを冗長化構成とした請求項 1 に記載の待機冗長二重化装置。

【請求項 9】

前記一の CPU は、割込制御、又は前記メモリに設けたフラグの状態を識別して各前記 CPU に、各前記 CPU に割り当てられた演算の開始を通知する請求項 2 から請求項 8 のいずれか 1 項に記載の待機冗長二重化装置。

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、一定周期で、CPU 毎に異なる演算を行うマルチ CPU を備えたコントローラを 2 重化した、待機冗長二重化装置に関するものである。 10

【背景技術】

【0002】

従来、マルチ CPU を用いたデータの等値化コントローラとして、それぞれの CPU がシステムバスコントローラを持ち、各 CPU が、システムバスコントローラを介してシステムバスに接続される構成を有するものとして、特許文献 1 に示すような待機冗長二重化装置が提案されている。

この装置では、システムバスコントローラを使用して、稼働側 CPU から待機側 CPU に等値化データを送信することで、2 つのコントローラ間でのデータの等値化を実現している。 20

【先行技術文献】

【特許文献】

【0003】

【特許文献 1】特開 2002 - 149212 号公報（段落 0020 - 0022、図 1、図 3）

【発明の概要】

【発明が解決しようとする課題】

【0004】

一定周期で異なる演算を行うマルチ CPU を有する待機冗長二重化装置では、稼働側 CPU から待機側 CPU に等値化すべきデータを転送するためのバスが存在するが、特許文献 1 に示す等値化コントローラでは、それぞれの CPU にデータの等値化を行うためのバスコントローラが存在しているため、CPU の数が増えると、その数と同一のバスコントローラが必要となる。 30

これにより、回路規模が増大するという課題があった。

また、等値化データの送受信と I/O ボードとのデータの送受信を、同一のシステムバスを使用する構成とするために、等値化データの送受信と I/O ボードの送受信を同時に転送できず、データ転送のレスポンスを向上できないという課題があった。

【0005】

この発明は、上記の課題を解決するためになされたものであり、CPU の数が増えても、バスコントローラの数が増えず、回路規模を抑えることが可能であり、また、等値化データのコントローラ間での送受信と、I/O ボードと CPU とのデータの送受信の同時実行を可能とする待機冗長二重化装置を提供することを目的とする。 40

【課題を解決するための手段】

【0006】

本発明に係る待機冗長二重化装置は、
2 つのコントローラで構成し、一の前記コントローラが稼働側コントローラとして動作する時は、他方のコントローラは待機側コントローラとして動作する待機冗長二重化装置において、
各前記コントローラは、それぞれ内部に複数の CPU と、
メモリと、 50

I/Oボードと、
 複数の前記CPUと、前記メモリと、前記I/Oボードとの間でデータを送受信するI/Oバスコントローラと、
 前記稼働側コントローラから等値化データバスを介して前記待機側コントローラに、前記稼働側コントローラと前記待機側コントローラの装置状態を等値化するために使用する等値化データを送信する、前記I/Oバスコントローラから独立した等値化バスコントローラとを備え、
前記等値化データは、各演算ルーチンの識別コードと版コードを有し、
前記待機側コントローラの各前記CPUは、前記識別コードと前記版コードを確認して各前記CPUに割り当てられた演算を実行するものである。

10

【発明の効果】

【0007】

本発明に係る待機冗長二重化装置は、
 各前記コントローラは、それぞれ内部に複数のCPUと、
 メモリと、
 I/Oボードと、
 複数の前記CPUと、前記メモリと、前記I/Oボードとの間でデータを送受信するI/Oバスコントローラと、
 前記稼働側コントローラから等値化データバスを介して前記待機側コントローラに、前記稼働側コントローラと前記待機側コントローラの装置状態を等値化するために使用する等値化データを送信する、前記I/Oバスコントローラから独立した等値化バスコントローラとを備え、
前記等値化データは、各演算ルーチンの識別コードと版コードを有し、
前記待機側コントローラの各前記CPUは、前記識別コードと前記版コードを確認して各前記CPUに割り当てられた演算を実行するもので、

20

複数のCPUが処理するデータの等値化処理において、CPU毎のバスコントローラを不要とすることができる。

また、等値化データバスとI/Oバスを独立のバスとし、それぞれのバスコントローラも独立しているので、等値化データの送受信とI/Oボードの送受信の同時送信が可能となり、処理の高速化を図ることができる。

30

【図面の簡単な説明】

【0008】

【図1】本発明の実施の形態1に係る待機冗長二重化装置の構成を示す図である。

【図2】本発明の実施の形態1に係る待機冗長二重化装置の割込制御の構成を示す図である。

【図3】本発明の実施の形態1に係る待機冗長二重化装置のメモリの構成を示す図である。

【図4】本発明の実施の形態1に係る待機冗長二重化装置の等値化データ送信データの構成を示す図である。

【図5】本発明の実施の形態1に係る待機冗長二重化装置の稼働側コントローラのCPU a0の動作を示すフローチャートである。

40

【図6】本発明の実施の形態1に係る待機冗長二重化装置の稼働側コントローラのCPU a1の動作を示すフローチャートである。

【図7】本発明の実施の形態1に係る待機冗長二重化装置の待機側コントローラのCPU b0の動作を示すフローチャートである。

【図8】本発明の実施の形態1に係る待機冗長二重化装置の待機側コントローラのCPU b1の動作を示すフローチャートである。

【図9】本発明の実施の形態1に係る待機冗長二重化装置の動作を示すタイミング図である。

【図10】本発明の実施の形態1に係る待機冗長二重化装置の動作を示すタイミング図で

50

ある。

【図 1 1】本発明の実施の形態 2 に係る待機冗長二重化装置の稼動側コントローラの CPU a 0 の動作を示すフローチャートである。

【図 1 2】本発明の実施の形態 2 に係る待機冗長二重化装置の稼動側コントローラの CPU a 1 の動作を示すフローチャートである。

【図 1 3】本発明の実施の形態 3 に係る待機冗長二重化装置のメモリの構成を示す図である。

【図 1 4】本発明の実施の形態 3 に係る待機冗長二重化装置の動作を示すタイミング図である。

【図 1 5】本発明の実施の形態 3 に係る待機冗長二重化装置の動作を示すタイミング図である。 10

【図 1 6】本発明の実施の形態 4 に係る待機冗長二重化装置のメモリの構成を示す図である。

【図 1 7】本発明の実施の形態 5 に係る待機冗長二重化装置のメモリの構成を示す図である。

【図 1 8】本発明の実施の形態 6 に係る待機冗長二重化装置の構成を示す図である。

【図 1 9】本発明の実施の形態 6 に係る待機冗長二重化装置の等値化送信データの構成を示す図である。

【図 2 0】本発明の実施の形態 7 に係る待機冗長二重化装置の構成を示す図である。

【図 2 1】本発明の実施の形態 7 に係る待機冗長二重化装置の共通メモリの構成を示す図である。 20

【図 2 2】本発明の実施の形態 7 に係る待機冗長二重化装置の個別メモリの構成を示す図である。

【図 2 3】本発明の実施の形態 8 に係る待機冗長二重化装置の等値化データ送信データの構成を示す図である。

【発明を実施するための形態】

【0009】

実施の形態 1 .

以下、本発明の実施の形態 1 を図を用いて説明する。

図 1 は、待機冗長二重化装置 100 (以下、装置 100 と称す) の構成を示す図である 30

図において、装置 100 は、稼動側コントローラ 10 a と待機側コントローラ 10 b と等値化データバス 20 により構成される。

稼動側コントローラ 10 a の演算結果は、等値化データバス 20 を介して、待機側コントローラ 10 b へ通知される。

待機側コントローラ 10 b は、稼動側コントローラ 10 a と同じ構成であるため、稼動側コントローラ 10 a を主として各コントローラの構成を説明する。

本明細書では、2つのコントローラを稼動側コントローラ 10 a、待機側コントローラ 10 b として固定して説明しているが、実際の運用では、稼動側コントローラ 10 a も待機側コントローラ 10 b も構成は等しく、それぞれの立場が入れ替わる場合がある。 40

【0010】

稼動側コントローラ 10 a は、CPU ボード 1 a と複数の I/O ボード 2 a と I/O バス 3 a より構成される。CPU ボード 1 a の演算ルーチンに必要な入力データは、I/O バス 3 a を介して I/O ボード 2 a を読み出すことによって得られる。

また、CPU ボード 1 a の演算ルーチンの演算結果である出力データは、I/O バス 3 a を介して I/O ボード 2 a に書き込まれる。

【0011】

装置 100 により制御される機器へのデータの出力は、稼動側コントローラ 10 a 上の I/O ボード 2 a よりおこなわれる。

待機側コントローラ 10 b の I/O ボード 2 b 側に出力データの書き込みをおこなって 50

も機器へのデータ出力はI/Oボード2 b内で切り離されており、待機側コントローラ10 bから機器には出力データは伝わらない。

【0012】

CPUボード1 aは、CPU a0からCPU a nまでの複数個のCPUと、メモリ11 aと、等値化バスコントローラ12 aと、I/Oバスコントローラ13 aと、特許請求の範囲における起動制御部に相当する割込制御14 aと、タイマ15 aを備える。

CPU a0からCPU a nまでの複数個のCPUは、それぞれ異なる演算を行う。

メモリ11 aには、各CPUが演算に使用するデータが配置される。

I/Oバスコントローラ13 aは、各CPUが使用するデータの読み出し要求または書き込み要求に従い、I/Oバス3 aを経由してメモリ11 aと複数のI/Oボード2 aとの間でデータの送信を行う。

10

【0013】

稼働側コントローラ10 aの等値化バスコントローラ12 aは、稼働側コントローラ10 aの各CPUのおこなった演算の結果等を等値化データバス20を経由して待機側コントローラ10 bの等値化バスコントローラ12 bに送信する。

待機側コントローラ10 bの等値化バスコントローラ12 bは受信したデータを待機側コントローラ10 bのメモリ11 bに格納する。

【0014】

また、割込制御14 aは、CPU a0からCPU a nまでのそれぞれのCPUに対して、割込信号バス16 aを使用して割り込みを通知する。

20

更に、割込制御14 aには、タイマ15 aが接続されており、予め設定したCPUに一定間隔で割り込みを通知することができる。

【0015】

図2は、割込制御14 aの構成を示す図である。

割込制御14 aには、CPU a0用からCPU a n用の各割込通知レジスタが用意されており、例えば、CPU a0用の割込通知レジスタにデータをセットすると、CPU a0への割込信号17 a0がONとなり、CPU a0に割り込みが通知される。

この割込通知レジスタと割込信号17 a0～17 a nがCPU毎に存在する。

【0016】

CPU a0用の割込通知レジスタからCPU a n用の割込通知レジスタは、相互に他のCPUからアクセス可能となっており、あるCPUから他のCPUに割り込みを通知することができる。

30

【0017】

図3は、メモリ11 aおよびメモリ11 b内に記録するデータの構成を示す図である。

メモリ11 aとメモリ11 bは、同一の構成となるので、メモリ11 aを用いて説明する。

入力データ11 a1は、I/Oボード2 aを読み出すことで得たデータである。

出力データ11 a2はI/Oボード2 aに書き込むデータである。

入力データ11 a1と出力データ11 a2は、I/Oボード2 aと制御対象となる機器が送受信する際に使用する形式で記録されている。

40

【0018】

また、メモリ11 aには、CPU a0からCPU a nがそれぞれ行う演算ルーチンp0から演算ルーチンpnが格納される。

演算ルーチンp0から演算ルーチンpnは、実際には、それぞれ入力データ11 a1から、それぞれのCPUが演算に使用するデータを演算に適した形式に変換する処理(演算1)と、実際に演算する処理(演算2)と、演算結果をI/Oボード2 aからの出力に適したデータ形式へ変換する処理(演算3)が含まれている。

【0019】

また、メモリ11 aには、CPU a0からCPU a nがそれぞれ行う演算ルーチンp0からpnが使用する演算用のデータが、CPU毎に、演算データpd0から演算データp

50

d nの順で配置される。

これらの演算データの具体的な内容を、CPU a 0が使用する演算データ p d 0を用いて説明する。

【0020】

演算データ p d 0は、実際には複数のデータで構成される。

これらのデータは、3種類のデータに分類される。

1種類目のデータは、入力データ 1 1 a 1からCPU a 0用の演算ルーチン p 0が使用するデータを取得して、演算ルーチン p 0の演算 1によって演算 2に適した形式に変換したデータである。

2種類目のデータは、演算ルーチン p 0の演算 2の結果得たデータと、その過去の蓄積データ群である。

3種類目のデータは、演算ルーチン p 0の演算 2による、演算結果のデータを、演算ルーチン p 0の演算 3によって、I/Oボード 2 aへの出力データ形式に変換したデータである。

【0021】

更に、メモリ 1 1 aは、等値化データ送信バッファ 1 1 a 3と、等値化データ受信バッファ 1 1 a 4及び等値化データ受信バッファ 1 1 a 5を有する。

等値化データ送信バッファ 1 1 a 3は、このコントローラが稼働側コントローラ 1 0 aとして動作する際に使用される。

稼働側コントローラ 1 0 aの等値化バスコントローラ 1 2 aは、等値化データ送信バッファ 1 1 a 3の内容を待機側コントローラ 1 0 bの等値化バスコントローラ 1 2 bに送信する。

また、送信時には等値化バスコントローラ 1 2 aは送信データの妥当性を確認するためのチェックサムを付与して送信する。

【0022】

等値化データ受信バッファ 1 1 a 4と等値化データ受信バッファ 1 1 a 5は、各コントローラが待機側コントローラとして動作する際に使用される。

この点について、待機側コントローラ 1 0 bの動作として以下に説明する。

待機側コントローラ 1 0 bの等値化バスコントローラ 1 2 bは、稼働側コントローラ 1 0 aから受信したデータをチェックサムを用いて確認する。

そして受信データが妥当であると判断した場合は、受信データを等値化データ受信バッファ 1 1 b 4又は等値化データ受信バッファ 1 1 b 5に交互に格納し、不正と判断した場合は、受信データを破棄する。

【0023】

この時、待機側の等値化バスコントローラ 1 2 bがデータの受信に使用していない他方の受信バッファは、待機側のCPUによってデータ処理に使用されている。

【0024】

ここで説明を、稼働側コントローラ 1 0 aに戻す。

稼働側コントローラ 1 0 aのメモリ 1 1 aには、CPU a 0からCPU a nがそれぞれ行う演算ルーチン p 0～演算ルーチン p nの実行回数が、実行カウント p c 0～実行カウント p c nとして格納される。

このカウント値はオーバーフローしても異常とはならず、0から再カウントし直す。

【0025】

図4は、等値化データ送信バッファ 1 1 a 3内に格納される送信データの構成を示す図である。

CPUの識別番号と、各CPU用の演算ルーチンの結果を保存するために必要とされる演算データサイズと、その演算データの3つのデータを一組とし、CPU a 0の演算ルーチン p 0からCPU a nの演算ルーチン p nまでの各組のデータが結合された構造となっている。

【0026】

10

20

30

40

50

次に、稼働側コントローラ 10 a の動作について説明する。

図 5 は、稼働側コントローラ 10 a の動作を示すフローチャートである。

稼働側コントローラ 10 a のタイマ 15 a と割込制御 14 a に対して、予め、周期 T 間隔で CPU a 0 に割り込みが通知されるように設定しておく。

まず、この割り込み通知に基づき、稼働側コントローラ 10 a の CPU a 0 が処理を開始する。

【 0 0 2 7 】

最初に、ステップ S T 1 0 1 0 において、タイマ 15 a からの周期 T に達したことを示す通知を解除する。

この解除により、次回のタイマ 15 a からの周期 T の通知に備える。

次に、ステップ S T 1 0 2 0 において、I / O ボード 2 a からデータの入力を行う。

CPU a 0 は、I / O バスコントローラ 13 a に対して、I / O ボード 2 a からの入力データの読み出し要求を行う。

I / O バスコントローラ 13 a は、読み出し要求を受けて、I / O バス 3 a 経由で複数の I / O ボード 2 a よりデータを読み出し、メモリ 11 a に入力データ 11 a 1 として格納する。

【 0 0 2 8 】

次に、ステップ S T 1 0 3 0 において、CPU a 0 は、他の各 CPU に対して処理開始を通知する。割込制御 14 a 上の CPU a 1 から CPU a n に対応する各割込通知レジスタに対してそれぞれ所定の値を書き込む。

すると、割込制御 14 a は、割込信号 17 a 1 から割込信号 17 a n を ON にして、CPU a 1 から CPU a n に対する割り込みの発生を通知する。

【 0 0 2 9 】

次に、ステップ S T 1 0 4 0 において、CPU a 0 は、演算ルーチン p 0 の演算を行う。

この演算ルーチン p 0 には、先に述べたように、CPU a 0 が演算に使用するデータを演算に適した形式に変換する処理（演算 1）と、実際に演算する処理（演算 2）と、演算結果を I / O ボード 2 a からの出力に適したデータ形式へ変換する処理（演算 3）が含まれている。

最初に、入力データ 11 a 1 内の、CPU a 0 が演算に使用するデータが、演算ルーチン p 0 の演算に適した形式にデータを変換されて演算データ p d 0 として格納される。（演算 1）

次に、本来の演算 2 の処理が動作し、演算結果が演算データ p d 0 に追加して格納される。（演算 2）

更に、演算 2 による演算結果を I / O ボード 2 a に適した形式に変換する処理（演算 3）が動作し、変換後のデータが出力データ 11 a 2 に格納される。

【 0 0 3 0 】

次に、ステップ S T 1 0 5 0 において、演算ルーチン p 0 の実行回数を示すメモリ 11 a 上の実行カウンタ p c 0 の値を 1 つ増加させる。

次に、ステップ S T 1 0 6 0 において、後述する CPU a 1 から CPU a n までの処理がすべて完了するまで待つ。

【 0 0 3 1 】

次に、ステップ S T 1 0 7 0 において、CPU a 1 から CPU a n までの処理がすべて完了すると、各 CPU の処理完了通知を解除し、次回の周期での処理完了通知に備えておく。

次に、ステップ S T 1 0 8 0 において、演算データ p d 0 から演算データ p d n までの演算 1 の結果のデータと演算 2 の結果のデータ（以下、「p 0 演算データ」と表記する）を結合して、図 4 で示す送信データを等値化データ送信バッファ 11 a 3 に格納する。

【 0 0 3 2 】

次に、ステップ S T 1 0 9 0 において、CPU a 0 は、等値化バスコントローラ 12 a

10

20

30

40

50

に対して、等値化データ送信バッファ 1 1 a 3 の送信データを、待機側コントローラ 1 0 b に送信するよう指示する。

等値化バスコントローラ 1 2 a は、CPU a 0 からの指示を受けて、等値化データ送信バッファ 1 1 a 3 のデータを待機側コントローラ 1 0 b に送信する。

【 0 0 3 3 】

次に、CPU a 0 は、ステップ S T 1 0 9 0 における等値化バスコントローラ 1 2 a の送信完了を待たずにステップ S T 1 1 0 0 の処理を指示する。

ステップ S T 1 1 0 0 においては、I / O ボード 2 a へのデータの出力処理を行う。

CPU a 0 は、I / O バスコントローラ 1 3 a に対して、メモリ 1 1 a の出力データ 1 1 a 2 から I / O ボード 2 a への書き込み要求を行う。

I / O バスコントローラ 1 3 a は、この書き込み要求を受けて、I / O バス 3 a 経由で I / O ボード 2 a に対してデータを書き込む。

【 0 0 3 4 】

図 6 は、稼働側コントローラ 1 0 a の CPU a 1 (~ CPU a n) の動作を示すフローチャートである。

次に、CPU a 0 以外の CPU a 1 から CPU a n までの動作を図 6 を用いて説明する。

CPU a 1 から CPU a n までの各 CPU の動作は同じなので、図 6 では、CPU a 1 を用いて説明する。

【 0 0 3 5 】

まず、CPU a 0 からの指令に基づく処理開始通知を受けて CPU a 1 が処理を開始すると、ステップ S T 1 1 1 0 において、CPU a 0 からの処理開始通知を解除する。

この解除により、次の通知に備える。

【 0 0 3 6 】

次に、ステップ S T 1 1 2 0 において、CPU a 1 が行う演算ルーチン p 1 の演算を行う。この演算ルーチン p 1 には、先に述べたように、CPU a 1 が演算に使用するデータを演算に適した形式に変換する処理（演算 1）と、実際に演算する処理（演算 2）と、演算結果を I / O ボード 2 a からの出力に適したデータ形式へ変換する処理（演算 3）が含まれている。

【 0 0 3 7 】

最初に、入力データ 1 1 a 1 の入力データの内の、CPU a 1 が演算に使用するデータが、演算ルーチン p 1 の演算に適した形式にデータを変換されて演算データ p d 1 として格納される。（演算 1）

次に、本来の演算 2 の処理が動作し、演算結果が演算データ p d 1 に追加して格納される。

更に、演算 2 による演算結果を I / O ボード 2 a に適した形式に変換する処理（演算 3）が動作し、変換後のデータが出力データ 1 1 a 2 に格納される。

【 0 0 3 8 】

次に、ステップ S T 1 1 3 0 において、演算ルーチン p 1 の実行回数を示すメモリ 1 1 a 上の実行カウンタ p c 1 の値を 1 つ増加させる。

次に、ステップ S T 1 1 4 0 において、CPU a 0 に対して処理完了を通知する。

処理完了は、割込制御 1 4 a を介して通知しても良いし、メモリ 1 1 a を介して通知しても良い。

【 0 0 3 9 】

次に、待機側コントローラ 1 0 b の動作を説明する。待機側コントローラ 1 0 b 上のタイマ 1 5 b と割込制御 1 4 b に対しても、稼働側コントローラ 1 0 a と同様に、予め、周期 T 間隔で CPU b 0 に割り込みが通知されるように設定しておく。

ただし、稼働側コントローラ 1 0 a と待機側コントローラ 1 0 b でタイミングが同期する仕組みはなく、非同期に動作する。

【 0 0 4 0 】

10

20

30

40

50

図7は、待機側コントローラ10bのCPUb0の動作を示すフローチャートである。

割込制御14bからの割り込み通知に基づき、待機側コントローラ10bのCPUb0が行う処理を図7を用いて説明する。

ステップST2010からステップST2030については、稼働側コントローラ10aのCPUa0が、図5のステップST1010からステップST1030で行う処理と同様である。

従って、CPUb0は、稼働側コントローラ10aのCPUa0と同様に、I/Oボード2bからデータを取得し、メモリ11bに入力データ11b1を記録し、自らデータ処理をし、他のCPUにも同様の処理を指示する。

【0041】

ステップST2035において、CPUb0は、等値化バスコントローラ12bが現時点で等値化データを受信しているバッファと異なる方の等値化データ受信バッファの内容を確認し、データのCPU識別番号が0の部分から、当該データサイズを取得する。

次に、そのサイズ分の演算データを取得して、メモリ11b内の演算データpd0に展開する。

この時、等値化データと共通の部分(上述の演算1の結果のデータと演算2の結果のデータに相当する部分)は、等値化データで上書きされることになる。

なお、演算2の結果部分のデータは、更にその後のステップST2040の演算で上書きされることになる。

【0042】

ステップST2040からステップST2070の処理及び、ステップST2100の処理は、図5において稼働側コントローラ10aのCPUa0が行うステップST1040からステップST1070及び、ステップST1100の処理と同様である。

但し、待機側コントローラ10bのI/Oボード2bは、このコントローラが待機側コントローラとして動作している間は、I/Oボード2b内において制御対象となる機器との接続を切り離しているため、機器に対して出力データは送信されない。

【0043】

図8は、待機側コントローラのCPUb1(～CPUbn)の動作を示すフローチャートである。

次に、CPUb0以外のCPUb1からCPUbnまでの動作を図8を用いて説明する。

CPUb1からCPUbnまでの各CPUの動作は同じなので、図8では、CPUb1を用いて説明する。

【0044】

ステップST2110及び、ステップST2120からステップST2140の処理は、稼働側コントローラ10aのCPUa1が行う、図6におけるステップST1110及びステップST1120からステップST2140と同様である。

図6と異なる点は、ステップST2110の後に、ステップST2115の処理が追加されている点である。

ステップST2115においては、図7のステップST2035と同様に、CPUb1は、等値化バスコントローラ12bが現時点で等値化データを受信しているバッファと異なる方の等値化データ受信バッファの内容を確認し、データのCPU識別番号が1の部分から、当該データサイズを取得する。

そして、そのサイズ分の演算データを取得して、メモリ11b内の演算データpd1に展開する。

この時、等値化データと共通の部分(上述の演算1の結果のデータと演算2の結果のデータに相当する部分)は、等値化データで上書きされることになる。

なお、演算2の結果部分のデータは、更にその後のステップST2120の演算で上書きされることになる。

【0045】

10

20

30

40

50

図 9、図 10 は、これまで説明した装置 100 の動作をタイミング図としてまとめたものである。

各図内のそれぞれの矢印は信号やデータの送信を示す。

図 9 の矢印 A 1、A 2 は、それぞれ図 10 の矢印 A 1、A 2 に繋がる。

2 つの図を並べて見ると、稼働側コントローラ 10 a と、待機側コントローラ 10 b のタイミングを視覚的に理解できる。

本発明の実施の形態 1 に係る待機冗長二重化装置 100 によれば、複数の CPU が処理するデータの等値化処理において、CPU 毎のバスコントローラを不要とすることができる。

また、等値化データバス 20 と I/O バス 3 a、3 b を独立のバスとし、それぞれの I/O バスコントローラも独立しているため、等値化データの送受信と I/O ボードの送受信の同時送信が可能となり、処理の高速化を図ることができる。

【0046】

実施の形態 2 .

以下、本発明の実施の形態 2 を図を用いて実施の形態 1 と異なる部分を中心に説明する。

実施の形態 1 では、各 CPU への処理の開始の通知として割込制御 14 a、14 b を使用していた。

本実施の形態では、割り込みを使用せず、タイマにフラグを配置し、フラグの ON/OFF を割り込みの通知信号として代用することとした。

【0047】

図 11 は、本実施の形態に係る稼働側コントローラの CPU a0 の動作を示すフローチャートである。

実施の形態 1 の CPU a0 の動作を示す図 5 と、本実施の形態の CPU a0 の動作を示す図 11 の差異は、まず、図 5 では、処理開始の割り込みが発生してから処理を開始している点である。

次に、図 11 においては、ステップ ST1005 をループさせて処理開始フラグを監視している点である。

本実施の形態では、CPU a0 がフラグの監視をするので、独立した割込制御は必要ない。

ステップ ST1005 において、CPU a0 が、処理開始を示すフラグが ON になったことを確認すると、ステップ ST1010 においてフラグを解除し、ステップ ST1020 において I/O ボードからのデータ入力処理を行う。

次に、ステップ ST1030 において、タイマ内または、メモリ領域に備えた、他の CPU に処理の開始を指示する各フラグを ON にする。

その他の処理は実施の形態 1 と同様である。

【0048】

図 12 は、本実施の形態に係る稼働側コントローラの CPU a1 (~ an) の動作を示すフローチャートである。

実施の形態 1 の CPU a1 の動作を示す図 6 と、本実施の形態の CPU a1 の動作を示す図 12 の差異は、まず、図 6 では、処理開始の割り込みが発生してから処理を開始している点である。

次に、図 12 においては、ステップ ST1105 をループさせて処理開始フラグを監視している点である。

本実施の形態では、CPU a1 がフラグの監視をするので、独立した割込制御は必要ない。

ステップ ST1105 において、CPU a1 が、処理開始を示すフラグが ON になったことを確認すると、ステップ ST1110 においてフラグを解除し、ステップ ST1120 において演算 p1 を行う。

その他の処理は実施の形態 1 と同様である。

待機側コントローラ 10b のフローチャートは明示しないが、同様にフラグを用いて処理を実現可能である。

【0049】

本発明の実施の形態 2 に係る待機冗長二重化装置によれば、装置内に配置するフラグの ON/OFF を割込通知信号の代用とすることで、特別な割込制御の必要がないシンプルな待機冗長二重化装置を提供できる。

【0050】

実施の形態 3 .

以下、本発明の実施の形態 3 を図を用いて実施の形態 2 と異なる部分を中心に説明する。

図 13 は、本実施の形態に係る稼働側コントローラのメモリ 311a 内に記録するデータの構成を示す図である。

本実施の形態では、CPU 数を超える演算データと演算ルーチンを準備する。

そして、各 CPU に対して複数の演算ルーチンを割り当てる。

図において、「p」の後に続く数字が、当該演算ルーチンを実行する CPU 番号であり、ハイフンに続く数字が、その番号の CPU に割り当てられた演算ルーチンの連番である。

【0051】

図 14、15 は、本実施の形態に係る待機冗長二重化装置の各コントローラの動作状況を示すタイミング図である。

各図内のそれぞれの矢印は信号やデータの送信を示す。

図 14 の矢印 A1、A2 は、それぞれ図 15 の矢印 A1、A2 に繋がる。

2 つの図を並べて見ると、稼働側コントローラと、待機側コントローラのタイミングを視覚的に理解できる。

【0052】

メモリ 311a を上記のように構成することにより、図 14、図 15 に示すように、稼働側コントローラ上の CPU a0 において演算ルーチン p0-1 から演算ルーチン p0-x の複数の演算を可能とし、同様に CPU a1 から CPU an において演算ルーチン p1-1 から演算ルーチン pn-z の複数の演算を実行できる。

【0053】

本実施の形態に係る待機冗長二重化装置によれば、メモリ 311a の構成として、演算ルーチンを実行する CPU 番号の属性データを設けることで、マルチ CPU 環境において、演算ルーチンに対して予め各 CPU を振り分けて演算をさせることができる。

【0054】

実施の形態 4 .

以下、本発明の実施の形態 4 を図を用いて実施の形態 3 と異なる部分を中心に説明する。

図 16 は、稼働側コントローラのメモリ 411a 内に記録するデータの構成の一部を示す図である。

各演算ルーチンに対して、属性データとして実行 CPU 番号、起動周期設定、起動オフセット設定、実行カウント及び送信カウントの設定項目を設けている。

【0055】

起動周期設定項目を用いることで、周期 T の倍数の周期で実行する演算ルーチンの実現を可能とする。

また、周期 T の倍数の周期で実行する演算ルーチンに対して、起動オフセット設定をすることができる。

起動オフセット設定は、演算ルーチンを実行するタイミングを周期 T を 1 として小数で設定する。

例えば、周期 T の 4 倍の周期で実行する演算ルーチンに対して、起動オフセットを 0.25 とすると、周期 T の 4 倍周期かつ、タイマに対して 1/4 周期分遅れて当該演算ルー

10

20

30

40

50

チンを実行する。

【 0 0 5 6 】

本実施の形態では、上述のように周期 T の倍数の周期で動作する演算ルーチンが存在する。演算ルーチンの中には、異なった周期で動作するものが混在することとなる。

一方、等値化データは周期 T で送信されるため、周期 T 単位では動作しない演算ルーチンのデータは送信する必要はない。

よって、これらの演算ルーチンのデータの再送信を避けるため、実行カウントと送信カウントをメモリ 4 1 1 a 内に追加して利用する。

【 0 0 5 7 】

稼働側コントローラの CPU a 0 が行う等値化送信データの作成時において、送信カウントと実行カウントを比較し、二つのカウントの値が異なる場合は、その演算ルーチンの演算データは等値化データとして送信すべきデータと判断して送信バッファに送信する。

もし、送信カウントと実行カウントを比較して、これらが一致する場合は、既にデータの送信は行われていて、当該演算ルーチンの新たな実行が行われていないと判断し、送信バッファへの送信はしない。

なお、送信カウントは、送信バッファへの送信処理完了後に実行カウントの値に書き換えるものとする。

【 0 0 5 8 】

本実施の形態に係る待機冗長二重化装置によれば、メモリ 4 1 1 a 内に起動周期設定と起動オフセット設定を設けることで、周期 T の倍数の周期で、更に必要に応じて、演算ルーチン毎に起動オフセットを設定して演算を実行できる。

また、送信カウントを設け、実行カウントと比較することで、不要な等値化データの送信が削減され、等値化データの送受信時間を短くすることができる。

【 0 0 5 9 】

実施の形態 5 .

以下、本発明の実施の形態 5 を図を用いて実施の形態 4 と異なる部分を中心に説明する。

図 1 7 は、稼働側コントローラのメモリ 5 1 1 a 内に記録するデータの構成の一部分を示す図である。

各 CPU が実行する演算ルーチンに対して、起動継続可否設定を追加する。

起動継続可否設定が可の場合、演算が設定周期 T で完了しなかった場合でも、演算処理を一時保留とし、データの入出力をおこない、次の周期で動作を再開して継続する仕組みを追加する。

【 0 0 6 0 】

起動継続可否設定の追加により、演算処理を保留した演算ルーチンは、途中でデータの入出力が行われるが、この入出力動作が支障をきたさない場合、一定周期 T 内で動作する演算ルーチンと一定周期 T 内で動作しない演算ルーチンの共存が可能となる。

本実施の形態に係る待機冗長二重化装置によれば、処理時間が大きく異なる処理ルーチンが一部に存在する場合でも、全体の処理に影響を及ぼさない待機冗長二重化装置を提供できる。

【 0 0 6 1 】

実施の形態 6 .

以下、本発明の実施の形態 6 を図を用いて実施の形態 1 と異なる部分を中心に説明する。

図 1 8 は、待機冗長二重化装置 6 0 0 (以下、装置 6 0 0 と称す)の構成を示す図である。

稼働側コントローラ 6 1 0 a の CPU ボード 6 0 1 a には、2つの等値化バスコントローラ 1 2 a 1、1 2 a 2 を備えている。

また、待機側コントローラ 6 1 0 b の CPU ボード 6 0 1 b には、2つの等値化バスコントローラ 1 2 b 1、1 2 b 2 を備えている。

10

20

30

40

50

そして、等値化バスコントローラ 1 2 a 1 と等値化バスコントローラ 1 2 b 1 を等値化データバス 2 0 で接続し、等値化バスコントローラ 1 2 a 2 と等値化バスコントローラ 1 2 b 2 を等値化データバス 2 1 で接続している。

等値化バスコントローラのセットと、等値化データバス 2 0、2 1 をそれぞれ二重化した構成として装置 6 0 0 の信頼性を高めている。

【 0 0 6 2 】

次に、装置 6 0 0 における、二重化された等値化データバスを使用する送信データの授受方法を説明する。

図 1 9 は、稼働側コントローラ 6 1 0 a から待機側コントローラ 6 1 0 b に送信される等値化送信データの構成の一部を示す図である。

10

【 0 0 6 3 】

送信データには、各演算ルーチンに対する実行カウント p c 0 から実行カウント p c n を連番として有している。

稼働側コントローラの CPU a 0 は、図 5 のステップ S T 1 0 9 0 での等値化データの送信開始にあたって、図 1 9 に示した送信データを、等値化バスコントローラ 1 2 a 1 と等値化バスコントローラ 1 2 a 2 の 2 つのコントローラに対して送信する。

待機側コントローラの各 CPU は、図 7 のステップ S T 2 0 3 5 または、図 8 のステップ S T 2 1 1 5 における等値化受信データの展開時において、等値化バスコントローラ 1 2 a 1 より等値化バスコントローラ 1 2 b 2 が受信した等値化受信データの実行カウントと、等値化バスコントローラ 1 2 a 2 より等値化バスコントローラ 1 2 b 2 が受信した等値化受信データの実行カウントを比較し、カウントが進んでいる方の演算データをメモリ 1 1 b に展開することとする。

20

【 0 0 6 4 】

本発明の実施の形態 6 に係る待機冗長二重化装置 6 0 0 によれば、等値化バスコントローラと、等値化データバスを二重化し、送受信するデータにカウントを付けることにより、いずれかのルートに故障が発生しても、もう一方の系統を使用してデータの等値化を図ることができる。

【 0 0 6 5 】

実施の形態 7 .

以下、本発明の実施の形態 7 を図を用いて実施の形態 1 と異なる部分を中心に説明する

30

。図 2 0 は、待機冗長二重化装置 7 0 0 (以下、装置 7 0 0 と称す)の構成を示す図である。

CPU ボード 7 0 1 a 上の CPU a 0 から CPU a n に対して、それぞれ個別にメモリ a 0 からメモリ a n を追加する。

同様に CPU ボード 7 0 1 b 上の CPU b 0 から CPU b n に対して、それぞれ個別にメモリ b 0 からメモリ b n を追加する。

【 0 0 6 6 】

図 2 1 は、このときの各 CPU が共通に使用するメモリ 7 1 1 a 及びメモリ 7 1 1 b 内のデータ構成を示す図である。

40

メモリ 7 1 1 a には、CPU a 0 から CPU a n が共通に使用するデータを配置する。

同様に、メモリ 7 1 1 b には、CPU b 0 から CPU b n が共通に使用するデータを配置する。

図 2 2 は、CPU 単位で個別に追加するメモリ a 0 からメモリ a n、メモリ b 0 からメモリ b n の構成を示す図である。

CPU a 0 から CPU a n、CPU b 0 から CPU b n がそれぞれ個別に使用するデータをのみを配置する。

【 0 0 6 7 】

本発明の実施の形態 7 に係る待機冗長二重化装置 7 0 0 によれば、メモリ 7 1 1 a またはメモリ 7 1 1 b は共通的に使用するデータのみの構成となり、メモリ 7 1 1 a、メモリ

50

7 1 1 b へのアクセス頻度を抑制できる。

これにより、他のCPUからのアクセスの集中により、演算が停止する頻度が減少するので、装置700の演算性能を向上することができる。

【0068】

実施の形態8 .

以下、本発明の実施の形態8を図を用いて実施の形態1と異なる部分を中心に説明する。

図23は、実施の形態8の等値化送信データの構成を示す図である。

送信データに各演算ルーチンの識別コードと版コードを追加する。

待機側コントローラの各CPUは、図7のステップST2035または図8のステップST2115にて、等値化データを展開するとき、受信した等値化データ内の各演算ルーチンの識別コードと版コードを、待機側コントローラ側でもつ各演算ルーチンの識別コードと版コードと比較する。

これらのコードが不一致の場合は、データの展開処理を実施せず、異常をコントローラの使用者に通知する。

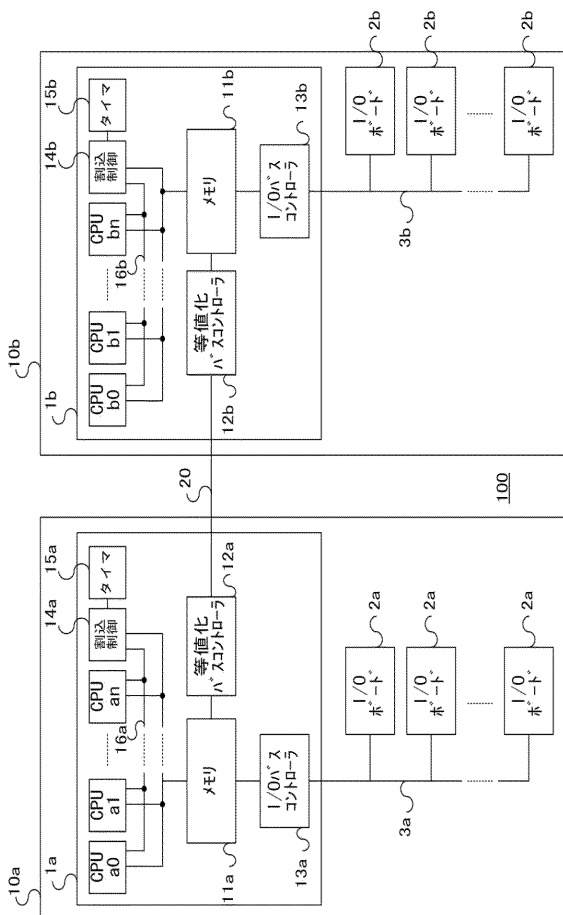
【0069】

本発明の実施の形態8に係る待機冗長二重化装置によれば、稼働側コントローラと待機側コントローラの間で、万一演算ルーチンが異なる状態となっても、誤ったデータで装置全体が等値化することを防止することができ、待機冗長二重化装置の信頼性を向上できる。

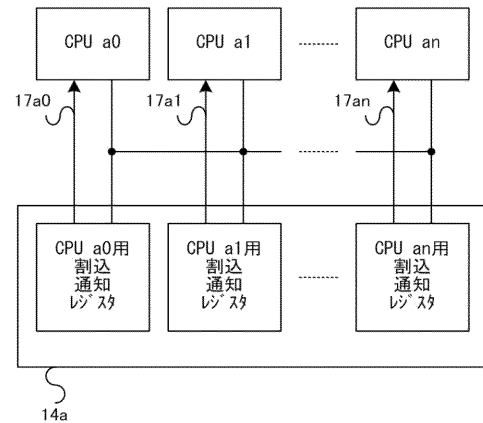
【0070】

尚、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、省略することが可能である。

【図1】



【図2】



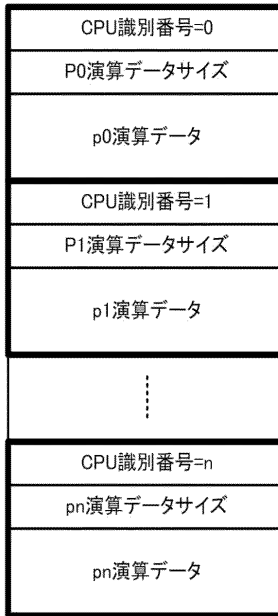
10

20

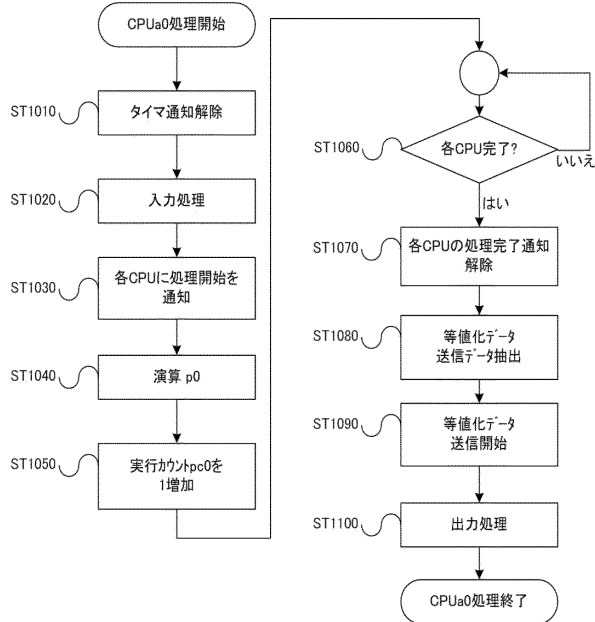
【図3】



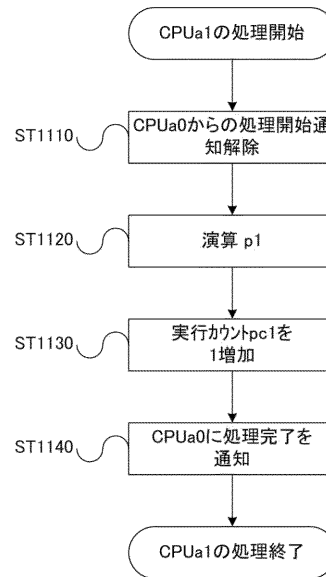
【図4】



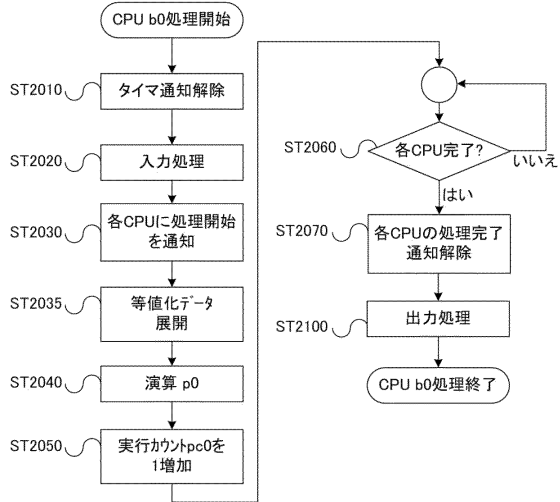
【図5】



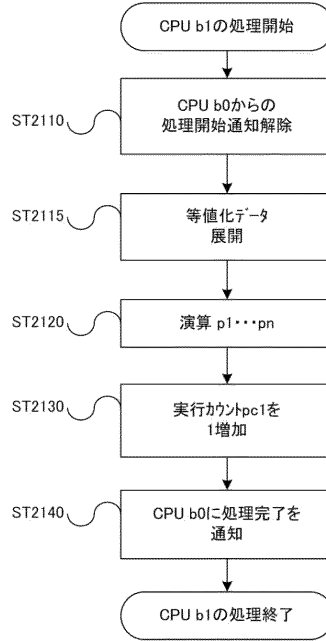
【図6】



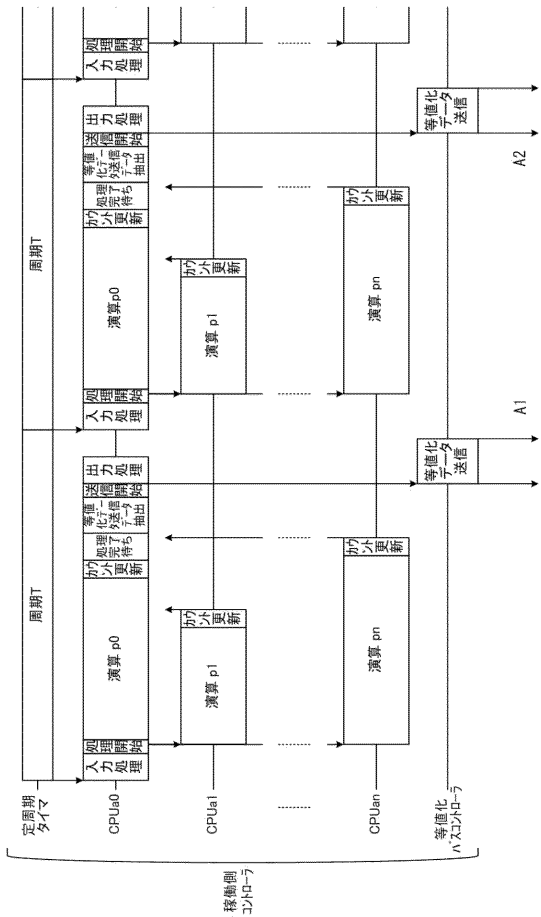
【図7】



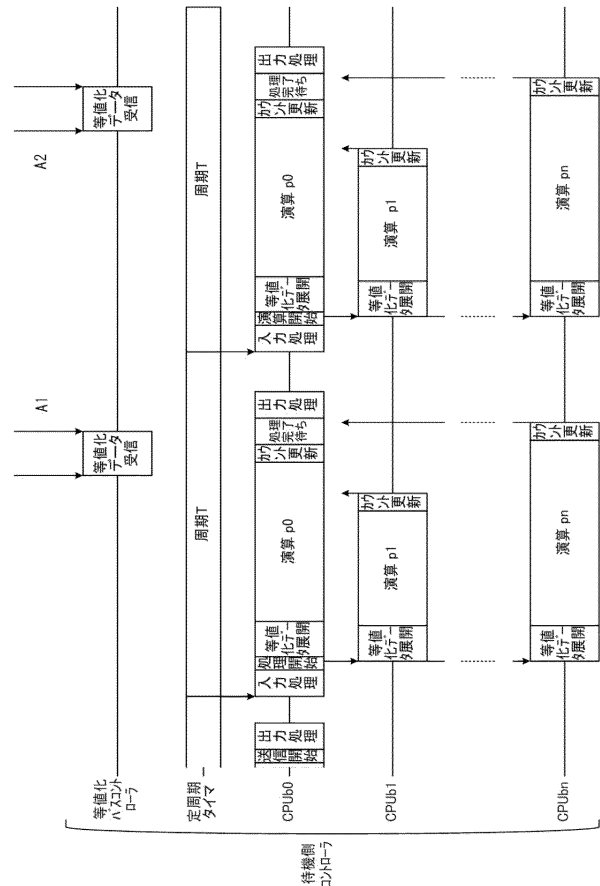
【図8】



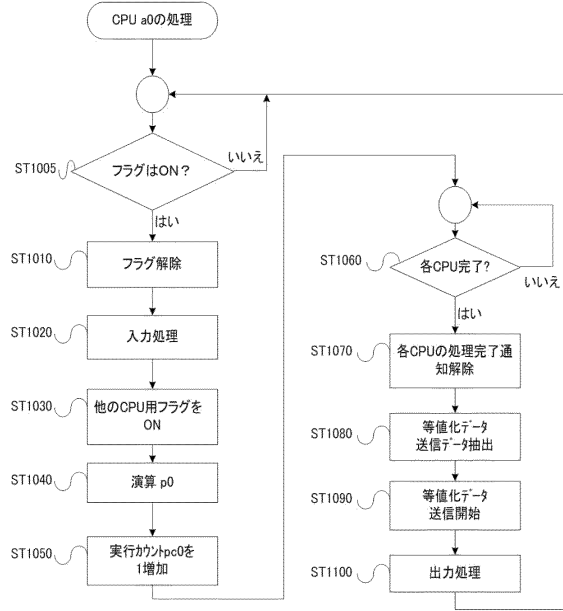
【図9】



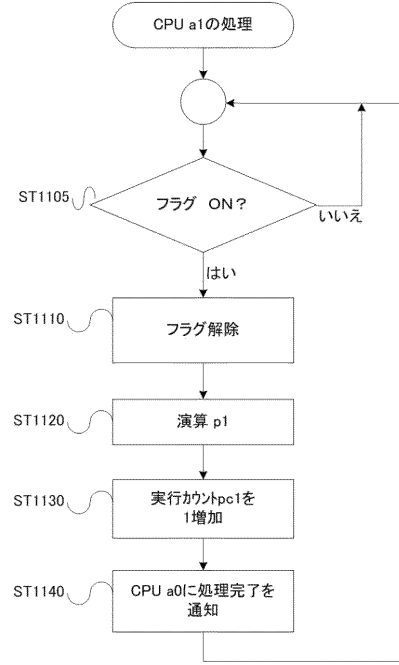
【図10】



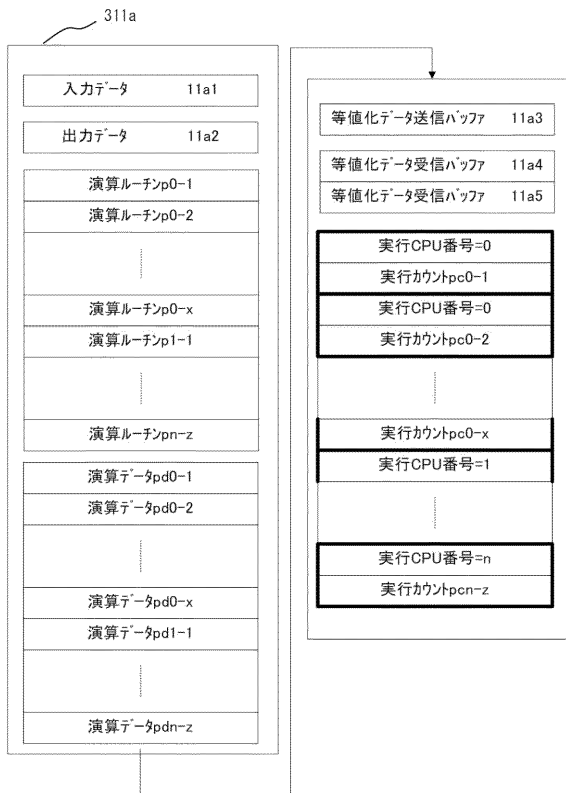
【図 1 1】



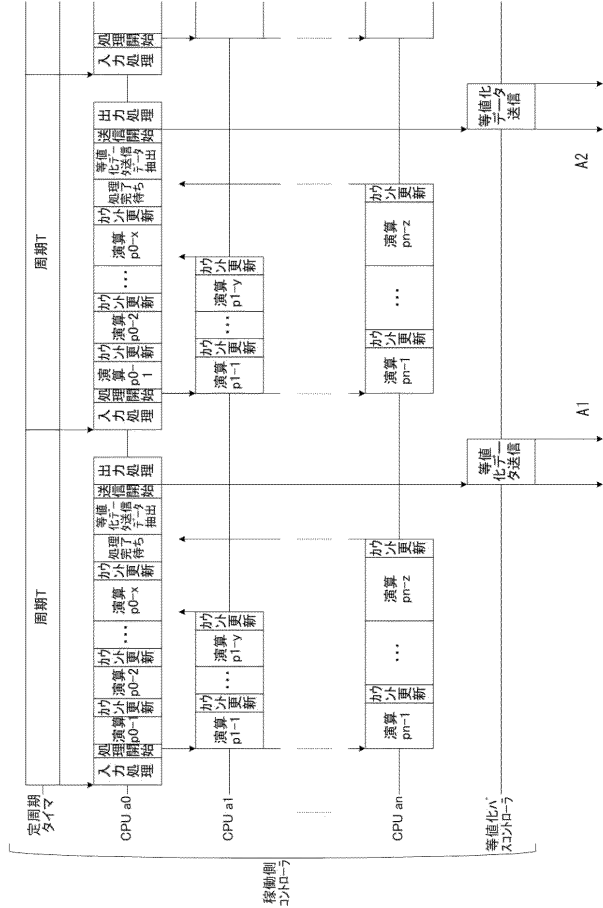
【図 1 2】



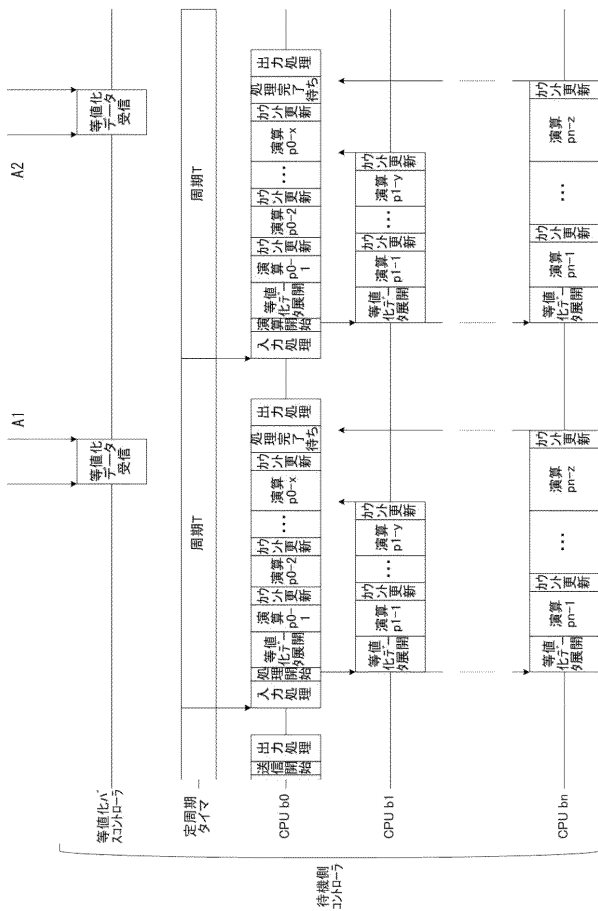
【図 1 3】



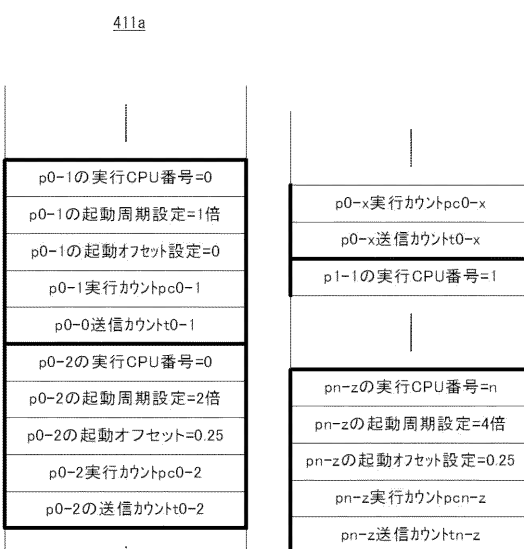
【図 1 4】



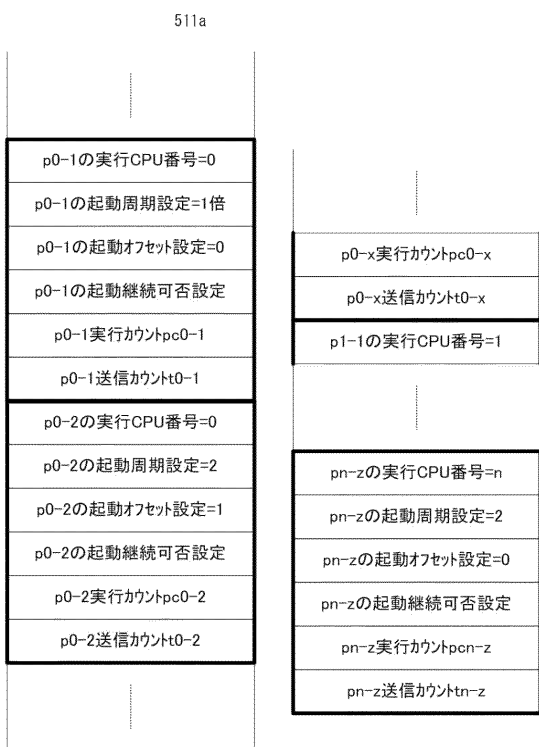
【図15】



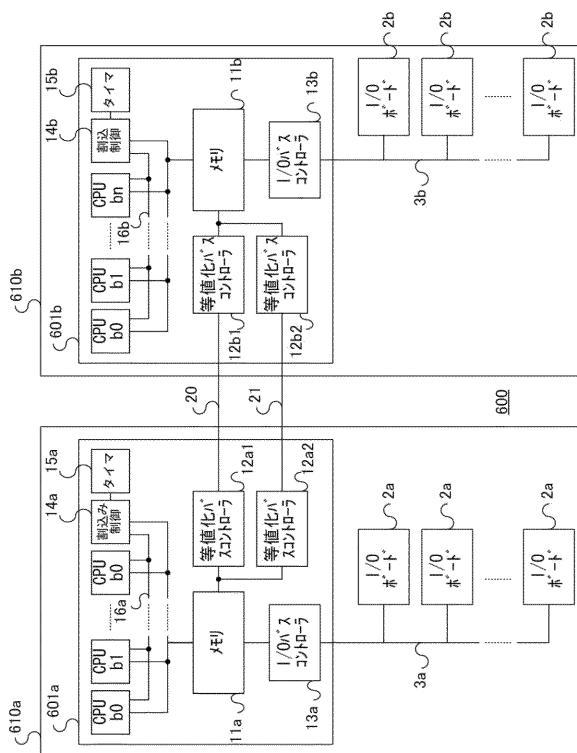
【図16】



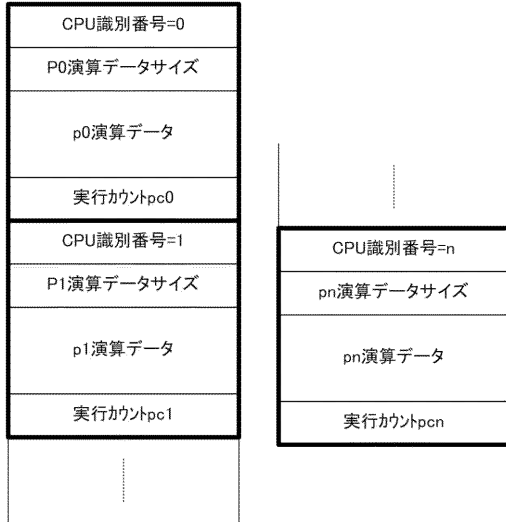
【図17】



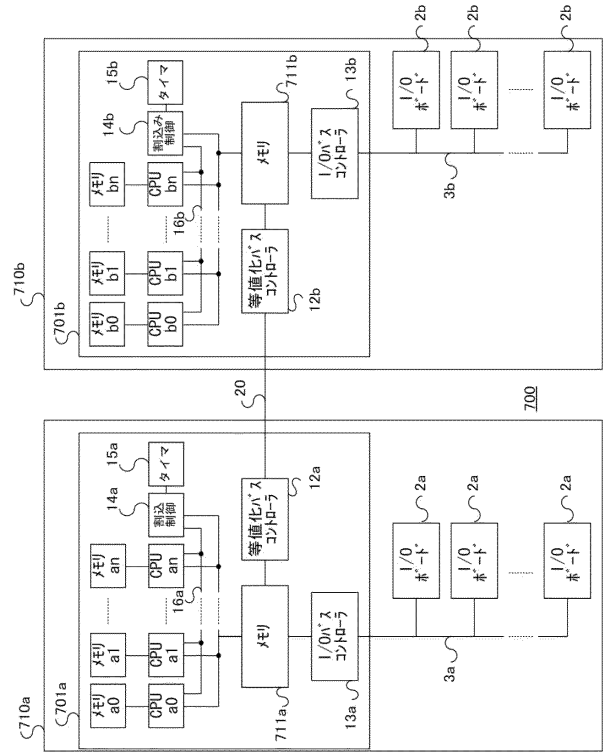
【図18】



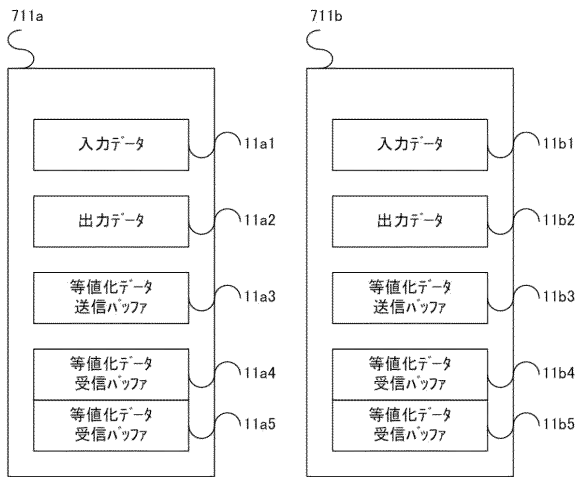
【図19】



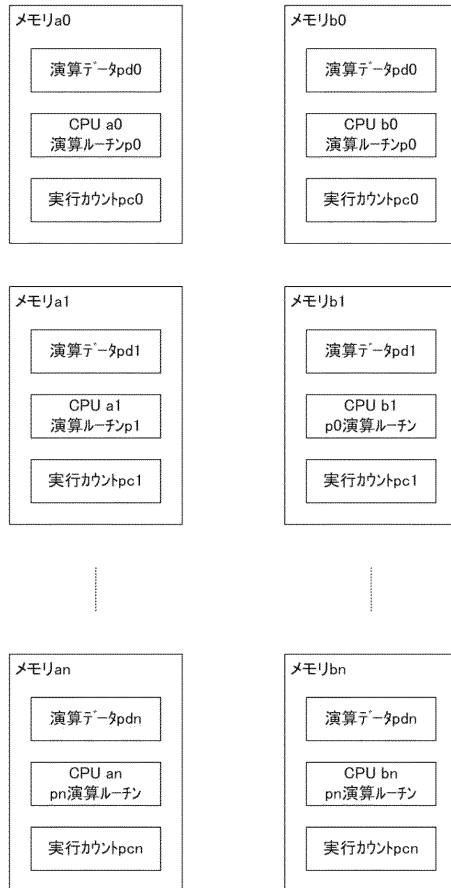
【図20】



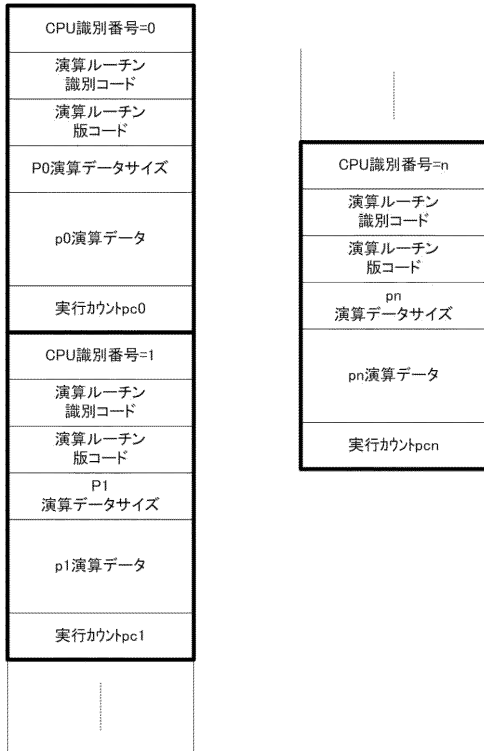
【図21】



【図22】



【図 23】



フロントページの続き

審査官 佐藤 彰洋

- (56)参考文献 特開2009-187314(JP,A)
特開平09-044203(JP,A)
特開2006-031727(JP,A)
特開2011-253390(JP,A)
特開2008-047052(JP,A)
特開2009-294961(JP,A)
特開平08-016534(JP,A)
特開平06-348527(JP,A)
特開平02-156363(JP,A)
特開平07-200044(JP,A)
特開平04-241035(JP,A)
特開2000-305919(JP,A)

(58)調査した分野(Int.Cl., DB名)

G05B 9/02-9/03
G06F 11/20