

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3573546号

(P3573546)

(45) 発行日 平成16年10月6日(2004.10.6)

(24) 登録日 平成16年7月9日(2004.7.9)

(51) Int. Cl.<sup>7</sup>

G06F 15/177

F I

G06F 15/177 681C

請求項の数 4 (全 26 頁)

(21) 出願番号	特願平7-281021	(73) 特許権者	000005223 富士通株式会社 神奈川県川崎市中原区上小田中4丁目1番 1号
(22) 出願日	平成7年10月27日(1995.10.27)	(74) 代理人	100092978 弁理士 真田 有
(65) 公開番号	特開平9-128351	(72) 発明者	上埜 治彦 神奈川県川崎市中原区上小田中1015番 地 富士通株式会社内
(43) 公開日	平成9年5月16日(1997.5.16)	審査官	殿川 雅也
審査請求日	平成12年12月27日(2000.12.27)		

最終頁に続く

(54) 【発明の名称】 並列計算機における並列プロセススケジューリング方法および並列計算機用処理装置

(57) 【特許請求の範囲】

【請求項1】

個別に処理を実行する複数の処理装置と、これらの複数の処理装置を相互に通信可能に接続する通信網とをそなえ、ある1つのジョブを、該複数の処理装置のうちの2以上の処理装置により、ステップ毎に同期させながら並列プロセスとして実行する並列計算機において、

データ処理動作とは非同期に行なわれるパケット送受信により該通信網を介して送信側処理装置から受信側処理装置へデータを転送し、特定種別のパケットであるメッセージパケットを受信すると、その受信データを、ベースアドレスと書込ポイントとの加算値に従って主記憶上のサイクリックキューであるメッセージ受信キューに格納するとともに、該書込ポイントの指示値を、該メッセージ受信キューにおける次の空き領域の先頭アドレスに更新し、

各処理装置が、当該並列プロセスとして今回のステップで割り付けられた処理を完了し、1以上の他処理装置が当該並列プロセスとして今回のステップで割り付けられた処理を完了するのを待つ並列同期待ち状態になった際、

当該並列プロセスの並列同期待ち状態になった処理装置では、

当該並列プロセスの割付を禁止すべく当該並列プロセスを不活性状態にするとともに、並列同期待ち状態になってから前記他処理装置が今回のステップを完了するまでの間に当該処理装置へ転送される予定の全メッセージパケットのデータ容量等の情報に基づいて、該書込ポイントの指示値の期待値を、当該並列プロセスの割付を再開する際に満たされるべ

10

20

き条件として設定し、

当該処理装置により実行可能な他のジョブが存在する場合には、該他のジョブについてのプロセスの割付を行ない、

該期待値と該書込ポイントの実指示値とを比較し前記メッセージパケットの受信量が期待値に到達したことを検知した場合に前記条件が満たされたものと判断して、現在実行中の処理に対する割込み信号を生成し、当該並列プロセスの割付を再開すべく当該並列プロセスを活性状態にすることを特徴とする、並列計算機における並列プロセススケジューリング方法。

【請求項 2】

複数のジョブがそれぞれ並列プロセスとして実行されている場合、

メッセージパケット中に、各ジョブに対応したプロセス識別子を設定し、

各処理装置において、該期待値および該ベースアドレスを前記プロセス識別子毎に設定するとともに、該メッセージ受信キューおよび該書込ポイントを前記プロセス識別子毎にそなえ、

当該並列プロセスの並列同期待ち状態になった処理装置では、該メッセージ受信キューへのデータ書込、該書込ポイントの更新、該期待値の設定および前記割込み信号の生成を前記プロセス識別子毎に行なうことを特徴とする、請求項 1 記載の並列計算機における並列プロセススケジューリング方法。

【請求項 3】

通信網を介して複数の他処理装置と相互に通信可能に接続され並列計算機を構成する処理装置であって、ある 1 つのジョブを、該複数の他処理装置のうちの 1 以上の他処理装置とともに、ステップ毎に同期させながら並列プロセスとして実行する並列計算機用処理装置において、

当該並列プロセスとして今回のステップで割り付けられた処理を完了し、前記 1 以上の他処理装置が当該並列プロセスとして今回のステップで割り付けられた処理を完了するのを待つ並列同期待ち状態になった際に、当該並列プロセスの割付を禁止すべく当該並列プロセスを不活性状態にする不活性化機能と、

該不活性化機能による当該並列プロセスの不活性化と同時に、当該並列プロセスの割付を再開する際に満たされるべき条件を設定する条件設定機能と、

実行可能な他のジョブが存在する場合には、当該並列プロセスが不活性状態である期間中、該他のジョブについてのプロセスの割付を行なう割付機能と、

前記条件が満たされると現在実行中の処理に対する割込み信号を生成する割込み生成機能と、

該割込み生成機能からの割込み信号に応じて、当該並列プロセスの割付を再開すべく当該並列プロセスを活性状態にする活性化機能と、

データ処理動作とは非同期に行なわれるパケット送受信により該通信網を介して該複数の他処理装置との間でデータ転送を行なう転送処理部とをそなえ、

該転送処理部が、当該並列プロセスを実行する前記 1 以上の他処理装置から特定種別のパケットであるメッセージパケットを受信すると、その受信データを、ベースアドレスと書込ポイントとの加算値に従って主記憶上のサイクリックキューであるメッセージ受信キューに格納するとともに、該書込ポイントの指示値を、該メッセージ受信キューにおける次の空き領域の先頭アドレスに更新するように構成され、

該割込み生成機能が、

該条件設定機能により、並列同期待ち状態になってから前記 1 以上の他処理装置が今回のステップを完了するまでの間に前記 1 以上の他処理装置から転送されてくる予定の全メッセージパケットのデータ容量等の情報に基づいて、該書込ポイントの指示値の期待値を設定される期待値レジスタと、

該期待値レジスタに設定された期待値と該書込ポイントの実指示値とを比較する比較器と、

該比較器による比較結果に応じ前記割込み信号を生成する割込み生成回路とから実現さ

10

20

30

40

50

れていることを特徴とする、並列計算機用処理装置。

【請求項 4】

複数のジョブがそれぞれ並列プロセスとして実行されている場合、メッセージパケット中に、各ジョブに対応したプロセス識別子が設定され、該期待値および該ベースアドレスが前記プロセス識別子毎に設定されるとともに、該メッセージ受信キューおよび該書込ポインタが前記プロセス識別子毎にそなえられていることを特徴とする、請求項 3 記載の並列計算機用処理装置。

【発明の詳細な説明】

【0001】

(目次)

発明の属する技術分野

従来技術(図9～図12)

発明が解決しようとする課題(図11, 図12)

課題を解決するための手段(図1)

発明の実施の形態

(a) 本実施形態の並列計算機の説明(図2, 図3)

(b) 第1実施形態の並列計算機用処理装置の説明(図4, 図5)

(c) 第2実施形態の並列計算機用処理装置の説明(図6, 図7)

(d) 第3実施形態の並列計算機用処理装置の説明(図8)

発明の効果

【0002】

【発明の属する技術分野】

本発明は、複数の処理装置〔以下、PE(Processor Element)という〕を相互に通信可能に接続して構成される並列計算機において、ある1つのジョブを2以上の処理装置によりステップ毎に同期させながら並列プロセスとして実行する際に適用される並列プロセススケジューリング方法、および、この方法を適用される処理装置に関し、特に、複数業務を複数PEにより並列的に行なう分散主記憶MIMD(Multiple Instruction stream Multiple Data stream)型並列計算機システムに用いて好適の技術に関する。

【0003】

【従来技術】

近年、例えば数値計算や画像処理等のように膨大なデータを高速に処理する必要性から、コンピュータシステムの高速化や大容量化が要求されている。これに伴って、複数のPEをそなえて相互に通信を行ないながら並列的に処理を行なう並列処理技術が研究・開発されている。

【0004】

一般に、並列計算機システムでは、例えば図9に示すように、n台のPE(PE番号として#0～#n-1が付与されているものとする)101が、通信網としてのPE間結合網100を介して相互に通信可能に接続されている。

各PE101には、図10に示すように、転送処理部102, 命令処理部(CPU)103および主記憶104がそなえられている。

【0005】

ここで、転送処理部102は、主記憶104上のデータの送受信処理を行なうものであり、命令処理部103は、PE101相互間の通信に際してプログラム処理を行なうものである。

なお、上述のように転送処理部102と命令処理部103とを独立に設けることにより、命令処理部103の負荷とオーバヘッドとを削減できるようになっている。また、転送処理部102は、送信処理と受信処理とを同時並列的に行なえるように構成され、これにより、データ転送速度とデータ転送効率との向上をはかっている。

【0006】

10

20

30

40

50

ところで、分散主記憶M I M D型並列計算機システムにおいては、通常、1つのジョブは、複数の異なるP E 1 0 1により並列的に処理されるプロセス（以下、並列プロセスという）として実行されている。このように実行されるジョブを並列ジョブと呼ぶ。さらに、近年、複数の並列ジョブの多重実行機能や、並列ジョブと非並列ジョブとの多重実行機能が要求されている。

#### 【0007】

システム内で複数の並列ジョブ（または、並列ジョブと非並列ジョブと）が多重実行されている時には、各P E 1 0 1上でプロセスをスケジュールし、プロセススイッチ（プロセスの切替）を行なう必要がある。このとき、複数のP E 1 0 1の相互間で協調することなく並列プロセスをスケジュールすると、並列プロセスのミクロな実行時間差のために並列プロセスの同期待ち時間が大きくなるという課題がある。この同期待ち時間の増大を避けるためには、並列プロセスのスケジューリングでは、複数のP E 1 0 1の相互間で協調し合ったスケジューリング、即ち、P E間協調スケジューリングを行なう必要がある。

10

#### 【0008】

図11は、システムで1つの並列ジョブのみを実行している場合の一般的な並列プロセススケジューリングの例を示すもので、この図11に示す例では、1つのジョブが、5台のP E 1 0 1上で、それぞれプロセス番号0～4を付された5つのプロセスとして並列的に実行されており、各プロセスは、同期点1, 2で示すように、ステップ毎に同期しながら実行されている。

#### 【0009】

一方、図12は、並列プロセスのP E間協調スケジューリングとしてギャングスケジューリングと呼ばれる同期スケジューリングを行なった場合の例を示すもので、この図12に示す例でも、図11に示した例と同様、5台のP E 1 0 1上で、それぞれプロセス番号0～4を付された5つのプロセスとして並列的に実行されており、各プロセスは、同期点1, 2で示すように、ステップ毎に同期しながら実行されている。

20

#### 【0010】

このギャングスケジューリングでは、ブロードキャストされたプロセス切替指令を受信したこと、または、P E 1 0 1の相互間で同期した時計が予定時刻になったことを契機にして、全ての並列プロセスが同時に割り付けられる。例えば図12に示す例では、タイムスライス毎に、他の並列ジョブや非並列ジョブのプロセス（図12では図示省略）がディスパッチ（割付）される。

30

#### 【0011】

このような同期スケジューリングによれば、並列プロセスをプロセススイッチしたこと起因するプロセス間同期オーバーヘッドの増大は起こらない。また、図11に示すようにシステムで並列ジョブが1つだけ動作している場合に比べ、単位時間当たりで並列ジョブの実行時間割合倍の性能向上を期待できる。

なお、図11および図12中、 $t$ は1タイムスライスであり、“ ”は各プロセスが並列同期待ち状態になった時点（つまり、今回のステップで割り付けられた処理を完了した時点）を示し、“ ”は各プロセスを実行する各P E 1 0 1で並列同期（バリア）を検出した時点（つまり、当該並列プロセスを実行する全てのP E 1 0 1が今回のステップで割り付けられた処理を完了した時点）を示している。

40

#### 【0012】

また、横方向の太線は、プロセスが実際に実行されている期間を示し、横方向の細線は、プロセスが並列同期待ち状態であることを示している。この細線で示すように、従来、各プロセスは、並列同期待ち状態で実際には何ら処理を行なわない状態であっても、並列同期を検出するまで、各タイムスライスに対してディスパッチ（割付）されている。

#### 【0013】

##### 【発明が解決しようとする課題】

しかしながら、図11および図12に示すいずれの並列プロセススケジューリング手法でも、各プロセスは、並列同期待ち状態で実際には何ら処理を行なわない状態であっても、

50

並列同期を検出するまで、各タイムスライスに対してディスパッチされている。

【0014】

従って、他に実行可能なプロセスが存在する場合でも、並列同期待ち状態の並列プロセスに対してタイムスライスが与えられてしまい、各並列プロセスの処理時間に差がある場合には、システム全体のスループット性能を低下させるなどの課題があった。

本発明は、このような課題に鑑み創案されたもので、並列同期待ち状態にある並列プロセスはディスパッチせず、代わりに実行可能な他のジョブのプロセスをディスパッチできるようにして、システムのスループット性能を低下させない協調スケジューリングを実現した、並列計算機における並列プロセススケジューリング方法および並列計算機用処理装置を提供することを目的とする。

10

【0015】

【課題を解決するための手段】

図1は本発明の原理ブロック図であり、この図1に示すように、本発明の並列プロセススケジューリング方法を適用される並列計算機1は、個別に処理を実行する複数の処理装置(以下、PEという)2と、これらの複数のPE2を相互に通信可能に接続する通信網3とをそなえて構成され、ある1つのジョブを、複数のPE2のうちの2以上のPE2により、ステップ毎に同期させながら並列プロセスとして実行するようになっている。

【0016】

そして、各PE2は、不活性化機能4、条件設定機能5、割付機能6、割込み生成機能7および活性化機能8を有している。

20

ここで、不活性化機能4は、当該並列プロセスとして今回のステップで割り付けられた処理を完了し、他のPE2が当該並列プロセスとして今回のステップで割り付けられた処理を完了するのを待つ並列同期待ち状態になった際に、当該並列プロセスの割付を禁止すべく当該並列プロセスを不活性化状態にする機能である。

【0017】

また、条件設定機能5は、不活性化機能4による当該並列プロセスの不活性化と同時に、当該並列プロセスの割付を再開する際に満たされるべき条件を設定する機能であり、割付機能6は、実行可能な他のジョブが存在する場合には、当該並列プロセスが不活性化状態である期間中、その他のジョブについてのプロセスの割付を行なう機能である。

【0018】

さらに、割込み生成機能7は、条件設定機能5により設定された前記条件が満たされた場合、現在実行中の処理に対する割込み信号を生成する機能であり、活性化機能8は、割込み生成機能7からの割込み信号に応じて、当該並列プロセスの割付を再開すべく当該並列プロセスを活性化状態にする機能である。

30

上述のような機能4~8を有するPE2では、実行中の並列プロセスが並列同期待ち状態になると、不活性化機能4により当該並列プロセスを不活性化することで、この並列プロセスの割付が禁止され、代わって、実行可能な他のジョブが存在する場合には、割付機能6により、この他のジョブについてのプロセスの割付が行なわれる。

【0019】

この他のジョブの実行中に条件設定機能5により設定された前記条件が満たされると、割込み生成機能7により、現在実行中の処理に対する割込み信号が生成された後、活性化機能8により、並列同期待ち状態であった並列プロセスが活性化され、この並列プロセスの割付が再開される。

40

このようにして、並列同期待ち状態にある並列プロセスをディスパッチせず代わりに実行可能な他のジョブのプロセスをディスパッチするとともに、所定条件が満たされると並列プロセスの割付を再開して次のステップの処理を実行するといった、並列プロセスの協調スケジューリングを実現することができる(請求項1,3)。

【0020】

割込み生成機能7を実現するための手法としては、下記項目〔1〕~〔3〕の3種類がある。

50

## 〔 1 〕 転送パケット数に応じた割込み生成機能

並列計算機 1 において、データ処理動作とは非同期に行なわれるパケット送受信により通信網 3 を介して送信側 P E 2 から受信側 P E 2 へデータ（パケット）を転送すべく、各 P E 2 にデータ転送用の転送処理部がそなえられている場合、割込み生成機能 7 を、カウンタ，期待値レジスタ，比較器および割込み生成回路から構成することができる。

## 【 0 0 2 1 〕

ここで、カウンタは、当該並列プロセスの並列同期待ち状態になると、当該並列プロセスを実行する他 P E 2 から転送されてくるパケットの数を、カウントアップまたはカウントダウンすることにより計数するものであり、期待値レジスタは、条件設定機能 5 により、並列同期待ち状態になってから他 P E 2 が今回のステップを完了するまでの間に他 P E 2 から転送されてくる予定のパケットの数に基づいて、カウンタによるカウント値の期待値を設定されるものである。また、比較器は、期待値レジスタに設定された期待値とカウンタによる実カウント値とを比較するものであり、割込み生成回路は、比較器による比較結果に応じ前記割込み信号を生成するものである。

10

## 【 0 0 2 2 〕

このように構成された割込み生成機能 7 を有する P E 2 では、実行中の並列プロセスが並列同期待ち状態になると、当該並列プロセスを実行する他 P E 2 から自 P E 2 へ転送されてくるパケットの数が、カウンタをカウントアップまたはカウントダウンすることにより計数される。

そして、そのカウンタによる実カウント値と条件設定機能 5 により期待値レジスタに設定された期待値とが一致したことが比較器にて検知されると、割込み生成回路により割込み信号が生成される。つまり、転送パケット数が期待値に到達したことを、並列プロセスの割付を再開する条件（並列同期待ち状態の解除条件）として、割込み生成機能 7 を実現することができる。

20

## 【 0 0 2 3 〕

なお、並列計算機 1 上で複数のジョブがそれぞれ並列プロセスとして実行されている場合、パケット中に、各ジョブに対応したプロセス識別子を設定し、各 P E 2 において、期待値をプロセス識別子毎に設定するとともに、カウンタによるカウント値を格納する領域をプロセス識別子毎に主記憶上に確保し、カウンタの計数，期待値の設定および割込み信号の生成をプロセス識別子毎に行なうことにより、ジョブ毎に、並列プロセスの非活性化 / 活性化を行なうことができる。

30

## 【 0 0 2 4 〕

## 〔 2 〕 メッセージパケットの受信量に応じた割込み生成機能

並列計算機 1 において、データ処理動作とは非同期に行なわれるパケット送受信により通信網 3 を介して送信側 P E 2 から受信側 P E 2 へデータ（パケット）を転送すべく各 P E 2 にデータ（パケット）転送用の転送処理部をそなえ、この転送処理部が、当該並列プロセスを実行する他 P E 2 から特定種別のパケットであるメッセージパケットを受信すると、その受信データを、ベースアドレスと書込ポインタとの加算値に従って主記憶上のサイクリックキューであるメッセージ受信キューに格納するとともに、書込ポインタの指示値を、メッセージ受信キューにおける次の空き領域の先頭アドレスに更新するように構成されている。

40

## 【 0 0 2 5 〕

このとき、割込み生成機能 7 を、期待値レジスタ，比較器および割込み生成回路から構成することができる。

ここで、期待値レジスタは、条件設定機能 5 により、並列同期待ち状態になってから他 P E 2 が今回のステップを完了するまでの間に他 P E 2 から転送されてくる予定の全メッセージパケットのデータ容量等の情報に基づいて、転送処理部の書込ポインタの指示値の期待値を設定されるものである。また、比較器は、期待値レジスタに設定された期待値と書込ポインタの実指示値とを比較するものであり、割込み生成回路は、比較器による比較結果に応じ前記割込み信号を生成するものである。

50

## 【 0 0 2 6 】

このように構成された割込み生成機能 7 を有する P E 2 では、実行中の並列プロセスが並列同期待ち状態になると、当該並列プロセスを実行する他 P E 2 からメッセージパケットを受信すると、その受信データが、ベースアドレスと書込ポインタとの加算値に従って主記憶上のサイクリックキューであるメッセージ受信キューに格納されるとともに、その書込ポインタの指示値が、メッセージ受信キューにおける次の空き領域の先頭アドレスに更新される。

## 【 0 0 2 7 】

そして、その書込ポインタの実指示値と条件設定機能 5 により期待値レジスタに設定された期待値とを比較し前記メッセージパケットの受信量が期待値に到達したことを検知したことが、比較器にて検知されると、割込み生成回路により割込み信号が生成される。つまり、メッセージパケット受信量が期待値に到達したことを、並列プロセスの割付を再開する条件（並列同期待ち状態の解除条件）として、割込み生成機能 7 を実現することができる（請求項 1, 3）。

10

## 【 0 0 2 8 】

なお、並列計算機 1 上で複数のジョブがそれぞれ並列プロセスとして実行されている場合、パケット中に、各ジョブに対応したプロセス識別子を設定し、各 P E 2 において、期待値およびベースアドレスをプロセス識別子毎に設定するとともに、メッセージ受信キューおよび書込ポインタをプロセス識別子毎にそなえ、メッセージ受信キューへのデータ書込、書込ポインタの更新、期待値の設定および割込み信号の生成をプロセス識別子毎に行なうことにより、ジョブ毎に、並列プロセスの非活性化 / 活性化を行なうことができる（請求項 2, 4）。

20

## 【 0 0 2 9 】

〔 3 〕同期検出（バリア検出）に応じた割込み生成機能

複数の P E 2 の相互間で各 P E 2 が当該並列プロセスとして今回のステップで割り付けられた処理を完了したか否かを示す 1 ビット以上の状態値を送受信するための状態通信部と、この状態通信部を通じて得られた当該並列プロセスを実行する P E 2 からの状態値が全て一致したことを検出すると同期検出信号を出力する同期検出機構とが各 P E 2 にそなえられている場合、条件設定機能 5 により、同期検出機構から同期検出信号が出力されることが前記条件として設定され、割込み生成機能 7 が同期検出機構により実現され、この同期検出機構からの同期検出信号を前記割込み信号として用いる。

30

## 【 0 0 3 0 】

このように構成された割込み生成機能 7 を有する P E 2 では、実行中の並列プロセスが並列同期待ち状態になった後、P E 2 の同期検出機構から同期検出信号が、割込み生成機能 7 の割込み信号として出力される。つまり、各 P E 2 の同期検出（バリア検出）を、並列プロセスの割付を再開する条件として、割込み生成機能 7 を実現することができる。

## 【 0 0 3 1 】

なお、並列計算機 1 上で複数のジョブがそれぞれ並列プロセスとして実行されている場合、パケット中に、各ジョブに対応したプロセス識別子を設定するとともに、割込み生成機能 7 を実現する同期検出機構をプロセス識別子毎にそなえ、同期検出機構による同期検出およびおよび割込み信号の生成をプロセス識別子毎に行なうことにより、ジョブ毎に、並列プロセスの非活性化 / 活性化を行なうことができる。

40

## 【 0 0 3 2 】

## 【 発明の実施の形態 】

以下、図面を参照して本発明の実施の形態を説明する。

（ a ）本実施形態の並列計算機の説明

図 2 は本発明の一実施形態としての並列プロセススケジューリング方法を適用された並列計算機の全体構成を示すブロック図であり、この図 2 に示すように、本実施形態の並列計算機 1 0 も、図 1 に示したものと同様、個別に処理を実行する n 台の P E （ P E 番号として # 0 ~ # n - 1 が付与されているものとする） 1 1 と、これらの P E 1 1 を相互に通信

50

可能に接続する P E 間結合網 1 2 とをそなえて構成され、ある 1 つのジョブを、 n 台の P E 1 1 のうちの 2 台以上の P E 1 1 により、ステップ毎に同期させながら並列プロセスとして実行するようになっている。

【 0 0 3 3 】

各 P E 1 1 は、図 2 の P E 番号 # 0 の P E 1 1 のブロック内に表記されるように、少なくとも転送処理部 1 3 , C P U 1 4 および主記憶 1 5 をそなえて構成されている。なお、図 2 においては、 P E 番号 # 0 の P E 1 1 についてのみその内部を図示しているが、他の P E 番号 # 1 ~ # n - 1 の P E 1 1 も、同様に構成され、後述する各種機能を有することは言うまでもない。

【 0 0 3 4 】

ここで、転送処理部 1 3 は、主記憶 1 5 上のデータの送受信処理を行なうもので、 C P U 1 4 によるデータ処理動作とは非同期に、パケット送受信により P E 間結合網 1 2 を介して他の P E 1 1 との間でデータ転送を行なうものであり、この転送処理部 1 3 ( または P E 1 1 の第 3 実施形態で後述する同期検出機構 6 1 ) に、後述する割込み生成機能 2 4 がそなえられている。この転送処理部 1 3 の詳細構成 ( 特に割込み生成機能 2 4 の構成 ) や動作については図 4 ~ 図 8 により後述する。

【 0 0 3 5 】

また、 C P U 1 4 は、主記憶 1 5 上の各種データ、プログラム等に基づいてデータ処理動作 ( 実際の並列プロセスの実行動作等 ) を行なうとともに、転送処理部 1 3 に対してデータ転送指令を行なうもので、後述する不活性化機能 2 1 , 条件設定機能 2 2 , 割付機能 2 3 および活性化機能 2 5 を有している。

ここで、不活性化機能 2 1 は、当該並列プロセスとして今回のステップで割り付けられた処理を完了し、他の P E 1 1 が当該並列プロセスとして今回のステップで割り付けられた処理を完了するのを待つ並列同期待ち状態になった際に、当該並列プロセスの割付を禁止すべく当該並列プロセスを不活性化状態にする機能である。

【 0 0 3 6 】

また、条件設定機能 2 2 は、不活性化機能 2 1 による当該並列プロセスの不活性化と同時に、当該並列プロセスの割付を再開する際に満たされるべき条件 ( 並列同期待ち状態の解除条件 ) を設定する機能であり、割付機能 2 3 は、実行可能な他のジョブが存在する場合には、当該並列プロセスが不活性化状態である期間中、その他のジョブについてのプロセスの割付を行なう機能である。

【 0 0 3 7 】

さらに、割込み生成機能 2 4 は、条件設定機能 2 2 により設定された条件が満たされた場合、現在実行中の処理に対する割込み信号を生成する機能であり、活性化機能 2 5 は、割込み生成機能 2 4 からの割込み信号に応じて、当該並列プロセスの割付を再開すべく当該並列プロセスを活性化状態にする機能である。

なお、不活性化機能 2 1 , 条件設定機能 2 2 , 割付機能 2 3 および活性化機能 2 5 は、実際には、後述するように、 O S ( O p e r a t i n g S y s t e m ) の基本的部分であるスーパーバイザにより実現される機能である。ここで、スーパーバイザとは、タイムシェアリング、入出力、マルチプログラミングなどの機能を制御すべく主記憶 1 5 に常駐しているプログラムのことで、狭義の O S である。

【 0 0 3 8 】

次に、上述のごとく構成された各 P E 1 1 の動作 ( 即ち、本実施形態の並列プロセススケジューリング方法 ) について説明する。

各 P E 1 1 において、 C P U 1 4 により実行中の並列プロセスが並列同期待ち状態になった時には、その並列同期待ち状態になった旨を示すシステムコールが前記条件を伴ってその並列プロセスから発行される。

【 0 0 3 9 】

このシステムコールを受け取ったスーパーバイザは、通知された条件が成立した時に割込み信号を割込み生成機能 2 4 により生成させるように、転送処理部 1 3 ( 割込み生成機能 2

10

20

30

40

50



4) のパラメータを設定するとともに、並列同期待ち状態になった並列プロセスを不活性状態に遷移させ、この並列プロセスがディスパッチされないようにする(以上が不活性化機能 2 1 および条件設定機能 2 2)。

【0040】

そして、CPU 1 4 は、並列同期待ち状態になった並列プロセス以外に実行可能なジョブが存在する場合には、割付機能 2 3 により、この他のジョブ(他の並列ジョブあるいは非並列ジョブ)についてのプロセスの割付が行なわれる。

この他のジョブの実行中に、条件設定機能 5 により転送処理部 1 3 (または同期検出機構 6 1) 側に設定された前記条件が成立した時には、割込み生成機能 2 4 により、現在実行中の処理に対する割込み信号が生成されて CPU 1 4 へ出力される。この割込み信号を通知されたスーパーバイザは、並列同期待ち状態であった該当並列プロセスを活性化し(活性化機能 2 5)、その並列プロセスのディスパッチを再開する。

10

【0041】

図 3 は本実施形態の作用を説明するための図で、この図 3 は、前述した図 1 2 に示すギャングスケジューリングを行なったものに対応している。この図 3 に示す例でも、ある並列ジョブが、図 1 2 に示した例と同様、5 台の PE 1 1 上で、それぞれプロセス番号 0 ~ 4 を付された 5 つのプロセスとして並列的に実行されており、各プロセスは、同期点 1, 2 で示すように、ステップ毎に同期しながら実行されている。

【0042】

図 3 中、 $t$  は 1 タイムスライスであり、“ ” は各プロセスが並列同期待ち状態になりシステムコールを発行した時点(つまり、今回のステップで割り付けられた処理を完了した時点)を示し、“ ” は各 PE 1 1 で前記条件が成立し割込み生成機能 2 4 により割込み信号が生成された時点を示している。また、横方向の太線は、プロセスが実際に実行されている期間を示している。

20

【0043】

図 3 に示す本実施形態のスケジューリング方法と、図 1 2 に示した従来のスケジューリング手法との異なる点は、図 1 2 に示した横方向の細線が無くなっている点である。つまり、従来、各プロセスは、並列同期待ち状態で実際には何ら処理を行なわない状態であっても、並列同期を検出するまで、各タイムスライスに対してディスパッチされていたが、本実施形態では、並列同期待ち状態の並列プロセスをディスパッチすることがなくなる。これにより、前記条件が成立するまでの間、新たに空いた CPU 1 4 のタイムスライスに、他のプロセスをディスパッチすることが可能になる。

30

【0044】

従って、複数の並列ジョブを同時に実行する、または、並列ジョブと非並列ジョブとを同時に実行するような多重ジョブ環境において、並列同期待ち状態にある並列プロセスをディスパッチする代わりに実行可能な他のジョブのプロセスをディスパッチするという協調スケジューリングが可能になるので、係る多重ジョブ環境にある並列計算機 1 0 のスループット性能が大幅に向上することになる。

【0045】

なお、図 3 では、本発明を、ギャングスケジューリングに適用した場合について説明したが、図 1 1 に示した一般的な並列プロセススケジューリングにも、上述と同様にして適用することが可能で、この場合も本実施形態と同様の作用効果を得ることができることは言うまでもない。

40

次に、各 PE 1 1 における割込み生成機能 2 4 を実現する 3 つの実施形態について、図 4 ~ 図 8 により詳細に説明する。

【0046】

(b) 第 1 実施形態の並列計算機用処理装置の説明

図 4 は本発明の第 1 実施形態としての並列計算機用処理装置の要部構成を示すブロック図であり、この図 4 では、転送処理部 1 3 における受信系と、この受信系に付加された割込み生成機能 2 4 とが詳細に図示されており、転送処理部 1 3 に本来そなえられている送信

50

系の図示は省略されている。

【 0 0 4 7 】

なお、図 4 中、16 は CPU 14 用の主記憶アクセス制御部で、この主記憶アクセス制御部 16 は、CPU 14 からの指令に応じ主記憶 15 に対してアクセスを行ない、主記憶 15 から CPU 14 へのデータ転送や、CPU 14 から主記憶 15 へのデータ転送を制御するものである。

また、図 4 中、30 は転送処理部 13 の一部を構成する主記憶アクセス制御部で、この主記憶アクセス制御部 30 は、転送処理部 13 の受信系および送信系からの指令に応じ主記憶 15 に対してアクセスを行ない、主記憶 15 から送信系へのデータ転送や、受信系から主記憶 15 へのデータ転送を制御するものである。さらに、この主記憶アクセス制御部 30 には、割込み生成機能 24 における各種レジスタ（後述）へのデータ設定を行なうためのアドレスデコーダ 30A としての機能もそなえられている。

図 4 に示すように、第 1 実施形態としての PE 11 の転送処理部 13 の受信系は、コマンドレジスタ 31、デコーダ 32、制御回路 33、入力バッファ 34、アドレスレジスタ 35 およびアドレス生成回路 36 から構成されている。

【 0 0 4 8 】

ここで、コマンドレジスタ 31 は、PE 間結合網 12 からのデータ受信時に、受信したパケットのヘッダ内に含まれる転送コマンド（コマンドコード）を一時的に保持するものであり、デコーダ 32 は、コマンドレジスタ 31 に保持されたコマンドコードを解析するものであり、制御回路 33 は、このデコーダ 32 の解析結果に基づいて、受信系の各部を制

【 0 0 4 9 】

入力バッファ 34 は、PE 間結合網 12 を介して送信側 PE 11 から受信したパケットを一時的に保持するもので、この入力バッファ 34 に保持されたパケットボディのデータは、図 5 に示すフローチャートに従って後述するごとく、アドレスレジスタ 35 に示されるアドレスと組になって、順次、主記憶アクセス制御部 30 を介して主記憶 15 に格納されるようになっている。

【 0 0 5 0 】

アドレスレジスタ 35 は、入力バッファ 34 に保持されているパケットボディを書き込むべき主記憶 15 上のアドレスを一時的に保持するもので、このアドレスレジスタ 35 には、まず、受信したパケットのヘッダにて指定されるアドレスデータ（パケットボディ受信アドレス）が保持され、以後、アドレス生成回路 36 により生成されたアドレスデータが保持されるようになっている。

【 0 0 5 1 】

アドレス生成回路 36 は、ヘッダ指定のアドレスデータがアドレスレジスタ 35 に設定されると、入力バッファ 34 から主記憶 15 へのデータ書込を行なう度に、アドレスレジスタ 35 に設定されたアドレス値に対して 1 回のデータ格納バイト長を加算するもので、その加算結果は、アドレスレジスタ 35 に設定されるようになっている。このアドレス生成回路 36 による加算処理は、パケットボディの受信（書込）を終了するまで行なわれる。

【 0 0 5 2 】

一方、上述のような転送処理部 13 の受信系には、割込み生成機能 24 が付加されている。第 1 実施形態としての PE 11 における割込み生成機能 24 は、図 4 に示すように、カウンタアドレスレジスタ 41、カウンタ値レジスタ 42、デクリメンタ 43、カウンタ期待値レジスタ 44、比較器 45 および割込み生成回路 46 によって実現されている。

【 0 0 5 3 】

ここで、カウンタアドレスレジスタ 41 は、受信したパケットのヘッダにて指定される受信カウンタアドレスを保持するものである。この第 1 実施形態では、後述するごとく並列同期待ち状態になった並列プロセスについて他の PE 11 から転送されてきたパケットの数を計数しているが、カウンタアドレスレジスタ 41 に保持される受信カウンタアドレスは、その計数結果（カウント値）を格納すべき主記憶 15 上のアドレスを指定するもので

10

20

30

40

50

ある。

【 0 0 5 4 】

カウンタレジスタ 4 2 は、並列同期待ち状態になった並列プロセスについてのパケットを受信する度に、主記憶 1 5 上の受信カウンタアドレス（カウンタアドレスレジスタ 4 1 に保持されたアドレス）から主記憶アクセス制御部 3 0 を介して読み出されたカウンタ値を保持するものである。

デクリメンタ 4 3 は、カウンタレジスタ 4 2 に保持されたカウンタ値から 1 を減算するもので、その減算結果は、比較器 4 5 へ出力されるとともに、主記憶アクセス制御部 3 0 を介して主記憶 1 5 上の受信カウンタアドレスに書き込まれる。このようにデクリメンタ 4 3 による減算結果を主記憶 1 5 上の受信カウンタアドレスに書き込むことにより、並列同期待ち状態になった並列プロセスについての受信パケット数に関する情報であるカウンタ値が更新される。

10

【 0 0 5 5 】

この第 1 実施形態では、上述したカウンタアドレスレジスタ 4 1 ，カウンタレジスタ 4 2 およびデクリメンタ 4 3 により、並列同期待ち状態の並列プロセスを実行する他の P E 1 1 から転送されてくるパケットの数をカウンタダウンにより計数するカウンタとしての機能が実現されている。

また、カウンタ期待値レジスタ 4 4 は、処理中の並列プロセスが並列同期待ち状態になると、前述した C P U 1 4 （スーパーバイザ）の条件設定機能 2 2 により、主記憶アクセス制御部 3 0 （アドレスデコーダ 3 0 A ）を介して所定のカウンタ期待値を設定されるものである。このカウンタ期待値は、並列同期待ち状態になってから他の P E 1 1 が今回のステップを完了するまでの間に他の P E 1 1 から転送されてくる予定のパケットの数に基づいて設定されるもので、例えば、主記憶 1 5 上の受信カウンタアドレスにおけるカウンタ値が “ 8 ” であり、並列同期待ち状態になってから受信する予定のパケット数が “ 6 ” ある場合には、所定のカウンタ期待値として “ 2 ” が設定されることになる。

20

【 0 0 5 6 】

比較器 4 5 は、カウンタ期待値レジスタ 4 4 に設定されたカウンタ期待値とデクリメンタ 4 3 の出力値（実カウンタ値）とを比較するもので、これらの値が一致した場合に一致信号〔不一致時に “ 0 ” （ L o w レベル）で一致時に “ 1 ” （ H i g h レベル）になる信号〕を出力するものである。

30

そして、割込み生成回路 4 6 は、比較器 4 5 による比較結果に応じ、C P U 1 4 （スーパーバイザ）に対して割込み信号を生成するもので、割込み保留レジスタ 4 7 ，割込み生成マスクレジスタ 4 8 および A N D ゲート 4 9 により構成されている。

【 0 0 5 7 】

ここで、割込み保留レジスタ 4 7 は、比較器 4 5 からの一致信号を保持し、その保持信号を A N D ゲート 4 9 へ出力するものである。

また、割込み生成マスクレジスタ 4 8 は、割込みを生成させるか否かを C P U 1 4 側から予め定めるためのマスク情報を設定され、その設定情報を A N D ゲート 4 9 へ出力するものである。このマスク情報としては、割込みを発生させる場合に “ 1 ” が設定される一方、割込みを発生させない場合（つまり割込み保留レジスタ 4 7 からの信号をマスクする場合）に “ 0 ” される。

40

【 0 0 5 8 】

さらに、A N D ゲート 4 9 は、割込み保留レジスタ 4 7 からの信号と割込み生成マスクレジスタ 4 8 からの信号との論理積を算出し、その結果を割込み生成信号として C P U 1 4 へ出力するものである。つまり、割込み生成マスクレジスタ 4 8 にマスク情報として “ 1 ” が設定されている場合に、比較器 4 5 からの一致信号が立ち上がって割込み保留レジスタ 4 7 に保持される信号が “ 1 ” になると、A N D ゲート 4 9 から C P U 1 4 へ出力される割込み生成信号が “ 1 ” に立ち上がり、C P U 1 4 において割込み処理が行なわれるようになっていく。

【 0 0 5 9 】

50

次に、第1実施形態としてのPE11の動作について、図5のフローチャート(ステップS1~S16)に従って説明する。ここでは、特に割込み生成機能24の動作を説明すべく、PE11が、ある並列プロセスについて並列同期待ち状態になってからその並列プロセスについてのパケットを他のPE11から受信する際の、転送処理部13の受信系および割込み生成機能24の動作について説明する。このとき、カウンタ期待値レジスタ44には、CPU14(スーパーバイザ)の条件設定機能22により、所定のカウンタ期待値が既に設定されているものとする。

**【0060】**

PE間結合網12は、このPE間結合網12に接続された各PE11の入力バッファ34の空きワード数を、常時、捕捉しており(ステップS1)、所定PE11を宛先(受信側PE)とするパケットが存在し、且つ、この受信側PE11の入力バッファ34が空いている場合には(ステップS2)、1ワード目にパケット送信開始信号を伴ってパケットの転送を開始し(ステップS3)、受信側PE11の入力バッファ34の空き状態に応じて、パケットの全体をその受信側PE11に転送する(ステップS4)。このステップS4による処理は、1つのパケットの転送を終了するまで(ステップS5でYES判定となるまで)繰り返し行なわれ、パケットの転送を終了すると、ステップS1に戻る。

10

**【0061】**

ステップS3, S4によりPE間結合網12から受信側PE11へのパケット転送が開始されると、受信側PE11の転送処理部13内の受信系では、入力バッファ34が空いている限り、パケットを読み込む(ステップS6)。このとき、パケットヘッダ内の各指定データが流れるタイミングで、各指定データが、対応するレジスタ31, 35, 41に読み込まれる。つまり、コマンドレジスタ31にはコマンドコードが、アドレスレジスタ35にはパケットボディ受信アドレスが、カウンタアドレスレジスタ41には受信カウンタアドレスが読み込まれる(ステップS7)。

20

**【0062】**

ステップS7によりコマンドレジスタ31に読み込まれたコマンドコードはデコーダ32により解読されて、パケット受信格納方法を制御するための信号が、制御回路33により生成される(ステップS8)。

そして、転送処理部13の受信系は、アドレスレジスタ35に設定されたアドレスと入力バッファ34からのパケットデータとを組にして主記憶アクセス制御部30へ送り、この主記憶アクセス制御部30を介して、パケットボディを主記憶15上の指定アドレスに格納する(ステップS9)。

30

**【0063】**

ステップS9により、パケットデータを1回格納すると、アドレス生成回路36により、アドレスレジスタ35のアドレス値に1回のデータ格納バイト長が加算され、アドレスレジスタ35に設定される(ステップS10)。

これらのステップS9およびS10による処理は、パケットボディを全て受信するまで(ステップS11でYES判定となるまで)繰り返し実行される。

**【0064】**

パケットボディを全て受信すると、転送処理部13の受信系は、カウンタアドレスレジスタ41に保持されているアドレスを主記憶アクセス制御部30に送り、主記憶15上のそのアドレスから、該当並列プロセスについてのカウント値を読み込み、カウント値レジスタ42に設定する(ステップS12)。

40

カウント値レジスタ42に設定されたカウント値はデクリメンタ43により1だけ減算された後、その減算結果(新たなカウント値)をデータとし、且つ、カウンタアドレスレジスタ41に保持されているデータをアドレスとして主記憶アクセス制御部30に送出する。これにより、デクリメンタ43による減算結果が、主記憶15上の受信カウンタアドレスに書き込まれ、並列同期待ち状態になった並列プロセスについての受信パケット数に関する情報であるカウント値が更新される(ステップS13)。

**【0065】**

50

デクリメンタ 4 3 による減算結果を主記憶 1 5 に書き込むと同時に、比較器 4 5 により、カウンタ期待値レジスタ 4 4 に設定されたカウンタ期待値とデクリメンタ 4 3 の出力値（実カウント値）とが比較され、これらの値が一致した場合（ステップ S 1 4 で Y E S 判定の場合）、比較器 4 5 からの一致信号が立ち上がり割込み保留レジスタ 4 7 に“ 1 ”が設定される（ステップ S 1 5 ）。

**【 0 0 6 6 】**

このとき、割込み生成マスクレジスタ 4 8 にマスク情報として“ 1 ”が設定されていれば、割込み生成回路 4 6（ANDゲート 4 9）から CPU 1 4 へ出力される割込み信号が“ 1 ”に立ち上がり（ステップ S 1 6）、CPU 1 4 において割込み処理が行なわれる。このステップ S 1 6 による処理を終了した後、もしくは、比較器 4 5 の比較結果が不一致であった場合（ステップ S 1 4 で N O 判定の場合）、転送処理部 1 3 の受信系は受信待機状態になる。

10

**【 0 0 6 7 】**

このように、第 1 実施形態としての P E 1 1 によれば、実行中の並列プロセスが並列同期待ち状態になると、当該並列プロセスを実行する他の P E 1 1 から自 P E 1 1 へ転送されてくるパケットの数がカウントダウンにより計数され、その実カウント値と条件設定機能 2 2 によりカウンタ期待値レジスタ 4 4 に設定された期待値とが一致したことが比較器 4 5 にて検知されると、割込み生成回路 4 6 により割込み信号が生成される。つまり、転送パケット数が期待値に到達したことを、並列プロセスの割付を再開する条件（並列同期待ち状態の解除条件）として、割込み生成機能 2 4 が実現される。

20

**【 0 0 6 8 】**

なお、上述した第 1 実施形態では、P E 1 1 の CPU 1 4 で実行される並列プロセスが 1 種類であるものとして説明しているが、第 1 実施形態では、並列計算機 1 0 上で複数のジョブがそれぞれ並列プロセスとして実行されている場合にも対応することができる。ただし、その場合、転送されるパケットのヘッダ中に、各ジョブに対応したプロセス識別子を設定する。また、割込み生成機能 2 4 をプロセス識別子毎にそなえて、カウンタ期待値をプロセス識別子毎にカウンタ期待値レジスタ 4 4 に設定するとともに、デクリメンタ 4 3 の出力値（カウント値）を格納する領域をプロセス識別子毎に主記憶 1 5 上に確保する。これにより、パケットヘッダ中のプロセス識別子に対応して、受信パケット数の計数、カウンタ期待値の設定および割込み信号の生成を行なえ、ジョブ毎に、並列プロセスの非活性化 / 活性化を行なうことができる。

30

**【 0 0 6 9 】**

また、上述した第 1 実施形態では、受信パケット数をデクリメンタ 4 3 により減算（カウントダウン）して計数しているが、逆にインクリメンタによりカウントアップして計数するようにしてもよい。この場合、当然、カウンタ期待値レジスタ 4 5 には、そのカウントアップ処理に対応した値がカウンタ期待値として設定される。例えば、主記憶 1 5 上の受信カウンタアドレスにおけるカウント値が“ 8 ”であり、並列同期待ち状態になってから受信する予定のパケット数が“ 6 ”ある場合には、所定のカウンタ期待値として“ 1 4 ”が設定されることになる。

**【 0 0 7 0 】**

（ c ）第 2 実施形態の並列計算機用処理装置の説明

図 6 は本発明の第 2 実施形態としての並列計算機用処理装置の要部構成を示すブロック図であり、この図 6 でも、第 1 実施形態と同様、転送処理部 1 3 における受信系と、この受信系に付加された割込み生成機能 2 4 とが詳細に図示されており、転送処理部 1 3 に本来そなえられている送信系の図示は省略されている。図 6 中、既述の符号と同一の符号はほぼ同一部分を示しているため、その詳細な説明は省略する。

40

**【 0 0 7 1 】**

ただし、第 2 実施形態では、並列計算機 1 0 において、プログラミングモデルとしてメッセージパッシングモデルが採用され転送処理部 1 3 によりパケットとしてメッセージパケットが転送されるものとする。

50

図6に示すように、第2実施形態としてのPE11の転送処理部13の受信系は、第1実施形態と同様のコマンドレジスタ31、デコーダ32、制御回路33および入力バッファ34を有するほか、アドレスレジスタ37、加算器38、メッセージ受信キューベースアドレスレジスタ39、書込ポインタ40、1加算器50、読出ポインタ51および比較器52を有して構成されている。

**【0072】**

ここで、入力バッファ34は、第1実施形態のものと同様、PE間結合網12を介して送信側PE11から受信した packets を一時的に保持するものであるが、この入力バッファ34に保持されたパケットボディのデータは、図7に示すフローチャートに従って後述するごとく、アドレスレジスタ37に示されるアドレスと組になって、順次、主記憶アクセス制御部30を介し主記憶15上のメッセージ受信キュー（サイクリックキュー）17に格納されるようになっている。

10

**【0073】**

アドレスレジスタ37は、入力バッファ34に保持されているパケットボディを書き込むべき主記憶15上のアドレスを一時的に保持するもので、このアドレスレジスタ37には、加算器41からの値がアドレスデータとして保持されるようになっている。

加算器38は、入力バッファ34に保持されているパケットボディを主記憶15上のメッセージ受信キュー17に格納する際の書込アドレスを順次生成するためのもので、レジスタ39に保持されるメッセージ受信キューベースアドレス（メッセージ受信キュー17の空きの先頭アドレス）と書込ポインタ40の値とを加算し、前記書込アドレスとしてアドレスレジスタ37へ出力する。

20

**【0074】**

書込ポインタ40は、初期値として0が設定されているが、主記憶15へのデータ書込を開始すると、その書込ポインタ40の値は、メッセージ受信キュー17に対してこの受信キュー17の1ブロック分のデータが書き込まれる度に、1加算器50により1ずつカウントアップされる。

従って、加算器38からの出力は、メッセージ受信キューベースアドレスを初期値として1ブロック分のデータ書込毎に1ずつ順に増加する。このような加算器38からのアドレス値は、パケットを全て書き込むまでアドレスレジスタ39に順次設定される。

**【0075】**

そして、パケットボディのデータは、アドレスレジスタ37に順次設定されるアドレスと組になって、主記憶アクセス制御部30を介して主記憶15のメッセージ受信キュー17に書き込まれる。

30

なお、読出ポインタ51は、サイクリックキューであるメッセージ受信キュー17の読出位置を示すものであり、比較器52は、1加算器50により1を加算された書込ポインタ40の値と読出ポインタ51の値とを比較し、一致した場合には、メッセージ受信キュー17でデータのオーバフローが発生したものと判断し、割込み信号を生成してCPU14に対して出力するものである。

**【0076】**

一方、上述のような転送処理部13の受信系にも、第1実施形態と同様、割込み生成機能24が付加されている。第2実施形態としてのPE11における割込み生成機能24は、図6に示すように、書込ポインタ期待値レジスタ53、比較器54および第1実施形態と同様の割込み生成回路によって実現されている。

40

ここで、書込ポインタ期待値レジスタ53は、処理中の並列プロセスが並列同期待ち状態になると、前述したCPU14（スーパーバイザ）の条件設定機能22により、主記憶アクセス制御部30（アドレスデコーダ30A）を介して所定の書込ポインタ期待値を設定されるものである。この書込ポインタ期待値は、並列同期待ち状態になってから他のPE11が今回のステップを完了するまでの間に他のPE11から転送されてくる予定の全メッセージパケットのデータ容量に基づいて設定されるもので、その全メッセージパケットを受信した場合に書込ポインタ40が示すものと考えられる値である。

50

## 【 0 0 7 7 】

比較器 5 4 は、書込ポインタ期待値レジスタ 5 3 に設定された書込ポインタ期待値と書込ポインタ 4 0 の実指示値とを比較するもので、これらの値が一致した場合に一致信号〔不一致時に “ 0 ” で一致時に “ 1 ” になる信号〕を出力するものである。なお、比較器 5 4 は、書込ポインタ期待値と書込ポインタ 4 0 の実指示値とを比較した結果、期待値と実指示値とが不一致になったこと、または、実指示値が期待値を超えたことを検知した場合に、前述のような所定信号を出力するものとして構成することもできる。

## 【 0 0 7 8 】

そして、割込み生成回路 4 6 は、第 1 実施形態のものと同様、比較器 5 4 による比較結果に応じ、CPU 1 4 (スーパーバイザ) に対して割込み信号を生成するもので、前述と全く同様の割込み保留レジスタ 4 7、割込み生成マスクレジスタ 4 8 および AND ゲート 4 9 により構成されている。

10

つまり、割込み生成マスクレジスタ 4 8 にマスク情報として “ 1 ” が設定されている場合に、比較器 5 4 から的一致信号が立ち上がって割込み保留レジスタ 4 7 に保持される信号が “ 1 ” になると、AND ゲート 4 9 から CPU 1 4 へ出力される割込み生成信号が “ 1 ” に立ち上がり、CPU 1 4 において割込み処理が行なわれるようになっている。

## 【 0 0 7 9 】

次に、第 2 実施形態としての PE 1 1 の動作について、図 7 のフローチャート (ステップ S 2 1 ~ S 3 9) に従って説明する。ここでも、特に割込み生成機能 2 4 の動作を説明すべく、PE 1 1 が、ある並列プロセスについて並列同期待ち状態になってからその並列プロセスについてのメッセージパケットを他の PE 1 1 から受信する際の、転送処理部 1 3 の受信系および割込み生成機能 2 4 の動作について説明する。このとき、書込ポインタ期待値レジスタ 5 3 には、CPU 1 4 (スーパーバイザ) の条件設定機能 2 2 により、所定の書込ポインタ期待値が既に設定されているものとする。

20

## 【 0 0 8 0 】

PE 間結合網 1 2 は、第 1 実施形態と同様、この PE 間結合網 1 2 に接続された各 PE 1 1 の入力バッファ 3 4 の空きワード数を、常時、捕捉しており (ステップ S 2 1)、所定 PE 1 1 を宛先 (受信側 PE) とするメッセージパケットが存在し、且つ、この受信側 PE 1 1 の入力バッファ 3 4 が空いている場合には (ステップ S 2 2)、1 ワード目にパケット送信開始信号を伴ってメッセージパケットの転送を開始し (ステップ S 2 3)、受信側 PE 1 1 の入力バッファ 3 4 の空き状態に応じて、メッセージパケットの全体をその受信側 PE 1 1 に転送する (ステップ S 2 4)。このステップ S 2 4 による処理は、1 つのメッセージパケットの転送を終了するまで (ステップ S 2 5 で YES 判定となるまで) 繰り返し行なわれ、メッセージパケットの転送を終了すると、ステップ S 2 1 に戻る。

30

## 【 0 0 8 1 】

ステップ S 2 3、S 2 4 により PE 間結合網 1 2 から受信側 PE 1 1 へのパケット転送が開始されると、受信側 PE 1 1 の転送処理部 1 3 内の受信系では、入力バッファ 3 4 が空いている限り、メッセージパケットを読み込む (ステップ S 2 6)。このとき、パケットヘッダ内のコマンドコードが流れるタイミングで、そのコマンドコードをコマンドレジスタ 3 1 に読み込む (ステップ S 2 7)。コマンドレジスタ 3 1 に読み込まれたコマンドコードはデコーダ 3 2 により解読されて、パケット受信格納方法を制御するための信号が、制御回路 3 3 により生成される (ステップ S 2 8)。

40

## 【 0 0 8 2 】

そして、転送処理部 1 3 の受信系では、レジスタ 3 9 に保持されるメッセージ受信キューベースアドレスと書込ポインタ 4 0 の値とを加算器 3 8 により加算した結果が、アドレスレジスタ 3 7 に書込アドレスとして設定された後 (ステップ S 2 9)、書込ポインタ 4 0 の値が、1 加算器 5 0 により 1 だけカウントアップされる (ステップ S 3 0)。

## 【 0 0 8 3 】

ステップ S 3 0 により 1 だけカウントアップされた書込ポインタ 4 0 の値 (実指示値) は、比較器 5 4 により、書込ポインタ期待値レジスタ 5 3 に設定された期待値と比較され、

50

これらの値が一致した場合（ステップS31でYES判定の場合）、比較器54からの一致信号が立ち上がり割込み保留レジスタ47に“1”が設定される（ステップS32）。

【0084】

このとき、割込み生成マスクレジスタ48にマスク情報として“1”が設定されていれば、割込み生成回路46（ANDゲート49）からCPU14へ出力される割込み信号が立ち上がり（ステップS33）、CPU14において割込み処理が行なわれ、転送処理部13の受信系は受信待機状態になる。

比較器54の比較結果が不一致であった場合（ステップS31でNO判定の場合）、比較器52により、ステップS30により1だけカウントアップされた書込ポインタ40の値（実指示値）と、読出ポインタ51の値とが比較される。そして、これらの値が一致した場合（ステップS34でYES判定の場合）には、メッセージ受信キュー17でデータのオーバフローが発生したものと判断されて、比較器52からCPU14へ出力される割込み信号が立ち上がり（ステップS35）、CPU14において割込み処理が行なわれ、転送処理部13の受信系は受信待機状態になる。

【0085】

比較器52の比較結果が不一致であった場合（ステップS34でNO判定の場合）、転送処理部13の受信系は、アドレスレジスタ37に設定されたアドレスと入力バッファ34からのパケットデータとを組にして主記憶アクセス制御部30へ送り、この主記憶アクセス制御部30を介して、パケットデータを主記憶15上のメッセージ受信キュー17内へ格納する（ステップS36）。

【0086】

ステップS36により、パケットデータを1回格納すると、図示しないアドレス生成回路によりアドレスレジスタ37のアドレス値に1回のデータ格納バイト長が加算され、その加算結果が、アドレスレジスタ37に設定される（ステップS37）。

ステップS36およびS37による処理は、メッセージ受信キュー17に対して1ブロック分のパケットデータ転送を完了するか、パケットの全てをメッセージ受信キュー17に転送するまで（ステップS38でYES判定となるまで）繰り返し実行される。

【0087】

また、上述したステップS29～S38による処理は、パケットの受信を完了するまで、つまり、パケットの全てをメッセージ受信キュー17に転送するまで（ステップS39でYES判定となるまで）繰り返し実行される。ステップS39でYES判定となると、転送処理部13の受信系は受信待機状態になる。

このように、第2実施形態としてのPE11によれば、実行中の並列プロセスが並列同期待ち状態になると、当該並列プロセスを実行する他のPE11からメッセージパケットを受信すると、その受信データが、書込ポインタ40の指示値に従って主記憶15上のメッセージ受信キュー17に格納されるとともに、その書込ポインタ40の指示値が、メッセージ受信キュー17における次の空き領域の先頭アドレスに更新される。

【0088】

そして、その書込ポインタ40の実指示値と書込ポインタ期待値レジスタに設定された期待値とが一致したことが比較器54にて検知されると、割込み生成回路46により割込み信号が生成される。つまり、メッセージパケット受信量が期待値に到達したことを、並列プロセスの割付を再開する条件（並列同期待ち状態の解除条件）として、割込み生成機能24が実現される。

【0089】

なお、上述した第2実施形態では、PE11のCPU14で実行される並列プロセスが1種類であるものとして説明しているが、この第2実施形態でも、並列計算機10上で複数のジョブがそれぞれ並列プロセスとして実行されている場合にも対応することができる。ただし、その場合、転送されるパケットのヘッダ中に、各ジョブに対応したプロセス識別子を設定する。また、割込み生成機能24をプロセス識別子毎にそなえ、書込ポインタ期待値をプロセス識別子毎に書込ポインタ期待値レジスタ53に設定し、メッセージ受信キ

10

20

30

40

50



ユーベースアドレスレジスタ39, 書込ポインタ40および読出ポインタ51等をプロセス識別子毎にそなえ、さらに、メッセージ受信キュー17を、プロセス識別子毎に主記憶15上に確保する。これにより、パケットヘッダ中のプロセス識別子に対応して、メッセージ受信キュー17へのデータ書込, 書込ポインタ40の更新, 期待値の設定および割込み信号の生成をプロセス識別子毎に行なうことにより、ジョブ毎に、並列プロセスの非活性化/活性化を行なうことができる。

【0090】

(d) 第3実施形態の並列計算機用処理装置の説明

図8は本発明の第3実施形態としての並列計算機用処理装置の要部構成を示すブロック図である。第3実施形態のPE11は、同期検出(バリア検出)に応じた割込み生成機能24を有するもので、図8に示すような同期検出機構61と前述と同様機能の割込み生成マスクレジスタ48およびANDゲート49とにより、第3実施形態の割込み生成機能24が実現されている。

10

【0091】

つまり、第3実施形態では、図8に示すように、各PE11の相互間が、状態通信部としてのBST(Barrier State)放送装置60により接続されている。このBST放送装置60は、複数のPE11の相互間で各PE11が当該並列プロセスとして今回のステップで割り付けられた処理を完了したか否かを示すバリア状態値(1ビット長の変数)を送受信するためのもので、例えば、PE番号#0, #1, #2, ..., #(n-1), #0, #1, ...の順に、各PE11のバリア状態値(後述する出力レジスタ62に保持される値)を全てのPE11に放送するものである。

20

【0092】

また、各PE11には、このBST放送装置60を通じて得られた当該並列プロセスを実行するPE11からのバリア状態値が全て一致したことを検出すると同期検出信号〔同期検出時に“1”(Highレベル)になる信号〕を出力する同期検出機構61がそなえられている。

そして、本実施形態の割込み生成機能24では、ANDゲート49により、同期検出機構61からの同期検出信号と、割込み生成マスクレジスタ48との論理積が算出され、その論理積結果が割込み信号としてCPU14に対して出力されるようになっている。

【0093】

つまり、割込み生成マスクレジスタ48にマスク情報として“1”が設定されている場合に、同期検出機構61からの同期検出信号が立ち上がると、ANDゲート49からCPU14へ出力される割込み生成信号が“1”に立ち上がり、CPU14において割込み処理が行なわれる。

30

次に、図8を参照しながら、この第3実施形態で用いられる同期検出機構61の構成および動作について説明する。

【0094】

同期検出機構61は、バリア状態値出力レジスタ62, バリア状態値入力レジスタ63, 現同期値レジスタ64, 排他的論理和ゲート65, ANDゲート66, バリアマスクレジスタ67, セレクタ68, 現在ポインタ69, 1加算器70, 同期開始PE番号ポインタ71, 比較器72, 前バリア同期値レジスタ73, NOTゲート74, ANDゲート75および否定論理比較器76から構成されている。

40

【0095】

なお、図8中、“<E”は、ラッチ(レジスタ64, 73, ポインタ71)の書込タイミングを与えるイネーブル信号の入力を示し、レジスタ48, 62, 67, 73に付された“ ”は、CPU14(プログラム)からアクセス可能であることを示すものである。ここで、バリア状態値出力レジスタ62は、自PE11のバリア状態値(BST\_\_OUT)を保持し、BST放送装置60に送出するものである。

【0096】

バリア状態値入力レジスタ63は、BST放送装置60から前述のごとく放送される各P

50

PE11のバリア状態値(BST\_\_IN)を順次受信して保持するものである。

現同期値レジスタ64は、現在の同期値(CSYNC; Current SYNChronization value)を保持するものであり、排他的論理和ゲート(EOR)65は、バリア状態値入力レジスタ63の値(BST\_\_IN)と現同期値レジスタ64の値(CSYNC)との排他的論理を算出するものである。つまり、排他的論理和ゲート65の出力値は、バリア状態値入力レジスタ63の値(BST\_\_IN)と現同期値レジスタ64の値(CSYNC)とが異なる場合に“1”になる。

【0097】

ANDゲート66は、排他的論理和ゲート65からの値とセクタ68からの値との論理積を算出して出力するものである。

バリアマスクレジスタ67は、本実施形態の並列計算機10においてPE11がn台そなえられる場合、nビット長のビット列として構成され、このバリアマスクレジスタ67のビットx(x=0~n-1)には、PE番号xのPE11からのバリア状態値を本PE11の並列同期対象とするか否かが設定される。例えば、PE番号xのPE11からのバリア状態値を本PE11の並列同期対象とする場合には“1”が、並列同期対象としない場合には“0”が設定される。

【0098】

セクタ68は、現在ポインタ69により指示されるビット位置のバリアマスク値をバリアマスクレジスタ67から選択してANDゲート66へ出力するものである。

現在ポインタ(C\_\_pointer; Current pointer)69は、現在、バリア同期値入力レジスタ63に保持されているバリア状態値を送出したPE11のPE番号を指示してセクタ68に出力するものである。この現在ポインタ69の指示値は、1マシンサイクル毎に1加算器70により1を加算される。この現在ポインタ69の初期値が適切に設定されれば、現在ポインタ69の示す値を、常に、バリア同期値入力レジスタ63からのバリア状態値に対応するPE番号に合わせることが可能である。

【0099】

同期開始PE番号ポインタ(SS\_\_pointer; Synchronization Start pointer)71は、BST放送装置60により放送されたバリア状態値が、現同期レジスタ64の値(CSYNC)と等しくなった最初のPE11のPE番号を保持するものである。つまり、“同期開始PE番号ポインタ71の指示値”~“現在ポインタ69の指示値”のPE番号をもつPE11で、且つ、この指示値範囲のPE番号のバリアマスク値が“1”であるPE11のバリア状態値(BST\_\_OUT)は、現同期値レジスタ64に保持されている値(CSYNC)と同じである。

【0100】

比較器72は、現在ポインタ69の指示値に1を加算した値と、同期開始PE番号ポインタ71の指示する値とを比較し、これらの値が一致した場合に比較結果として“1”をANDゲート75へ出力するものである。

前バリア同期値レジスタ73は、最後に同期したときのバリア状態値(LBSY; Last Barrier SYNchronization value)を保持するものである。

【0101】

そして、ANDゲート66の出力値は、現同期値レジスタ64および同期開始PE番号ポインタ71にイネーブル信号として入力される。

また、NOTゲート74は、ANDゲート66の出力値を反転するものであり、ANDゲート75は、NOTゲート74の出力値と比較器72からの比較結果との論理積を算出し、前バリア同期値レジスタ73へイネーブル信号として出力するものである。

【0102】

さらに、否定論理比較器(!比較器)76は、前バリア同期値レジスタ73に保持されている値(LBSY)とバリア状態値出力レジスタ62に保持されている値(BST\_\_OUT)とについて後述するような手順で比較を行なうことによってバリア同期検出を行ない

10

20

30

40

50

、バリア同期が検出されると、“ 1 ”を同期検出信号として出力するものである。

【 0 1 0 3 】

上述のごとく構成された同期検出機構 6 1 では、ANDゲート 6 6 からの出力値は、今回受信したバリア状態値の送信元 P E 1 1 が同期対象（セレクタ 6 8 からの出力値が “ 1 ”）であり、且つ、今回受信したバリア状態値と現在の同期値（C S Y N C）とが異なる場合に “ 1 ”になる。

そして、ANDゲート 6 6 からの出力値が “ 1 ”になったタイミングで、現同期値レジスタ 6 4 および同期開始 P E 番号ポインタ 7 1 にイネーブル信号が入力され、現同期値レジスタ 6 4 の値（C S Y N C）としてバリア状態値入力レジスタ 6 3 の値（B S T \_\_ I N）が設定されるとともに、同期開始 P E 番号ポインタ 7 1 の値として現在ポインタ 6 9 の値が設定される。これにより、このとき現在ポインタ 6 9 の指示する P E 番号が、最初に同期した P E 1 1 の P E 番号として同期開始 P E 番号ポインタ 7 1 に保持される。

10

【 0 1 0 4 】

比較器 7 2 からの信号は、前述した通り、現在ポインタ 6 9 の指示値に 1 を加算した値と同期開始 P E 番号ポインタ 7 1 の指示する値とが一致した場合、即ちバリア同期を検出したタイミングで “ 1 ” に立ち上がる。このとき、“同期開始 P E 番号ポインタ 7 1 の値” ~ “ポインタ 7 1 の値に n - 2 を加算した値” を P E 番号としてもつ P E 1 1 は同期していることになる。

【 0 1 0 5 】

また、NOTゲート 7 4 の出力値は、ANDゲート 6 6 からの出力値が “ 0 ” の場合、即ち、現在ポインタ 6 9 の指示する P E 番号の P E 1 1 からのバリア状態値が現在の同期を崩さない場合に、“ 1 ”になる。

20

従って、比較器 7 2 からの信号が “ 1 ” になり、且つ、NOTゲート 7 4 の出力値が “ 1 ” である場合、つまりANDゲート 7 5 の出力値が “ 1 ” になった時点で、同期対象の全ての P E 1 1 のバリア状態値が現同期値レジスタ 6 4 の値（C S Y N C）であることが示されたことになる。

【 0 1 0 6 】

上述のごとくANDゲート 7 5 の出力値が “ 1 ” になったタイミングで、ANDゲート 7 5 の出力値 “ 1 ” がイネーブル信号として前バリア同期値レジスタ 7 3 に与えられ、この前バリア同期値レジスタ 7 3 の値（L B S Y）として現同期値レジスタ 6 4 の値（C S Y N C）が設定される。

30

そして、否定論理比較器（！比較器）7 6 により、前バリア同期値レジスタ 7 3 の値（L B S Y）とバリア状態値出力レジスタ 6 2 の値（B S T \_\_ O U T）とが比較される。この比較器 7 6 による同期検出の手順は次の通りである。

【 0 1 0 7 】

この操作の前では、“ L B S Y ” = “ B S T \_\_ O U T ” であるとする。“ B S T \_\_ O U T ” の値を反転することにより、“ B S T \_\_ O U T ” ! = “ L B S Y ” となる。ここで、“ ! ” は C 言語での否定論理を意味しており、“ ! = ” は “ n o t e q u a l ” を意味している。同期対象の全ての P E 1 1 のバリア状態値が一致していること即ちバリア同期が検出された時、“ L B S Y ” = “ B S T \_\_ O U T ” が設定される。

40

【 0 1 0 8 】

従って、“ L B S Y ” ! = “ B S T \_\_ O U T ” の期間がバリア同期待ちの期間であり、“ L B S Y ” = “ B S T \_\_ O U T ” の期間がバリア同期期間でこのとき比較器 7 6 は同期検出信号として “ 1 ” を出力する。

そして、前述した通り、割込み生成マスクレジスタ 4 8 にマスク情報として “ 1 ” が設定されている場合に、同期検出機構 6 1（比較器 7 6）からの同期検出信号が立ち上がると、ANDゲート 4 9 から C P U 1 4 へ出力される割込み生成信号が “ 1 ” に立ち上がり、C P U 1 4 において割込み処理が行なわれる。

【 0 1 0 9 】

このように、本発明の第 3 実施形態としての P E 1 1 によれば、実行中の並列プロセスが

50

並列同期待ち状態になると、各PE11の同期検出機構61の同期検出信号が、割込み生成機能24の割込み信号として利用される。つまり、各PE11の同期検出（バリア検出）が、並列プロセスの割付を再開する条件として用いられて、割込み生成機能7が実現される。

【0110】

なお、上述した第3実施形態では、PE11のCPU14で実行される並列プロセスが1種類であるものとして説明しているが、この第3実施形態でも、並列計算機10上で複数のジョブがそれぞれ並列プロセスとして実行されている場合にも対応することができる。ただし、その場合、転送されるパケットのヘッダ中に、各ジョブに対応したプロセス識別子を設定する。また、割込み生成機能24を実現する同期検出機構61をプロセス識別子毎にそなえ、同期検出機構61による同期検出およびおおよび割込み信号の生成をプロセス識別子毎に行なうことにより、ジョブ毎に、並列プロセスの非活性化/活性化を行なうことができる。

10

【0111】

【発明の効果】

以上詳述したように、本発明の並列計算機における並列プロセススケジューリング方法および並列計算機用処理装置によれば、複数の並列ジョブを同時に実行する、または、並列ジョブと非並列ジョブとを同時に実行するような多重ジョブ環境において、並列同期待ち状態にある並列プロセスをディスパッチする代わりに実行可能な他のジョブのプロセスをディスパッチするという協調スケジューリングが可能になるので、係る多重ジョブ環境にある並列計算機のスループット性能の向上に寄与するところが大きい（請求項1～4）。

20

【0113】

また、各ジョブに対応したプロセス識別子毎に割込み生成等を行なうことにより、並列計算機上で複数のジョブがそれぞれ並列プロセスとして実行されている場合でも、各ジョブ毎に、並列プロセスの非活性化/活性化を行なうことができる（請求項2, 4）。

【図面の簡単な説明】

【図1】本発明の原理ブロック図である。

【図2】本発明の一実施形態としての並列プロセススケジューリング方法を適用された並列計算機の全体構成を示すブロック図である。

【図3】本実施形態の作用を説明するための図である。

30

【図4】本発明の第1実施形態としての並列計算機用処理装置の要部構成を示すブロック図である。

【図5】第1実施形態の並列計算機用処理装置の動作を説明するためのフローチャートである。

【図6】本発明の第2実施形態としての並列計算機用処理装置の要部構成を示すブロック図である。

【図7】第2実施形態の並列計算機用処理装置の動作を説明するためのフローチャートである。

【図8】本発明の第3実施形態としての並列計算機用処理装置の要部構成を示すブロック図である。

40

【図9】一般的な並列計算機システムの構成を示すブロック図である。

【図10】一般的な並列計算機用処理装置の構成を示すブロック図である。

【図11】一般的な並列プロセススケジューリングの例を示す図である。

【図12】並列プロセスのPE間協調スケジューリングとしてギャングスケジューリングを行なった場合の例を示す図である。

【符号の説明】

- 1 並列計算機
- 2 PE（処理装置）
- 3 通信網
- 4 不活性化機能

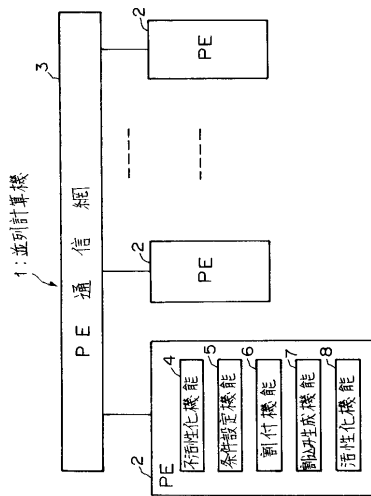
50

5	条件設定機能	
6	割付機能	
7	割込み生成機能	
8	活性化機能	
10	並列計算機	
11	PE (処理装置)	
12	PE間結合網 (通信網)	
13	転送処理部	
14	CPU	
15	主記憶	10
16	主記憶アクセス制御部	
17	メッセージ受信キュー (サイクリックキュー)	
21	不活性化機能	
22	条件設定機能	
23	割付機能	
24	割込み生成機能	
25	活性化機能	
30	主記憶アクセス制御部	
30A	アドレスデコーダ	
31	コマンドレジスタ	20
32	デコーダ	
33	制御回路	
34	入力バッファ	
35	アドレスレジスタ	
36	アドレス生成回路	
37	アドレスレジスタ	
38	加算器	
39	メッセージ受信キューベースアドレスレジスタ	
40	書込ポインタ	
41	カウンタアドレスレジスタ	30
42	カウント値レジスタ	
43	デクリメンタ	
44	カウンタ期待値レジスタ	
45	比較器	
46	割込み生成回路	
47	割込み保留レジスタ	
48	割込み生成マスクレジスタ	
49	ANDゲート	
50	1加算器	
51	読出ポインタ	40
52	比較器	
53	書込ポインタ期待値レジスタ	
54	比較器	
60	BST放送装置 (状態通信部)	
61	同期検出機構	
62	バリア状態値出力レジスタ (BST__OUT)	
63	バリア状態値入力レジスタ (BST__IN)	
64	現同期値レジスタ (CSYNC)	
65	排他的論理和ゲート (EOR)	
66	ANDゲート	50

- 67 バリアマスクレジスタ
- 68 セレクタ
- 69 現在ポインタ ( C \_ p o i n t e r )
- 70 1 加算器
- 71 同期開始 P E 番号ポインタ ( S S \_ p o i n t e r )
- 72 比較器
- 73 前バリア同期値レジスタ ( L B S Y )
- 74 NOTゲート
- 75 ANDゲート
- 76 否定論理比較器 ( ! 比較器 )

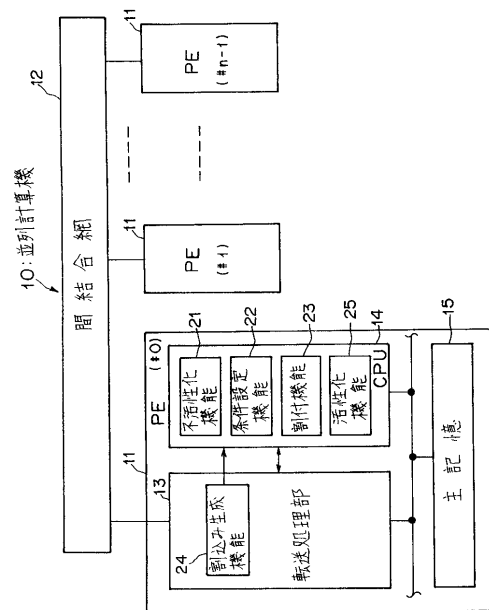
【 図 1 】

本発明の原理ブロック図



【 図 2 】

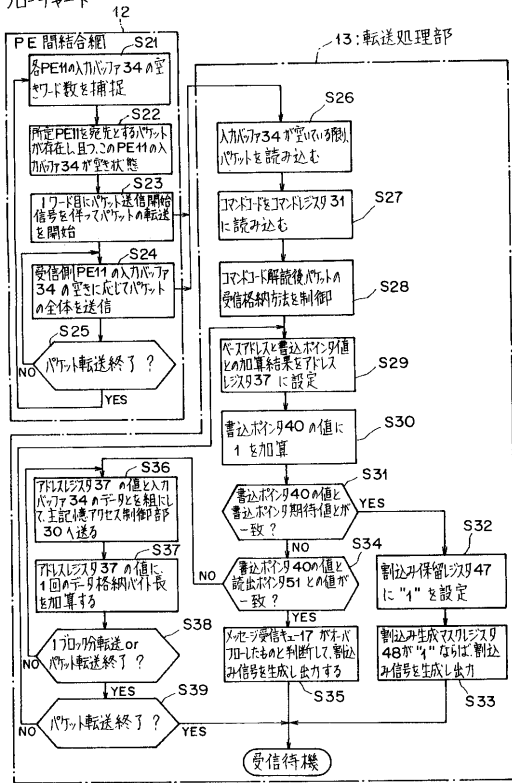
本発明の一実施形態としての並列プロセススケジューリング方法を適用された並列計算機の全体構成を示すブロック図





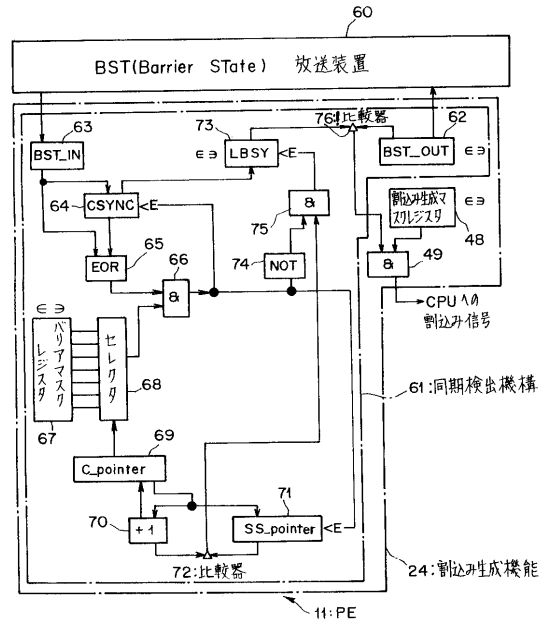
【 図 7 】

第2実施形態の並列計算機用処理装置の動作を説明するためのフローチャート



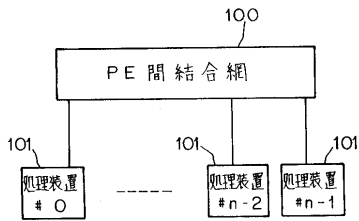
【 図 8 】

本発明の第3実施形態としての並列計算機用処理装置の要部構成を示すブロック図



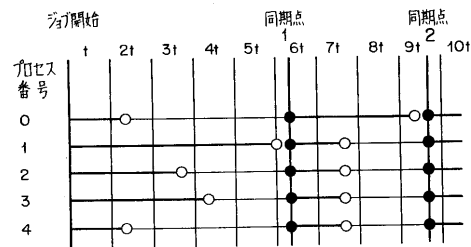
【 図 9 】

一般的な並列計算機システムの構成を示すブロック図



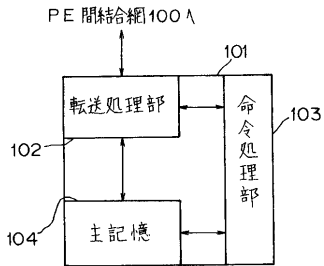
【 図 11 】

一般的な並列プロセススケジューリングの例を示す図



【 図 10 】

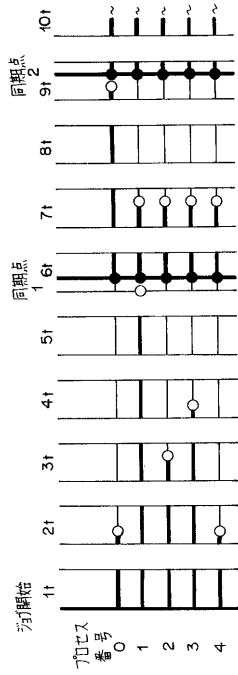
一般的な並列計算機用処理装置の構成を示すブロック図





【 図 1 2 】

並列プロセスのPE間協調スケジューリングとスキャンスケジューリングを行なった場合の例を示す図



---

フロントページの続き

- (56)参考文献 特開平03 - 144847 (JP, A)  
特開平05 - 290005 (JP, A)  
特開平06 - 187303 (JP, A)  
特開平07 - 200486 (JP, A)  
特開平07 - 134728 (JP, A)  
高木浩光外, バリア同期のためのタスクスケジューリングアルゴリズムとその性能評価, 情報処理学会研究報告, 1994年 3月11日, 第94巻, 第22号, p. 73 - 80

- (58)調査した分野(Int.Cl.<sup>7</sup>, DB名)

G06F 15/16-15/177

9/46