

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4238246号
(P4238246)

(45) 発行日 平成21年3月18日 (2009. 3. 18)

(24) 登録日 平成20年12月26日 (2008. 12. 26)

(51) Int. Cl.

F I

G 0 6 F 9/48 (2006. 01)
 G 0 6 F 9/42 (2006. 01)
 G 0 6 F 1/32 (2006. 01)
 G 0 6 F 1/04 (2006. 01)

G O 6 F 9/46 4 5 5 D
 G O 6 F 9/46 4 5 2 Z
 G O 6 F 9/42 3 3 O R
 G O 6 F 1/00 3 3 2 Z
 G O 6 F 1/04 3 O 1 C

請求項の数 6 (全 61 頁)

(21) 出願番号 特願2005-287950 (P2005-287950)
 (22) 出願日 平成17年9月30日 (2005. 9. 30)
 (62) 分割の表示 特願2003-1616 (P2003-1616)
 の分割
 原出願日 平成15年1月7日 (2003. 1. 7)
 (65) 公開番号 特開2006-99782 (P2006-99782A)
 (43) 公開日 平成18年4月13日 (2006. 4. 13)
 審査請求日 平成18年1月6日 (2006. 1. 6)
 (31) 優先権主張番号 特願2002-2816 (P2002-2816)
 (32) 優先日 平成14年1月9日 (2002. 1. 9)
 (33) 優先権主張国 日本国 (JP)

前置審査

(73) 特許権者 000005821
 パナソニック株式会社
 大阪府門真市大字門真1006番地
 (74) 代理人 100090446
 弁理士 中島 司朗
 (72) 発明者 ▲くら▼田 和司
 大阪府門真市大字門真1006番地 松下
 電器産業株式会社内
 (72) 発明者 田中 哲也
 大阪府門真市大字門真1006番地 松下
 電器産業株式会社内
 (72) 発明者 藤井 茂樹
 大阪府門真市大字門真1006番地 松下
 電器産業株式会社内

最終頁に続く

(54) 【発明の名称】 プロセッサ

(57) 【特許請求の範囲】

【請求項 1】

メモリに格納されている複数のレジスタ値群を用いて、これらレジスタ値群それぞれに対応するプログラムを逐次実行するプロセッサであって、

複数のレジスタ群と、前記複数のレジスタ群のうちいずれか1つを、プログラム実行の基礎とする実行対象レジスタ群として選択し、当該選択先を所定時間毎に切り替える選択先切り替え手段と、

前記切り替えがなされる毎に、前記複数のレジスタ群のうち前記選択がなされていない1のレジスタ群に前記レジスタ値群を復帰する復帰手段と、

前記切り替えがなされる毎に、前記復帰に先行して、当該復帰の対象となったレジスタ群にある各レジスタ値を、これらレジスタ値に対応する前記メモリ中のレジスタ値群に上書きすることにより退避する退避手段と、

前記切り替えがなされる毎に、前記実行対象レジスタ群にあるプログラムカウンタ値を含むレジスタ値群によりプログラムを特定し、前記特定したプログラムを実行するプログラム実行手段と、

プログラムの実行を中断する中断事由が発生していることを検出する中断事由発生検出手段と、

前記中断事由が解消したことを検出する中断事由解消検出手段と、

プログラムの実行をするハードウェア部分の消費電力を低減する消費電力低減手段とを備え、

10

20

前記所定時間は、所望の処理時間が異なるプログラムに応じて、プログラム実行手段を占有する時間を配分するために定められる、プログラムの実行時間枠となるタイムスロットに関連づけられたカウント値によって決まる時間であり、

さらに、前記復帰の候補となる優先順位が対応づけられた候補レジスタ値群が複数存在し、

前記復帰手段は、前記タイムスロットを所定の順番で周期的に切り替えるために、現在実行対象となっているタイムスロット及び次の実行対象とすべきタイムスロットを特定する情報を参照して、次の実行対象とすべきタイムスロットを特定した後、前記カウント値と複数の候補レジスタ値群とをタイムスロットと関連づけた時間割付情報を参照して、前記特定したタイムスロットに割り当てられているプログラムの実行時間枠を取得し、前記特定したタイムスロットに割り当てられている複数の候補レジスタ値群のうちから、

10

前記優先順位と、

時間割付情報から特定される複数の候補レジスタ値群毎に対応するプログラムそれぞれに係る中断事由の状態と

に基づいて、前記復帰の対象とするレジスタ値群を決定して前記復帰を実施し、

前記プログラム実行手段は、プログラムの実行に際し、

当該プログラムについての中断事由の発生が検出された場合、当該中断事由が解消するまで、前記特定したタイムスロットに割り当てられている当該プログラムの実行時間枠を放棄し、

当該プログラムについての中断事由の発生が検出されない場合、前記特定したタイムスロットに割り当てられている当該プログラムの実行時間枠を放棄せずに当該プログラムを実行し、

20

前記消費電力低減手段は、前記放棄が実施されている間、前記低減を実施することを特徴とするプロセッサ。

【請求項 2】

前記プログラム実行手段は、プログラム実行用のCPUを有し、前記消費電力低減手段は、前記放棄が実施されている間、前記CPUに供給されているクロック信号を遮断することにより、前記低減を実施することを特徴とする請求項 1 記載のプロセッサ。

【請求項 3】

30

前記プログラム実行手段は、プログラム実行用のCPUを有し、前記消費電力低減手段は、前記放棄が実施されている間、前記CPUに印加されている電圧を低下又は 0 にすることにより、前記低減を実施することを特徴とする請求項 1 記載のプロセッサ。

【請求項 4】

さらに、非定期的に実行すべきプログラムの実行要求を受け付ける実行要求受付手段を備え、

前記消費電力低減手段は、前記要求が受け付けられたとき、前記低減を取り止めることを特徴とする請求項 1 記載のプロセッサ。

【請求項 5】

40

前記実行要求は、割り込み処理要求であることを特徴とする請求項 4 記載のプロセッサ。

【請求項 6】

前記実行要求は、前記選択先切り替え手段により例外的な前記切り替えを伴って、緊急的にプログラムを実行するためのイベント処理要求であることを特徴とする請求項 4 記載のプロセッサ。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、プロセッサに関し、特に、効率的にプログラムを実行する技術に関する。

50

【背景技術】

【0002】

従来、複数のタスクを1つのプロセッサで擬似並行的に処理するためのOS (Operating System) がある。このOSは、以下の処理をプロセッサに実行させる。即ち、プロセッサは、タスク毎に対応するプログラムの実行位置やデータの格納位置などを示す制御情報 (以下、「コンテキスト」という。) をプロセッサ内部のレジスタ群に格納し、格納されたコンテキストにもとづいてプログラムやデータを取得してタスクを実行し、タスクの実行に伴ってコンテキストを更新し、このコンテキストをメモリに書き込むという処理を繰り返す。

【0003】

ここで、便宜的に、外部のメモリに格納されているコンテキストをレジスタ群に書き込むことを復帰といい、レジスタ群に格納されているコンテキストをこのメモリに書き込むことを退避ということとする。図32は、このようなOSの支配のもと、タスクを実行する従来のプロセッサ1001の機能ブロック図であり、上述のコンテキストを格納しているコンテキストメモリ1010も合わせて示す。

【0004】

プロセッサ1001は、タスクを実行する処理実行部1002と、処理実行部1002における処理に関する制御を実施する制御部1003と、実行するタスクのコンテキストを格納するレジスタ群1004とを有する。レジスタ群1004は、1つのタスクに対応するコンテキスト、即ち、レジスタ値群を格納するためのレジスタの集まりである。

処理実行部1002は、新たなコンテキストがレジスタ群1004に格納される毎に、このコンテキストにもとづいてプログラムやデータを取得してタスクを実行する。制御部1003は、レジスタ群1004内に既に格納されているコンテキストをコンテキストメモリ1010に上書きすることにより退避し、次にどのコンテキストをレジスタ群1004への格納対象とすべきかを判断し、この判断により格納対象とされたコンテキストをコンテキストメモリ1010から読み出してレジスタ群1004に書き込むことにより復帰する。

【0005】

したがって、処理実行部1002は、上述の退避、判断及び復帰中においては、レジスタ群1004の内容であるコンテキストを更新している最中であるので、プログラムの実行を中断せざるを得ない。図33は、プロセッサ1001におけるタスク実行の流れ及びレジスタ群1004とコンテキストメモリ1010との間におけるコンテキストの入出力の状況を示す図である。

【0006】

この図の左から右へと時間が経過するものとする。ここでは、例えば、タスクA、タスクB、タスクC及びタスクDの4つのタスクがあるものとし、タスクの実行順序は、各タスクが均等に処理されるようにタスクA、タスクB、タスクC及びタスクDの順となっているものとし、タスクDの次は、また、タスクAから実行されるものとする。

従来、各タスクの実行に際し、事前に上述の退避、判断及び復帰に実施される。つまり、各タスクを実行する上で、上述の退避、判断及び復帰がオーバーヘッドとなっている。ここで、タスクAからタスクDまでの処理を1サイクルの処理と定義づけると、上述のOSは、この1サイクルにおける、各タスクの実行順序及びタスクの実行時間、つまり、タスクのスケジュールを規定している。

【0007】

制御部1003は、このスケジュールにもとづいて、上述の判断、即ち、レジスタ群1004への格納対象とするコンテキストを決定する。以上のような処理を実施することにより、1つのプロセッサで複数のタスクを擬似並行的に処理することができる。また、上述の退避、判断及び復帰をハードウェアで実行し、複数のタスクを擬似並行的に処理するプロセッサもある。(例えば、特許文献1)

【特許文献1】特開平10-11301号公報

10

20

30

40

50

【発明の開示】

【発明が解決しようとする課題】

【0008】

しかしながら、このように1つのプロセッサで複数のタスクを擬似並行的に処理する場合、1つのタスクのみを継続的に実行する場合に比べ、タスクの実行までに上述の退避、判断及び復帰のオーバーヘッドが生じ、タスクの実行開始から次のタスクの実行開始までの時間がかかり、効率的にタスクの実行が行えないという問題がある。

特に、携帯電話などの通信機器及びTV受像機などの放送機器などにおいては、リアルタイムに近い状態でのデータ処理を実行する必要があるが、このような機器において、複数のタスクを擬似並行的に処理する場合、上述のオーバーヘッドがある従来のタスクの実行方法では、上述のようなリアルタイム近い状態でのデータ処理の実施は困難となる。

10

【0009】

そこで本発明は、上記問題点に鑑みてなされたものであり、1つのプロセッサで擬似並行的に複数のタスク、即ち、プログラムを実行する場合に、効率的にタスクを実行可能なプロセッサを提供することを目的とする。

【課題を解決するための手段】

【0010】

上記の目的を達成するために、本発明に係るプロセッサは、メモリに格納されている複数のレジスタ値群を用いて、これらレジスタ値群それぞれに対応するプログラムを逐次実行するプロセッサであって、複数のレジスタ群と、前記複数のレジスタ群のうちいずれか1つを、プログラム実行の基礎とする実行対象レジスタ群として選択し、当該選択先を所定時間毎に切り替える選択先切り替え手段と、前記切り替えがなされる毎に、前記複数のレジスタ群のうち前記選択がなされていない1のレジスタ群に前記レジスタ値群を復帰する復帰手段と、前記切り替えがなされる毎に、前記復帰に先行して、当該復帰の対象となったレジスタ群にある各レジスタ値を、これらレジスタ値に対応する前記メモリ中のレジスタ値群に上書きすることにより退避する退避手段と、前記切り替えがなされる毎に、前記実行対象レジスタ群にあるプログラムカウンタ値を含むレジスタ値群によりプログラムを特定し、前記特定したプログラムを実行するプログラム実行手段と、プログラムの実行を中断する中断事由が発生していることを検出する中断事由発生検出手段と、前記中断事由が解消したことを検出する中断事由解消検出手段と、プログラムの実行をするハードウェア部分の消費電力を低減する消費電力低減手段とを備え、前記所定時間は、複数のカウンタ値がそれぞれのプログラムに対応づけられて格納されたテーブル情報にもとづいて、前記カウンタ値により特定された時間であり、前記プログラム実行手段は、プログラムの実行に際し、当該プログラムについての中断事由の発生が検出された場合、当該中断事由が解消するまで、当該プログラムの実行のために割り当てられるべき実行期間を放棄し、前記消費電力低減手段は、前記放棄が実施されている間、前記低減を実施することを特徴とするプロセッサ。

20

30

【発明の効果】

【0011】

上記構成により、前記所定時間が複数のカウンタ値にもとづいて特定され、前記所定時間を非固定的な時間に設定することができるため、前記切り替えの間隔、つまりプログラム実行時間を非固定的に設定することができる。また、プログラムの実行に影響を及ぼすことなく、事前にレジスタ値群が実行対象レジスタ群以外のレジスタ群に復帰されるので、当該レジスタ値にもとづいてプログラムを実行しようとする際、前記復帰及び退避などのオーバーヘッドが解消されているため、速やかにプログラムが実行され得る環境となっており、プログラムがより高速に実行される。

40

【0012】

また、上述のオーバーヘッドの解消により、頻繁に前記切り替えがなされても、プログラムの実行時間には影響がなく、擬似的なリアルタイム処理の並列実行に効果的である。

さらに、実行中のプログラムにおける実行予定時間のうち、前記放棄がなされた時間に

50

において、消費電力低減手段により前記低減が実施され、省電力化が図られる。さらに、プログラムの実行中断により残された時間において、何もしないことにより、プログラムの実行周期をできる限り均一に保つ（周期性を保つ）ことができる。

【 0 0 1 3 】

また、前記プログラム実行手段は、プログラム実行用のCPUを有し、前記消費電力低減手段は、前記放棄が実施されている間、前記CPUに供給されているクロック信号を遮断することにより、前記低減を実施するとしてもよい。

これにより、容易に前記省電力化が図られる。

また、前記プログラム実行手段は、プログラム実行用のCPUを有し、前記消費電力低減手段は、前記放棄が実施されている間、前記CPUに印加されている電圧を低下又は0にすることにより、前記低減を実施するとしてもよい。

10

【 0 0 1 4 】

これにより、容易に前記省電力化が図られる。

さらに、非定常的に実行すべきプログラムの実行要求を受け付ける実行要求受付手段を備え、前記消費電力低減手段は、前記要求が受け付けられたとき、前記低減を取り止めてもよい。

これにより、緊急的に実行されるプログラム以外の実行要求は、受け付けられないため、プログラム実行手段、即ち、CPUによるプログラム実行の要否判断の処理を省くことができ、効率的なプログラム実行がなされる。

【 0 0 1 5 】

20

また、前記実行要求は、割り込み処理要求であるとしてもよい。

これにより、割り込み処理要求が発生した場合、消費電力低減手段による前記低減が取り止められる。

また、前記実行要求は、前記前記選択先切り替え手段により例外的な前記切り替えを伴って、緊急的にプログラムを実行するためのイベント処理要求であるとしてもよい。

【 0 0 1 6 】

これにより、イベント処理要求が発生した場合、消費電力低減手段による前記低減が取り止められる。

【 発明を実施するための最良の形態 】

【 0 0 1 7 】

30

< 第 1 の実施形態 >

(1 . 構成)

本発明の第 1 の実施形態について、図面を参照しながら詳細に説明する。図 1 は、本発明の第 1 の実施形態における仮想マルチプロセッサ 100 の機能ブロック図である。

仮想マルチプロセッサ 100 は、プレートの逐次切り替え実行を行うプロセッサである。ここで、プレートとは、従来のタスク又はOSのスケジューリング単位である 1 以上のタスクをまとめたものであり、プロセッサ側から見た実行対象をいう。1つのプレートは、1つの実行位置、即ち、プログラム中の命令位置を有する。

【 0 0 1 8 】

また、図 1 には、上述のプレート毎に用意されるプレートに対応するプログラムの実行位置を示すプログラムカウンタの値、上述のプログラム及びデータの格納位置を示すアドレスなどの制御情報、即ち、コンテキストが格納されているコンテキストメモリ 200 も合わせて示している。ここで、上述のプレートに対応するプログラムを実行することを、プレートの実行ということとする。

40

【 0 0 1 9 】

上述のコンテキストメモリ 200 は、SRAMにより構成され、複数のプレートの各コンテキストが格納されているコンテキスト格納領域 201 及び各プレートの実行順序を示す実行順序テーブル 300 が格納されている実行順序テーブル格納領域 202 を有する。この実行順序テーブル 300 は、1つのプレートに含まれる管理用アプリケーションプログラム（以下、「管理アプリケーション」という。）により生成されたプレートの実行ス

50

ケジュールを定めたテーブルであり、コンテキスト格納領域 201 における複数のメモリアドレス（以下、「コンテキストアドレス」という。）の値の配列であり、これらコンテキストアドレスの値により示されるそれぞれの場所には、各プレートを実行するのに必要なコンテキストが存在する。

【0020】

なお、実行順序テーブル格納領域 202 において、各コンテキストアドレスの値が格納されている物理的位置を示すアドレスを以下、テーブルアドレスということとする。上述のテーブルアドレスは、数値で示されるものであって、この数値の大小関係は、このテーブルアドレスにより特定される位置に格納されたコンテキストアドレスの上記配列における順序に対応する。

10

【0021】

つまり、スケジューリング 1 サイクルにおいては、テーブルアドレスの値が小さい順に、このテーブルアドレスの値により示される物理的位置に格納されたコンテキストアドレスの値により特定されるコンテキストアドレスと対応するプレートが実行されることとなる。このようにして、実行順序テーブル 300 に、プレートの実行順序が、A B C A D C と示されている場合、このような順序で 1 サイクル分の各プレートの実行が終了し、このサイクルが複数回繰り返されることとなる。

【0022】

各プレートの 1 回分の処理時間をいずれも同じ時間となっており、処理に時間がかかるプレートは、上述のプレート A のように、1 サイクルにおける実行回数が増えている。

20

図 34 は、仮想マルチプロセッサ 100 と外部プロセッサ 30 との接続関係を示す図である。

【0023】

上述の機能部の他に、仮想マルチプロセッサ 100 に関連し、以下の機能部がある。即ち、プログラム及びデータなどを格納し、DRAMからなる主メモリ 10 と、図示していないが、この主メモリ 10 と仮想マルチプロセッサ 100 との間に接続され、SRAMからなる上述のプログラム及びデータなどを格納するためのキャッシュメモリとが存在し、これらはバス 700 を介して仮想マルチプロセッサ 100 に接続されている。

【0024】

30

また、仮想マルチプロセッサ 100 は、複数のプレートを擬似並行的に実行するものであるが、これら複数のプレートのうちの 1 つが、外部のプロセッサに処理を依頼する場合もあり、そのための外部プロセッサが、上記と同様にバスを介して仮想マルチプロセッサ 100 に接続されている。即ち、図 34 に示すように、外部プロセッサ 30 と主メモリ 10 はバス 700 によって接続されており、仮想マルチプロセッサ 100 と外部プロセッサ 30 は、主メモリ 10 を介して接続されている。

【0025】

仮想マルチプロセッサ 100 は、上述のバス 700 を介して、主メモリ 10 に外部プロセッサ 30 が実行する処理の入力データをライトし、主メモリ 10 から外部プロセッサ 30 が実行した処理の結果であるデータをリードする。

40

また、外部プロセッサ 30 は、主メモリ 10 から処理の入力データをリードし、主メモリ 10 に処理結果であるデータをライトする。

【0026】

外部プロセッサ処理要求信号線 701 及び外部プロセッサ処理完了信号線 702 は、上述のバス 700 とは独立して設けられ、それぞれ仮想マルチプロセッサ 100 と外部プロセッサ 30 と接続する。より具体的には、上記外部プロセッサ処理要求信号線 701 は、処理要求信号を仮想マルチプロセッサ 100 から外部プロセッサ 30 へと送信するための信号系統であり、また、外部プロセッサ処理完了信号線 702 は、処理完了信号を外部プロセッサ 30 から仮想マルチプロセッサ 100 へと送信するための信号系統である。

【0027】

50

これらそれぞれの信号の伝達により、仮想マルチプロセッサ 100 から外部プロセッサへの処理の依頼と、外部プロセッサから仮想マルチプロセッサ 100 への処理完了の通知とがなされる。仮想マルチプロセッサ 100 は、SIMD (Single Instruction / MultipleData) 型のプロセッサであって、処理実行部 110 と、処理制御部 111 と、レジスタ切替部 112 と、切替入替制御部 113 と、第 1 レジスタ群 114 と、第 2 レジスタ群 115 と、サイクルカウンタ 116 と、外部 I/F 117 と、DMA 管理部 118 と、完了通知レジスタ 119 と、制御レジスタ 120 と、設定フラグレジスタ 121 とを有する。

【0028】

第 1 レジスタ群 114 及び第 2 レジスタ群 115 は、コンテキストの内容であるプログラムカウンタの値、プログラム及びデータの格納位置を示すアドレスなどのレジスタ値をそれぞれ格納するためのレジスタの集まりである。

また、各レジスタは、プレートが実行される際、このレジスタに格納されたプログラムカウンタなどの値が更新されたか否かを示す更新フラグの値を格納するための 1 ビットの記憶領域を有する。

【0029】

上述の値が更新された場合、更新フラグの値は 1 を示し、更新がなされていない場合、更新フラグの値は 0 を示す。なお、デフォルトの状態では、第 1 レジスタ群 114 及び第 2 レジスタ群 115 の各レジスタの更新フラグの値は 0 となっている。レジスタ切替部 112 は、切替入替制御部 113 からの指示にしたがい、第 1 レジスタ群 114 及び第 2 レジスタ群 115 のうちいずれか 1 つのレジスタ群をプログラム実行の際に使用するレジスタ群 (以下、「カレントレジスタ群」) として選択する。

【0030】

ここで、便宜的にカレントレジスタ群として選択されていない方のレジスタ群をスタンバイレジスタ群といい、上述の選択対象を現在選択がなされているレジスタからもう一方のレジスタに切り替えることを、レジスタ切り替えということとする。処理実行部 110 は、上述のレジスタ切り替え、つまり、カレントレジスタ群の選択がなされると、このカレントレジスタ群に格納されているコンテキストにもとづいて、実行するプログラム及びこのプログラムの実行に必要なデータの所在を特定し、DMA 管理部 118 にこれらプログラム及びデータを取得するように指示する。

【0031】

これらプログラム及びデータの取得は、キャッシュメモリからの取得を前提としており、キャッシュメモリに目的とするデータが存在する場合 (以下、「キャッシュヒット」という。)、または、例外的に後述する外部記憶装置からプログラム及びデータを取得した場合、処理実行部 110 は、これらプログラム及びデータを取得し、これらプログラム及びデータを用いて、処理制御部 111 からの指示にしたがいプレートを実行する。

【0032】

ただし、処理実行部 110 は、例外的に DMA 管理部 118 に図示されない外部記憶装置からプログラム及びデータを取得するように指示する場合もある。この外部記憶装置は、同じく図示されない他の入出力装置などによってアクセス (例えばリード、ライト) することも可能であり、複数の装置からアクセスする場合と、1 つの装置からアクセスする場合とではそのアクセスの時間が異なる (複数の装置からのアクセス時間の方が長くなる)。

【0033】

一方、キャッシュメモリに目的とするデータが存在しない場合、即ち、キャッシュミスが生じた場合、または、外部記憶装置へのアクセスに時間がかかると判断された場合、処理実行部 110 は、その旨を処理制御部 111 に通知し、処理制御部 111 からの指示にしたがって以降の処理を実施する。このプレートの実行に伴い、処理実行部 110 は、コンテキストに格納されている値でプログラムカウンタなどの更新すべき値を更新し、値を更新したレジスタに含まれる更新フラグの値を 1 にする。

10

20

30

40

50

【 0 0 3 4 】

また、処理実行部 1 1 0 は、プレートの実行中に後述の待ち要因が発生した場合、プレートをスリープさせたことを示す通知を、処理制御部 1 1 1 を介して切替入替制御部 1 1 3 に出力すると共に、設定フラグレジスタ 1 2 1 内における発生した待ち要因の種別に対応する後述の設定フラグの値を 1 に変更する。

また、処理実行部 1 1 0 は、先の待ち要因の発生したとき、外部プロセッサ 3 0 に処理を依頼する。

【 0 0 3 5 】

さらに、この待ち要因の解消となる事由が、外部プロセッサ 3 0 から処理結果を受け取ることである場合、処理実行部 1 1 0 は、外部プロセッサ 3 0 に処理を依頼してプレートをスリープさせたことを示す通知を、処理制御部 1 1 1 を介して切替入替制御部 1 1 3 に出力する。このとき、切替入替制御部 1 1 3 は、後述の外部 I / F 1 1 7 に処理要求信号を出力する。

10

【 0 0 3 6 】

なお、図 1 は、ある瞬間の状態を示す図であり、処理実行部 1 1 0 (プレート B 実行) とあるのは、第 1 レジスタ群 1 1 4 にプレート B に対応するコンテキストが格納され、さらに、この第 1 レジスタ群 1 1 4 がカレントレジスタとして選択されている時点の状態を示すものである。仮想マルチプロセッサ 1 0 0 内部でのデータの入出力と、コンテキストメモリ 2 0 0、主メモリ 1 0、キャッシュメモリ及び外部プロセッサとにおけるデータの入出力は、DMA 転送により実施され、DMA 管理部 1 1 8 は、この DMA 転送を制御する。

20

【 0 0 3 7 】

また、DMA 管理部 1 1 8 は、各機能部からの要求にもとづいて DMA 転送を実施してその転送が完了したとき、その旨を示す信号 (以下、「転送完了信号」という。) を切替入替制御部 1 1 3 に出力し、また、キャッシュメモリからプログラム及びデータを取得しようとして、キャッシュミスが生じた場合、キャッシュミスが生じたことを示す信号 (以下、「キャッシュミス信号」という。) を切替入替制御部 1 1 3 に出力する。

【 0 0 3 8 】

さらに、DMA 管理部 1 1 8 は、図示されない外部記憶装置からプログラム及びデータを取得しようとして、外部記憶装置へのアクセスに時間がかかる (レイテンシが大きい) と判断されたとき、メモリレイテンシが大きいことを示す信号 (以下、「メモリアクセス遅延信号」という) を切替入替制御部 1 1 3 に出力する。

30

外部 I / F 1 1 7 は、外部プロセッサなどの外部機器に接続するインターフェースであり、例えば、外部プロセッサに依頼した処理が完了したことを示す、外部プロセッサ処理完了信号などの外部からの信号を切替入替制御部 1 1 3 に通知する。

【 0 0 3 9 】

また、外部 I / F 1 1 7 は、例えば、外部プロセッサ 3 0 に処理を依頼することを示す処理要求信号を外部プロセッサ 3 0 に通知する。

つまり、前述したように、外部 I / F 1 1 7 は、切替入替制御部 1 1 3 から外部 I / F 1 1 7 に処理要求信号が通知されたとき、処理要求信号を外部プロセッサ 3 0 に出力する。処理制御部 1 1 1 は、プレート実行の開始から終了まで、処理実行部 1 1 0 において、継続的にプレートを実行させるように制御する。

40

【 0 0 4 0 】

サイクルカウンタ 1 1 6 は、クロックカウンタを有し、切替入替制御部 1 1 3 及び処理制御部 1 1 1 を介し、処理実行部 1 1 0 において、管理アプリケーションが含まれるプレートが実行されたことにより決定された値を、このカウント値を初期値として登録する。

また、サイクルカウンタ 1 1 6 は、上述の初期値から 1 クロック毎に 1 ずつディクリメントすることによりカウントダウンを行い、値が 0 となったとき、切替入替制御部 1 1 3 にその旨を示す信号 (以下、「カウント値 0 信号」という。) を出力すると共に、再度初期値設定を行ってカウントダウンを開始するという処理を繰り返す。

50

【 0 0 4 1 】

なお、上述のカウント値の初期値は、各プレート共通の値である。制御レジスタ 1 2 0 は、現在の実行中のプレート及び次に実行すべきプレートを特定する情報を格納するためのレジスタである。より具体的には、制御レジスタ 1 2 0 は、現在の実行中のプレートに対応するコンテキストアドレスの値が格納されている物理的位置のテーブルアドレス（以下、「カレントアドレス」という。）の値及び次に実行すべきプレートに対応するコンテキストアドレスが格納されている物理的位置のテーブルアドレス（以下、「ネクストアドレス」という。）の値を格納しており、さらに、実行順序テーブル格納領域 2 0 2 において、最も小さなテーブルアドレス（以下、「テーブル先頭アドレス」という。）の値と、最も大きなテーブルアドレス（以下、「テーブル終端アドレス」という。）の値とを格納している。

10

【 0 0 4 2 】

ところで、タスクの実行時に使用する共有資源が、他の機器により利用されており、直ちに使用できないことによる待機の発生及びいわゆる割り込みの発生などをトリガとしてソフトウェアにより、非定常的に現在実行中のタスクを中断してタスクの実行スケジュールから除外し、このタスクに割り当てられた実行時間を他のタスクに明け渡し、また、タスク中断のトリガとなった要因が解消されたときに、このタスクの実行順序を再度スケジューリングして実行することは、公知技術である。

【 0 0 4 3 】

本発明は、さらに、ハードウェアにより、プレート実行中断のトリガとなり得る所定の要因（以下、「待ち要因」という。）が発生したとき、レジスタ切り替え、即ち、プレートの切り替えを実施し、レジスタ切り替えがなされたプレートを実行順序から除外するスリープを実行し、また、この待ち要因が解消したとき、このプレートの実行を試みるウエイクアップを実施する。

20

【 0 0 4 4 】

このようなスリープ及びウエイクアップを実施する際、ウエイクアップのトリガとなる待ち要因を特定するために、プレート毎にどのような待ち要因でスリープが発生しているのかを知る必要がある。

また、外部からの割り込み要求などが到来した場合、外部からの要求に応えるべきかを事前に知る必要がある。

30

【 0 0 4 5 】

そのため、各プレートのコンテキストの中に、スリープの発生のトリガとなった待ち要因の種別、割り込み要求を受け付けるか否かなどをそれぞれ示す設定フラグの値を格納する領域を有する。このように待ち要因には複数の種別があり、これら種別毎に設定フラグ、つまり、設定フラグ群がある。

この設定フラグ群は、コンテキスト格納領域 2 0 1 に格納されている各コンテキスト内に含まれている。設定フラグレジスタ 1 2 1 は、2 つのプレートに含まれている設定フラグ群それぞれを格納するためのレジスタである。プレートの実行に際し、実行されたプレート自らがプレートの実行の中断を要求し、又は、外部からプレートの実行を中断させる要求などが到来したことなどにより、待ち要因が発生したとき、現在実行中のプレートに対応するコンテキストの複数設定フラグのうち発生した待ち要因の種別に対応する設定フラグの値が処理実行部 1 1 0 により 0 から 1 に変更される。

40

【 0 0 4 6 】

ここで、上述のプレートの実行を中断させる要求には、いわゆる割り込み要求を含んでおり、このような割り込みが発生した場合に実行中のプレートを必ずしも中断する必要はなく、割り込みを受け付けるか否かは、プレート毎に定められている。そのため、割り込みを許容するプレートにおいては、プログラム設計者により割り込み要求を受け付けるか否かを示す設定フラグの値に割り込み可能を示す値である 1 が格納されている。

【 0 0 4 7 】

完了通知レジスタ 1 1 9 は、上述の待ち要因が現在解消されているか否か、割り込み要

50

求が発生したか否かなどをそれぞれ示すフラグ（以下、「完了通知フラグ」という。）を格納するためのレジスタである。この完了通知フラグは、1つのコンテキスト中に含まれる設定フラグの数と同数存在し、各完了通知フラグは、待ち要因の種別毎に存在する各設定フラグのそれぞれと1対1に対応している。

【0048】

つまり、完了通知フラグの集まりを構成する完了通知フラグ群のビット列長は、設定フラグの集まりを構成する設定フラグ群のビット列長と同一であり、先頭からのビット位置が同じであれば同じ待ち要因についての値を格納している。切替入替制御部113は、サイクルカウンタ116からカウント値0信号を受け取ったとき、処理実行部110からプレートのスリープさせたことを示す通知を受け取ったとき、及び、DMA管理部118よりキャッシュミス信号を受け取ったとき、または、メモリアクセス遅延信号を受け取ったとき、後述のフラグ判定処理を実施して現状のネクストアドレスの値をそのままにしておくのか、それとも、他のテーブルアドレスの値を新たなネクストアドレスの値とするのかを判断してネクストアドレスの値の見直しを図りつつ、制御レジスタ120内に格納されているネクストアドレスの値を新たなカレントアドレスの値として格納し直すと共に、上述のネクストアドレスの値の次に大きなテーブルアドレスの値を新たなネクストアドレスの値として格納し、上述のレジスタ切り替えを実施するようにレジスタ切替部112に指示する。

【0049】

さらに、切替入替制御部113は、このレジスタ切り替えが実施された結果、スタンバイレジスタ群となったレジスタ群の各レジスタの更新フラグを参照し、更新フラグの値が1となっているレジスタに格納されているレジスタ値をコンテキスト格納領域201のこのレジスタ値に対応する記憶領域、即ち、レジスタ群への格納時に変更される前のレジスタ値が読み出された場所に上書きすることにより退避するようにDMA管理部118に指示すると共に、後述のフラグ判定処理を実施する。

【0050】

また、切替入替制御部113は、前記退避に伴って、当該退避されるコンテキストの設定フラグを、設定フラグレジスタ121から読み出して、コンテキスト格納領域201内のこの設定フラグに対応する記憶領域、即ち、設定フラグレジスタ121への格納時に設定フラグが読み出された場所に上書きすることにより設定フラグの値を退避する。

上述の退避及び後述のフラグ判定処理が完了すると、切替入替制御部113は、制御レジスタ120内のネクストアドレスの値により特定されるプレートのコンテキストをスタンバイレジスタ群の各レジスタに格納、即ち、復帰する。このとき、切替入替制御部113は、各レジスタの更新フラグの値を0にクリアする。

【0051】

切替入替制御部113は、上述の退避、フラグ判定処理及び復帰を、カレントレジスタ群に格納されたコンテキストを用いた処理実行部110によるプレートの実行と並行して実施する。なお、上述の退避、フラグ判定処理及び復帰に要する時間は、上述のプレートの実行時間よりも短くなるように、管理アプリケーションなどにより、上述のレジスタ切り替えのタイムスパンが定められている。

【0052】

以下、切替入替制御部113におけるフラグ判定処理の詳細について説明する。切替入替制御部113は、コンテキスト格納領域201内に存在し、制御レジスタ120のネクストアドレスの値により特定されるコンテキストに含まれる設定フラグ群のビット列のうち、1の値が格納されているビット（以下、「有効ビット」という。）の位置を把握する。

【0053】

このような有効ビットの位置を把握する処理を有効ビット位置検索処理ということとする。このとき、設定フラグ群のビット列のうち、有効ビットが1つもない場合、切替入替制御部113は、ネクストアドレスの値をそのままにしてフラグ判定処理を終了する。

設定フラグ群のビット列のうち、有効ビットがある場合、切替入替制御部 113 は、完了通知レジスタ 119 内の完了通知フラグ群のビット列における上述の有効ビットの位置に対応する位置のみを参照し、この有効ビットの位置に対応する位置全てに 1 の値が格納されている場合は、ネクストアドレスの値をそのままにしてフラグ判定処理を終了する。

【0054】

なお、有効ビットの位置に対応する位置とは、設定フラグ群のビット列の先頭位置から有効ビットまでのビット数と同じビット数分、完了通知フラグ群のビット列の先頭位置から離れた位置（以下、「有効ビット対応位置」という。）のことである。一方、完了通知フラグ群の有効ビット対応位置のいずれかに 0 の値が格納されている場合、切替入替制御部 113 は、実行順序テーブル格納領域 202 に示されている現時点におけるネクストアド
10
ドレスの値につづき、次に大きな値のテーブルアドレスの値を暫定的に新たなネクストアドレスの値とする。

【0055】

そして、切替入替制御部 113 は、この暫定的なネクストアドレスの値により特定されるコンテキストに含まれている設定フラグについて、上述の有効ビット位置検索処理を実施した後、完了通知フラグ群の有効ビット対応位置の値に応じてネクストアドレスを見直す上述の処理を、フラグ判定処理が終了するまで繰り返し実施する。

なお、現状のネクストアドレスの値がテーブル終端アドレスの値である場合、ネクストアドレスの値につづき、次に大きな値のテーブルアドレスの値とは、テーブル先頭アドレ
20
スの値とする。

（2．データ）

以下、仮想マルチプロセッサ 100 において用いられるデータについて説明する。

【0056】

図 2 は、コンテキストメモリ 200 に格納されている実行順序テーブル 300 及びコンテキストのデータ構成を示す図である。実行順序テーブル 300 は、コンテキストが格納されているコンテキスト格納領域 201 内のコンテキストアドレスの値が複数示されており、これらコンテキストアドレスの値の物理的な格納位置、即ち、テーブルアドレスが各
コンテキストアドレスにより特定されるプレートの実行順序を決定している。

【0057】

つまり、小さな値のテーブルアドレスが割り当てられている場所に格納されているコン
30
テキストアドレスほどこのコンテキストアドレスに対応するプレートの実行順序が先順位となっている。上述のフラグ判定処理により、プレートの実行順序が飛ばされることがあるものの、基本的にはコンテキストアドレスの値の物理的な格納位置、即ち、配列により示される順序に従ってプレートが実行され、テーブル最終アドレスがカレントアドレスとなっている場合における、カレントアドレスにより特定されるプレートが実行された後は、テーブル先頭アドレスをカレントアドレスとすることにより特定されるプレートが実行される。

【0058】

つまり、コンテキストアドレスの物理的な格納位置、即ち、テーブルアドレスは、1 サイ
40
クルのプレートの実行順序を示す。図 2 において、テーブルアドレスは、2000 から 2005 までの値を有しており、この場合、テーブル先頭アドレスの値は、2000 となり、テーブル終端アドレスの値は、2005 となる。

小さい値のテーブルアドレスに格納されているコンテキストアドレスのプレートほど実行順序が先となる。図 3 は、完了通知フラグ群及び設定フラグ群の各フラグについて説明する図である。完了通知フラグ群は、強制待ちフラグ、キューの空き通知フラグ、割り込み通知フラグ、ソフトウェアの待ち指示フラグ、外部プロセッサ処理完了通知フラグ、転送通知フラグなどを有する。

【0059】

強制待ちフラグは、0 の値が固定的に格納されているフラグである。キューの空き通知
50
フラグは、プレートの実行中、容量的に空きが少ないキューに命令などをキュー登録しよ

うとした場合など、キューを管理する外部機器からキューに空きがないことを示す通知が到来したか否かを示すフラグであり、キューに空きがないことを示す通知がなされた場合、0の値をとり、キューに空きがある状態では、1の値をとる。

【0060】

割り込み通知フラグは、外部機器から割り込み要求が到来したか否かを示すフラグであり、割り込み要求が到来していない場合、デフォルト値である1の値をとり、また、割り込み要求が到来した場合、切替入替制御部113により0の値にされる。ソフトウェアの待ち指示フラグは、アプリケーションソフトなどからプレートの実行を待たせる指示が到来したか否かを示すフラグであり、上述の指示が到来していない場合、デフォルト値である1の値をとり、上述の指示が到来した場合、切替入替制御部113により0の値にされる。

10

【0061】

外部プロセッサ処理完了通知フラグは、プレートの実行時に外部のプロセッサに処理を依頼した場合、依頼した処理が完了したことを示す通知が到来したか否かを示すフラグであり、すなわち、キャッシュミス信号あるいはメモリアクセス遅延信号を発生させる要因となったDMA転送や、プログラムによって要求されるDMA転送などの完了を示すフラグであり、この通知が到来していない場合、デフォルト値である0の値をとり、また、依頼した処理が完了したことを示す通知が到来した場合、切替入替制御部113により、1の値にされる。

【0062】

20

転送通知フラグは、DMA転送が完了したことを示す通知が到来したか否かを示すフラグであり、この通知が到来していない場合、デフォルト値である0の値をとり、また、DMA転送が完了したことを示す通知が到来した場合、切替入替制御部113により、1の値にされる。なお、これら上述のフラグそれぞれは、そのフラグをトリガとして、新たなプレートが実行された場合、そのプレートの実行によりそのフラグの値が、切替入替制御部113により上述のデフォルト値にリ復帰されることとなる。

【0063】

設定フラグ群を構成する各フラグは、完了通知フラグ群の各フラグに対応するフラグであって、対応するフラグを無視するか否かを示す。コンテキスト中に含まれる設定フラグ群を構成する設定フラグの値が0の場合、対応する完了通知フラグの値に影響されことなく、このコンテキストにもとづいてプレートが実行される。

30

また、設定フラグの値に1がある場合、対応する完了通知フラグの値により、この設定フラグを含むコンテキストに対応するプレートを実行対象とするか否かが切替入替制御部113により判断される。なお、設定フラグの値に1が格納される例としては、以前この設定フラグを含むコンテキストに対応するプレートが実行されたとき、待ち要因が発生し、プレートの実行が中断されたなどが挙げられ、このようなプレートは、待ち要因が解消されるまで、スリープ状態となる。

【0064】

より具体的には、対応するものどうしの完了通知フラグの値及び設定フラグの値が1の場合、この設定フラグを含むコンテキストに対応するプレートが実行対象とされ、対応するものどうしの完了通知フラグの値が0で、設定フラグの値が1の場合、このコンテキストに対応するプレートが非実行対象とされ、即ち、プレートの実行順序から除外される。

40

つまり、対応する完了通知フラグの値及び設定フラグの値が1の場合、この設定フラグを含むコンテキストに対応するプレートが切替入替制御部113によりウエイクアップされ、対応する完了通知フラグの値が0で、設定フラグの値が1の場合、このプレートのスリープ状態となる。

【0065】

また、設定フラグの値が0の場合、この0の値が格納されている場所の対応する種別の待ち要因については切替入替制御部113より無視がなされ、この待ち要因に対応する完了通知フラグも参照されない。

50

図4は、タスクとプレートとの関係を示す図である。ソフトウェア側には、複数のタスクが存在し、そのうち、リアルタイムで処理するタスクは、OSを介さず直接、管理アプリケーションの管理下に置かれている。なお、実行順序テーブル300の内容を更新するための管理アプリケーションが、上述のリアルタイムで処理するタスクに含まれており、このタスク、即ち、プレートが実行されるとき、例えば、新たな処理を実行すべき状態となっていれば、処理実行部110により、実行順序テーブル300の内容が更新されることとなる。

【0066】

これにより、1サイクル分プレートが実行される毎に、少なくとも1回はタスクのスケジューリングを見直す機会が発生する。より具体的には、管理アプリケーションに対応するプレートが実行される毎に、前記条件の発生の有無を判断し、この条件が発生した場合に、新たな内容の実行順序テーブルであるダミーテーブルを実行順序テーブル格納領域202内に生成し、プレート実行の基礎とする実行順序テーブルを現状の実行順序テーブル300からダミーテーブルへと切り替えることにより更新する。

【0067】

一方、リアルタイムで処理する必要のないタスクは、まず、OSの管理下に置かれ、さらに管理アプリケーションの管理下に置かれている。プレートは、ハードウェア側、即ち、処理実行部110における1実行単位であり、1つのタスクそのものである場合と、複数のタスクをまとめたものである場合とがあり、これらのプレートの割り振りは、プログラミングにより自由に決定可能である。

(3.動作)

以下、仮想マルチプロセッサ100において、1つのプロセッサにより擬似並行的にプレートを実行する際、効率的なプレートの実行を実現する処理について説明する。

【0068】

図5は、仮想マルチプロセッサ100において実施される処理を示すフローチャートである。切替入替制御部113は、コンテキストメモリ200内の実行順序テーブル300を参照し、制御レジスタ120内に、テーブル先頭アドレス、カレントアドレス、ネクストアドレス及びテーブル終端アドレスを登録する(ステップS500)。

第1レジスタ群114及び第2レジスタ群115内の全ての値をクリアする(ステップS501)。切替入替制御部113は、第1レジスタ群114及び第2レジスタ群115のうちカレントレジスタ群及スタンバイレジスタ群を選択する(ステップS502)。

【0069】

そして、切替入替制御部113は、カレントアドレスにより特定されるコンテキストをカレントレジスタ群に格納させる(ステップS503)。つづいて、切替入替制御部113は、処理制御部111を介して処理実行部110にプレート実行の指示を与え、その結果、処理実行部110は、カレントレジスタ群に格納されているコンテキストにもとづいて、プレートの実行を開始する(ステップS504)。

【0070】

サイクルカウンタ116は、プレート実行開始と共に、プレートの実行により設定される初期値からカウントダウンを開始する(ステップS505)。切替入替制御部113は、外部I/F117及びDMA管理部118からの信号入力を受付け、入力された信号に応じて完了通知レジスタ119の完了通知フラグの値を更新する(ステップS506)。

そして、切替入替制御部113は、フラグ判定処理を実施して、制御レジスタ120内のネクストアドレスの値を確定する(ステップS507)。つづいて、切替入替制御部113は、スタンバイレジスタ群を構成する各レジスタの更新フラグの値が1となっているコンテキスト部分のみをコンテキスト格納領域201に退避させ(ステップS508)、スタンバイレジスタ群に格納されている値を全てクリアする(ステップS509)。

【0071】

さらに、切替入替制御部113は、制御レジスタ120内のネクストアドレスにより特定されるコンテキストをスタンバイレジスタ群に格納させる(ステップS510)。そし

10

20

30

40

50

て、切替入替制御部 113 は、制御レジスタ 120 内のネクストアドレスをカレントアドレスとして登録し直し、ネクストアドレスの次のアドレスを新たなネクストアドレスとして登録し直す（ステップ S511）。

【0072】

切替入替制御部 113 は、サイクルカウンタ 116 よりカウント値 0 信号が入力された、即ち、プレート実行の終了が指示されたか又は処理実行部 110 より実行中のプレートをスリープさせたことを示す通知があったかどうか、さらに、その通知が外部プロセッサ 30 に処理を依頼してからプレートをスリープさせたことを示す通知であったかどうかを判断し（ステップ S512）、これら信号の入力又は通知がなされた場合、処理実行部 110 にプレートの実行を終了させ、さらに、外部プロセッサ 30 に処理を依頼してからプレート
10
プレートをスリープさせた場合は、切替入替制御部 113 から外部 I/F 117 に処理要求信号を通知させ、外部 I/F 117 から外部プロセッサ 30 に処理要求信号を出力させ（ステップ S513）、レジスタ切替部 112 にレジスタ切り替えを実施させ（ステップ S514）、プレートの実行を開始するステップ S504 に戻る。

【0073】

一方、これら信号の入力又は通知がなされない場合、現在のプレートの実行を継続し、終了又はスリープさせたことを示す通知があったかどうかを判断するステップ S512 に戻る。図 6 は、フラグ判定処理を示すフローチャートである。切替入替制御部 113 は、コンテキスト格納領域 201 内に存在し、制御レジスタ 120 のネクストアドレスにより
20
特定されるコンテキストに含まれる設定フラグ群のビット列のうち、1 の値が格納されている有効ビットの位置を把握する有効ビット位置検索処理を実施する（ステップ S600）。

【0074】

そして、切替入替制御部 113 は、有効ビットが 1 つも存在しないかどうかを判断し（ステップ S601）、有効ビットが 1 つも存在しない場合、制御レジスタ 120 におけるネクストアドレスの値をそのままにしてフラグ判定処理を終了する。一方、有効ビットが 1 以上存在する場合、切替入替制御部 113 は、完了通知レジスタ 119 内の完了通知フラグ群のビット列における上述の有効ビットの位置のみを参照し（ステップ S602）、この有効ビット全てに 1 の値が格納されているか否かを判断する（ステップ S603）。

【0075】

完了通知フラグ群の有効ビット全てに 1 の値が格納されている場合、切替入替制御部 113 は、ネクストアドレスの値をそのままにしてフラグ判定処理を終了する（ステップ S604）。一方、完了通知フラグ群のビット列における有効ビットのいずれかに 0 の値が格納されている場合、切替入替制御部 113 は、実行順序テーブル格納領域 202 内に示されている現時点におけるネクストアドレスの次のアドレスを暫定的に新たなネクストアドレスとし（ステップ S604）、有効ビット位置検索処理を実施するステップ S600
30
に戻る。

【0076】

以上の処理を実施することにより、仮想マルチプロセッサ 100 は、プレートの終了から次のプレートの開始までの間に、コンテキストの退避、判断及び復帰に要する時間を介在させることなくプレートを実行することができる。図 7 は、このような状態を時系列的に説明する図である。コンテキストの退避、判断及び復帰は、プレートの実行と並行して
40
実施されている。

【0077】

このため、プレートの終了から次のプレートの開始までの間に、コンテキストの退避、判断及び復帰に要する時間が介在することが回避されている。以上のように、本実施形態によれば、1 つの仮想マルチプロセッサ 100 において、擬似並行的に複数のプレートを実行しても、コンテキストの退避、判断及び復帰は、プレートの実行と並行して実施するため、プレートの実行時間にコンテキストの退避、判断及び復帰に要する時間が介在することがないので、効率的にタスクを実行することができる。
50

【 0 0 7 8 】

なお、本実施の形態では、この実行順序テーブル 3 0 0 は、プレートの実行順序を示し、この実行順序テーブル 3 0 0 内におけるコンテキストアドレスの値の物理的格納位置、即ち、配列により基本的なプレートの実行順序が定められているが、この方法に限らず、例えば、実行順序テーブル 3 0 0 内にコンテキストアドレスの値とテーブルアドレスの値との組を格納し、このテーブルアドレスの値が次の実行順序に対応するコンテキストアドレスの物理的な格納位置を示すことにより、つまり、このテーブルアドレスの値をポインタとして用いることにより、プレートの実行順序を定めるとしてもよい。

【 0 0 7 9 】

その場合、格納されているコンテキストアドレスの値の物理的な格納位置に実行順序が左右されず、また、このポインタを変更することにより、容易に実行順序の変更を行うことができる。図 8 は、このようなポインタを用いる場合の実行順序を示すテーブルのデータ構成を示す図であり、コンテキストの格納位置との対応関係を示すために、コンテキスト格納領域 2 0 1 におけるデータ構成も合わせて示している。

10

【 0 0 8 0 】

この実行順序テーブルは、上述のコンテキストアドレスと次の実行順序を指し示すポインタとしてのテーブルアドレスとの組を格納している。

また、本実施の形態において、切替入替制御部 1 1 3 は、フラグ判定処理を、実行順序テーブル 3 0 0 により示される実行順序にもとづいて実施するものとしたことにより、スリープされたプレートがウェイクアップするタイミングは、実行順序テーブル 3 0 0 に規定されたそのプレートの実行順序が巡ってきたときとなるが、フラグ判定処理に際して、スリープしているプレートのコンテキスト内の設定フラグ群を優先的に参照して、フラグ判定処理を実施し、ウェイクアップの条件が揃えば、スリープしているプレートの実行順序を待たずに実行するとしてもよい。

20

【 0 0 8 1 】

図 9 は、このような場合におけるプレート実行状況を時系列的に示す図である。t 0 において、プレート A は、外部プロセッサ 3 0 に処理を依頼し、外部のプロセッサ 3 0 から処理結果を受け取るまでスリープに入るいわゆる突き放しを実施している。

ここで、この突き放しについて説明する。まず、仮想マルチプロセッサは、主メモリ 1 0 に外部プロセッサ 3 0 が実施する処理の入力データをライトした後、処理要求信号を出力し外部プロセッサ 3 0 に処理を要求すると共に、プレートをスリープさせる。外部プロセッサ 3 0 は、主メモリ 1 0 から命令をフェッチし動作すると共に、主メモリから処理の入力データをリードして要求された処理を行い、その処理結果であるデータを主メモリにライトする。

30

【 0 0 8 2 】

その後、外部プロセッサ 3 0 は、処理完了信号を出力し、仮想マルチプロセッサ 1 0 0 に処理が完了した旨を通知する。処理の完了が通知されると、図 5、図 6 において先述したフラグ判定処理などにより、仮想マルチプロセッサ 1 0 0 は、スリープしたプレートを再度実行し、主メモリ 1 0 から外部プロセッサ 3 0 が処理した結果であるデータをリードする。

40

【 0 0 8 3 】

このように、プレートを実行することにより、仮想マルチプロセッサ 1 0 0 は、外部のプロセッサに処理を依頼し、外部のプロセッサから処理結果を受け取るまでスリープする。以下、具体的に動作を説明する。まず、t 0 では、設定フラグ群における、外部プロセッサ処理完了通知フラグの値が 1 に変更される。

t 2 から t 4 までにおいて、プレート A について優先的にフラグ判定処理を実施するが、完了通知レジスタ 1 1 9 内の有効ビット、即ち、外部プロセッサ処理完了通知フラグの値は 0 であるためウェイクアップされない。プレート D の実行期間中である t 5 において、外部プロセッサ処理完了通知フラグの値が 1 となっているためウェイクアップされる。

【 0 0 8 4 】

50

このウエイクアップに伴って、切替入替制御部 113 は、ウエイクアップされたプレートのコンテキストをスタンバイレジスタ群に格納する。そして、t6において、切替入替制御部 113 は、レジスタ切替部 112 にレジスタ切り替えを実施させる。そして、処理実行部 110 は、カレントレジスタに格納されたプレート A に対応するコンテキストにもとづいて、プレート A を実行する。

【0085】

なお、この後、再度プレート A が実行されるのは、プレート A の正規の実行順序が巡ってきたときとなる。

また、図 9 においては、スリープしているプレートが 1 つの場合を示したが、複数のプレートがスリープしている場合も考えられ、このような場合、早くスリープされたプレートのコンテキストに対して優先的にフラグ判定処理を実施してもよく、逆に、遅くスリープされたプレートのコンテキストに対して優先的にフラグ判定処理を実施してもよい。

【0086】

図 10 は、遅くスリープされたプレートのコンテキストに対して優先的にフラグ判定処理を実施する例を時系列的に示す図である。プレート B につづきプレート D がスリープ状態となっており、図中における 5 回目の判定において、プレート B 及びプレート D が実行可能な状態と判定され、この時点においてスリープしている時間が最も短いプレート D が優先的にウエイクアップされ、これにつづいてプレート B がウエイクアップされている。

【0087】

これは、先にスリープされたプレートの実行の順序は、後にスリープされたプレートよりも近づいてきているため、先にスリープされたプレートをウエイクアップして実行してしまうと、直ぐに、またこのプレートの実行順序が巡ってきてしまうため、この時点を描えれば、プレートの実行開始の間隔が極端に狭くなり、プレートの実行時間が均等化されない可能性があるためである。

【0088】

また、図 9 及び図 10 を用いて説明した上述のフラグ判定処理において、プレートのスリープ時間が決められた時間を超える場合、上述の優先を取り止めるとしてもよい。

また、本実施の形態では、完了通知フラグ群及び設定フラグ群は、強制待ちフラグ、キューの空き通知フラグ、割り込み通知フラグ、ソフトウエアの待ち指示フラグ、外部プロセッサ処理完了通知フラグ、転送通知フラグなどを有するとしたが、これ以外のフラグを設けてもよい。

【0089】

例えば、複数のプレートがスリープしている場合、種々の完了通知フラグがどのプレートに対応するものなのかが判らなくなる。そこで、プレートを識別するための番号が対応づけられているフラグを、完了通知フラグ群及び設定フラグ群に設け、スリープが生じたとき、処理実行部 110 から実行中のプレートを特定するためのユニークな番号を発生させ、この番号を設定フラグ群内に新たに設けたフラグに登録すると共に、この番号をスリープの発生のトリガとなった外部機器に対し通知する。

【0090】

待ち要因が解消されたとき、例えば、外部プロセッサなどの外部機器により、外部プロセッサ処理管理信号などと共に、上述の番号が合わせて通知されることにより、外部プロセッサ処理完了通知フラグが 1 とされるのに加え、この番号を示すフラグも 1 とされるので、どのプレートに対応するものかを判定することが可能となる。

また、本実施の形態では、待ち要因が発生した場合、実行中のプレートに割り当てられている実行時間を放棄して他のプレートに明け渡す例を示したが、これによらず、プレートの実行時間を他のプレートに明け渡さずに、上述の放棄がなされている時間分だけ、プロセッサの駆動周波数を下げ、電力消費を抑えるとしてもよい。

【0091】

その場合、プレートの実行に割り当てられた絶対時間が変わらないようにするため、例えば、プロセッサの駆動周波数、即ち、クロック周波数を 2 分の 1 に低減した場合、サイク

10

20

30

40

50

ルカウンタ 116 は、1クロックあたり1づつディクリメントしていたものを、1クロックあたり2づつディクリメントすることとなる。本実施の形態では、退避、フラグ判定処理及び復帰に要する時間は、上述のプレートの実行時間よりも短くなるように、管理アプリケーションなどにより、上述のレジスタ切り替えのタイムスパンが定められているとしたが、これはこのタイムスパンの下限値を決定する制約条件である。

【0092】

一方、このタイムスパンの上限値を決める制約条件について、本実施の形態では、特筆していないが、このタイムスパンは、プレートを実行することができる限界の時間であり、各プレートの実質的な実行時間が、この限界の時間に対して極めて短い時間で処理を実施していた場合、各プレートの実行に際して、処理が行われていない無駄な時間が生じていることとなり、このような無駄を避けるために、管理アプリケーションは、プレートの実質的な実行時間と上述のタイムスパンがほぼ同様の時間で、かつ、上述のタイムスパンが各プレートの実質的な実行時間以上となるようにタイムスパンの上限値を決定してもよい。

10

【0093】

ところで、通信処理などのリアルタイムな処理では、プレートの実行により処理すべきデータの到来する周期がほぼ決まっている。つまり、1サイクルにおける各プレートの処理負荷が決まっており、上述のように決められた上限値及び下限値を有するタイムスパンを用いて、1サイクル分の各プレートが実行されたとしても、次のサイクルで処理すべきデータが到来していなかったのでは意味がない。

20

【0094】

そこで、処理実行部 110 における処理能力が、1サイクルで処理すべきプレートの処理負荷に勝って処理時間に余裕が生じている場合、このような余裕が生じなくなる程度に、処理実行部 110 におけるプロセッサの駆動周波数、即ち、処理能力を低減し、各プレートの実質的な実行時間を大きくするようにしてもよい。

その場合、各プレートの実行に際して、処理が行われていない無駄な時間が削減され、かつ、プロセッサの駆動周波数を下げることにより消費電力も低減される。

【0095】

また、現在実行中のプレートの次に実行すべきプレート、即ち、スタンバイレジスタ群に格納されているコンテキストに対応するプレートの実行に必要なデータを事前にキャッシュに格納しておくとしてもよい。

30

その場合、スタンバイレジスタ群に格納されているコンテキストを事前に参照し、次のプレートの実行に必要なデータを把握し、このデータをキャッシュに格納する機能部が必要となる。

【0096】

また、メインメモリの容量を物理的な容量よりも仮想的に大きくするために、「プログラムによって指定される論理アドレス」と「プロセッサが物理的にアクセスする物理アドレス」との対応関係（マッピング）を管理する「仮想メモリ機能部」を本実施の形態に付与してもよい。

なお、この仮想メモリ機能は公知技術である。この場合、キャッシュには、論理アドレスと物理アドレスとの対応関係（マッピング）を示す「アドレス変換テーブル」を格納する「アドレス変換テーブルキャッシュ」が含まれ、この「アドレス変換テーブルキャッシュ」に、次に実行すべきプレートに必要なデータを示す論理アドレスを物理アドレスに変換できる「アドレス変換テーブル」を格納しておくとしてもよい。

40

【0097】

その場合、現在実行中のプレートの次に実行すべきプレート、即ち、スタンバイレジスタ群に格納されているコンテキストを事前に参照し、次のプレートの実行に必要なデータを把握し、さらに、そのデータの論理アドレスを物理アドレスに変換できるアドレス変換テーブルの値を把握して、それをアドレス変換テーブルキャッシュに格納する機能部が必要となる。

50

【 0 0 9 8 】

また、サイクルカウンタ 1 1 6 におけるカウント値の初期値は、各プレート共通の値であるとしたが、プレート毎に異なっているとしてもよい。つまり、この初期値は、プレートの実行のために用意される時間を示し、プレートの処理の内容に応じてプレート固有の実行時間を割り当ててもよい。その場合、図 1 1 に示すように、実行順序テーブル格納領域 2 0 2 内にコンテキストアドレスに対応づけられてカウント値の初期値が格納される。

【 0 0 9 9 】

そして、切替入替制御部 1 1 3 は、制御レジスタ 1 2 0 内のカレントアドレス及びネクストアドレスそれぞれに上述のカウント値の初期値を対応づけて格納する。

また、これに伴い、サイクルカウンタ 1 1 6 は、オーバーヘッドを解消しつつ随時異なる初期値にもとづいてカウントダウンを実施するために、2 つ設けることが望ましい。

また、サイクルカウンタ 1 1 6 は、図 1 2 に示す機能部構成とすることもできる。この図において、サイクルカウンタ 1 1 6 は、入出力部 3 0 1 と、第 1 メモリ 3 0 2 と、第 2 メモリ 3 0 3 と、カウントダウン部 3 0 4 とを有する。入出力部 3 0 1 は、切替入替制御部 1 1 3 からカウント値の初期値を受信し、受信した上述の初期値を交互に第 1 メモリ 3 0 1 及び第 2 メモリ 3 0 3 に格納する機能を有する。

【 0 1 0 0 】

また、入出力部 3 0 1 は、カウントダウン部 3 0 4 からの信号を受け取ったとき、切替入替制御部 1 1 3 にカウント値 0 信号を出力する。第 1 メモリ 3 0 2 及び第 2 メモリ 3 0 3 は、F I F O メモリである。カウントダウン部 3 0 4 は、第 1 メモリ 3 0 2 及び第 2 メモリ 3 0 3 から交互に初期値を読み出してその初期値にもとづきカウントダウンを実施する。

【 0 1 0 1 】

以上のような構成により、プレートの処理の内容に応じてプレート固有の実行時間を割り当てることができる。

また、処理実行部 1 1 0 は、N 段のパイプライン処理によりプレートを実行するとしてもよい。図 1 3 は、4 段のパイプライン処理を実施する例を示す。

この場合、1 段目で、プレート A の処理の実行が終了しても、2 段目から 4 段目までは、まだプレート A が実行中の状態である。つまり、この状態は、カレントレジスタ群のレジスタ値が必要であるため、プレート A のレジスタ値の退避はできない。1 段目のプレート A の処理の実行が終了してから 3 クロック経過したとき、4 段目のプレート A の実行が終了するので、ここでプレート A のレジスタ値が退避可能となる。

【 0 1 0 2 】

よって、例えば、N 段のパイプライン処理によりプレートが実行される場合は、1 段目のプレートの実行が終了してから少なくとも (N - 1) クロック後でなければ、レジスタ値、即ち、コンテキストを退避できないこととなる。

また、本実施の形態では、一旦プレートがスリープすると、ウエイクアップする機会が、実行順序テーブル 3 0 0 に示された、次の実行順序のみであったが、待ち要因が発生してスリープしても、この待ち要因が瞬時に解消される場合もあるため、これを考慮し、スリープとなった直後から 1 回分のプレート実行までの間については、スリープしたプレートの実行順序を最先の順序とし、このスリープしたプレートを優先的に実行させてもよい。図 1 4 は、このようなプレート実行がなされた場合における、プレート実行状況を説明する図である。

【 0 1 0 3 】

この図において、プレート C がスリープした直後にスリープの原因となった待ち要因が解消されており、プレート D の実行中は、プレート C の実行順序を最先の順位にしているので、プレート D が実行された後にプレート C が実行されている。ただし、この場合、待ち要因の解消タイミングが、プレート D の実行終了間際であった場合、本実施の形態においては、スタンバイレジスタの数は、1 つのみであるため、本来プレート D の後に実施されるはずであるプレート A の復帰の時間のマージンを見込まなければならない。

【 0 1 0 4 】

そのため、スリーブしたプレートの実行順序を最優先とする時間（以下、「優先期間」という。）は、この図の場合、プレートDの実行時間よりも短くなっている。このような不都合を解消すべく、レジスタ群を少なくとも3つ設け、このうちの1つをカレントレジスタ群、また、残りの2つをスタンバイレジスタ群とすることにより、図14におけるプレートCのコンテキストと、プレートAのコンテキストとを同時に、2つのスタンバイレジスタ群に格納することにより、上述の優先期間は、プレートDの実行終了間際まで長くなる。

【 0 1 0 5 】

また、第1の実施形態では、プレートの数は、レジスタ群の数を上回っていることを前提として説明していたが、これに限らず、プレートの数がレジスタ群の数以下であってもよい。その場合、コンテキストメモリ内のコンテキスト格納領域を備えなくても良く、退避、復帰のステップが不要となる。

具体的には、図5におけるS508（退避）、S509（スタンバイレジスタ群のクリア）が不要となり、S510とS514の処理が変更される。つまり、第1の実施形態では、S510（スタンバイレジスタ群への格納）は、S507のフラグ判定処理で決定される制御レジスタ120内のネクストアドレスにより特定されるコンテキストをスタンバイレジスタ群に格納させる処理であったが、コンテキスト格納領域が無い場合は、S510を「S507で判断されるネクストアドレスにより特定されるコンテキストにあらかじめ対応づけられているレジスタ群」を「次のカレントレジスタ群」として切替入替制御部113に選択し記憶させる処理とすればよい。

【 0 1 0 6 】

そして、S514で、切替入替制御部113が先ほど「次のカレントレジスタ群」として選択したレジスタ群がカレントレジスタ群になるように、レジスタ切替部112はレジスタ切り替えを実施すればよい。この場合、切替入替制御部113は、上記のネクストアドレスにより特定されるコンテキストに対応付けられているレジスタ群がどれであることを把握して記憶する機能を、レジスタ切替部112は、切替入替制御部113が記憶したレジスタ群にカレントレジスタ群を切り替える機能を、それぞれ有することとなる。

【 0 1 0 7 】

ここで、プレート及びレジスタ群の数が2つの場合、より具体的には、プレートAは第1レジスタ群、プレートBは第2レジスタ群に対応付けられているとする。例えば、切替入替制御部113は、プレートAを実行するときは、コンテキストメモリからレジスタ値群を復帰する動作なしで、第1レジスタ群をカレントレジスタ群として選択する。

また、レジスタ群及びプレート（プレートA、B、Cがあり、プレートCは第3レジスタ群に対応付けられているものとする。）の数が3つの場合であっても、カレントレジスタ群として選択されているのが実行対象レジスタ群であって、それ以外がスタンバイレジスタ群となる。プレートCを実行する際、第3レジスタ群をカレントレジスタ群として選択される。

【 0 1 0 8 】

選択先を切り替えた後も、コンテキストメモリにレジスタ値群を退避する動作は不要である。コンテキスト格納領域を備えない上記の構成でも、複数のプログラムが所定時間毎に頻繁に切り替えられながら高速に実行されるので、擬似的なリアルタイム処理の並列実行に効果的である。

また、本第1の実施形態では、コンテキストメモリ200は、SRAMにより構成されたとしたが、DRAMであってもよく、さらに、これらメモリに限らず、高速にアクセスされるレジスタや低速にアクセスされる外部記憶装置であってもよい。

【 0 1 0 9 】

また、本第1の実施形態では、管理アプリケーションが1つのプレートに含まれているとしたが、複数のプレートに含まれているとしてもよい。

< 第2の実施形態 >

10

20

30

40

50

(1 . 構成)

第 2 の実施形態について、図面を参照しながら詳細に説明する。

【 0 1 1 0 】

図 1 5 は、第 2 の実施形態における仮想マルチプロセッサ 4 0 0 の機能ブロック図である。仮想マルチプロセッサ 4 0 0 は、第 1 の実施形態における仮想マルチプロセッサ 1 0 0 と同様に、複数のプログラムを擬似並行的に実行するプロセッサである。

この仮想マルチプロセッサ 4 0 0 は、複数のタイムスロットそれぞれに割り当てられたプログラムを逐次実行することにより、複数のプログラムを擬似並行的に実行する。ここで、タイムスロットとは、プログラムを時分割して実行する場合における、最小のプログラム実行時間枠のことである。

10

【 0 1 1 1 】

第 1 の実施形態における仮想マルチプロセッサ 1 0 0 では、プレートそのもの、即ち、プログラムの実行順序が規定されているが、第 2 の実施形態の仮想マルチプロセッサ 4 0 0 では、まず、タイムスロットの順序が規定され、さらに、タイムスロット毎に実行されるプログラムの候補が用意されている点で第 1 の実施形態とは異なる。

より具体的には、各タイムスロットには、自タイムスロットへの割り当て候補となる 4 つ又は 1 つのプログラムが対応づけられおり、通常、これらのプログラムのうちいずれか 1 つが前記割り当ての対象となる。原則的には、規定されたタイムスロットの順序にもとづいて各タイムスロットに割り当てられているプログラムが逐次実行される。

【 0 1 1 2 】

20

また、仮想マルチプロセッサ 4 0 0 は、前記割り当ての対象となるプログラムが実行不可能な状態となった場合における処理の内容が、第 1 の実施形態における仮想マルチプロセッサ 1 0 0 とは異なる。なお、プロセッサ側からみた実行対象となるプログラムのことを、便宜上、ロジカルプロセッサ (以下、 「 L P 」 という。) という。

図 1 5 には、プログラムの実行位置を示すプログラムカウンタの値、プログラム及びデータの格納位置を示すアドレスなどの制御情報、即ち、コンテキストが格納されているコンテキストメモリ 5 0 0 も合わせて示している。このコンテキストメモリ 5 0 0 は、S R A M により構成され、各プログラムに対応するコンテキストが格納されているコンテキスト格納領域 5 0 1 を有する。

【 0 1 1 3 】

30

コンテキスト格納領域 5 0 1 は、コンテキストメモリ 5 0 0 内の決められた記憶領域にあって、各プログラムに対応するコンテキストの値が格納されている。さらに、図 1 5 には、基準信号を生成する発振器 6 0 0 と、この基準信号にもとづいて仮想マルチプロセッサ 4 0 0 に供給するクロックパルスを生成する P L L 6 0 1 とが示されている。

また、これらの機能部の他に、仮想マルチプロセッサ 4 0 0 に関連し、図示していない以下の機能部がある。即ち、プログラム及びデータなどを格納し、DRAMからなる主メモリと、S R A M からなる上述のプログラム及びデータなどを格納するためのキャッシュメモリとが、バスを介して仮想マルチプロセッサ 4 0 0 に接続されている。

【 0 1 1 4 】

仮想マルチプロセッサ 4 0 0 は、時間割付情報格納レジスタ 4 0 2 と、処理実行部 4 1 0 と、処理制御部 4 1 1 と、レジスタ切替部 4 1 2 と、スケジューリング制御部 4 1 3 と、第 1 レジスタ群 4 1 4 と、第 2 レジスタ群 4 1 5 と、サイクルカウンタ 4 1 6 と、外部 I / F 4 1 7 と、D M A 管理部 4 1 8 と、リクエスト検出部 4 1 9 と、制御レジスタ 4 2 0 と、設定フラグレジスタ 4 2 1 と、L P 状態格納レジスタ 4 2 4 と、プロセッサ状態格納部 4 2 5 と、クロックマスク部 4 2 7 とを有する。

40

【 0 1 1 5 】

時間割付情報格納レジスタ 4 0 2 は、タイムスロット及びプログラムに関する情報が格納されている。より具体的には、この情報は、タイムスロットのプログラム実行時間枠と実行順序などを規定する 1) タイムスロット情報と、各タイムスロットにおいて実行されるプログラムの候補を特定するための 2) L P 情報と、プログラムの候補の範囲を限定す

50

るための３）優先度＆モード情報と、実行可能なプログラムの候補がタイムスロットに存在しない場合における処理の内容を決定するための４）アイドルオプション情報と、実行中のプログラムが中断した場合における処理の内容を決定するための５）タイムオプション情報とからなる。

【０１１６】

タイムスロット毎に、これら５種類の情報が存在し、タイムスロット単位で互いに関連付けられている。第１レジスタ群４１４及び第２レジスタ群４１５は、コンテキストの内容であるプログラムカウンタの値、プログラム及びデータの格納位置を示すアドレスなどのレジスタ値をそれぞれ格納するためのレジスタの集まりである。

レジスタ切り替部４１２は、スケジューリング制御部４１３からの指示にしたがい、第１レジスタ群４１４及び第２レジスタ群４１５のうちいずれか１つのレジスタ群をプログラム実行の際に使用するレジスタ群、即ち、カレントレジスタ群として選択する。第１の実施形態と同様に、この選択先の切り替えを、レジスタ切り替えといい、カレントレジスタ群として選択されていない方のレジスタ群をスタンバイレジスタ群ということとする。

【０１１７】

ちなみに、このレジスタ切り替えのことを、コンテキストスイッチングと呼ぶ場合もある。処理実行部４１０は、プログラム実行するためのＣＰＵなどを有し、上述のレジスタ切り替え、つまり、カレントレジスタ群の選択がなされると、このカレントレジスタ群に格納されているコンテキストにもとづいて、実行するプログラム及びこのプログラムの実行に必要なデータの所在を特定し、ＤＭＡ管理部４１８にこれらプログラム及びこのプログラムの実行に必要なデータを取得するように指示する。

【０１１８】

そして、処理実行部４１０は、プログラムの実行に必要なデータが取得されたとき、処理制御部４１１からの指示にしたがいプログラムの実行を開始する。

また、処理実行部４１０は、プログラムの実行中に、処理が続行できない状況となった場合、プログラムの実行を中断する旨を示すウエイトコードを処理制御部４１１に出力する。

【０１１９】

なお、上述のＣＰＵは、クロックマスク部４２７を介しＰＬＬ６０１から出力されるクロックパルスに同期して駆動しており、このクロックパルスが途絶えた場合、プログラム実行も停止する。ＬＰ状態格納レジスタ４２４は、各プログラムの状態及び属性などを示すプログラム状態情報と各タイムスロットの設定内容が示されたタイムスロット設定情報を格納するためのレジスタである。

【０１２０】

より具体的には、このタイムスロット設定情報は、タイムスロットを特定するためのＩＤ（以下、「ＴＳ＿ＩＤ」という。）と、このタイムスロットの属性を示すコードとからなる。ここで、タイムスロットの属性のコードとして、そのタイムスロットが、定期的に行われることを原則とするプログラムの実行用として用意されたタイムドリブン型タイムスロットである場合には「ＴＤ」が、また、イベントが発生した場合にのみ実行するような非定期的なプログラム実行のために用意されたイベントドリブン型タイムスロットである場合には「ＥＤ」が示されている。

【０１２１】

つまり、タイムドリブン型タイムスロットは、イベントが発生しなくても、実行対象となり得るが、イベントドリブン型タイムスロットは、イベントが発生したときでなくては、実行対象とはなり得ない。上述のプログラム状態情報には、プログラムを特定するＩＤの値と、プログラムの状態を特定するコードと、このプログラムの属性を示すコードと、プログラムを起床又は回復させるタイミング（以下、「起床タイミング」という。）を示すコードとが示されている。

【０１２２】

プログラムの状態を特定するコードとしては、waiting、ready、running、suspended-w

10

20

30

40

50

aiting、suspended-ready、non-existentの6つがある。このうち、ready及びrunningは、現在、プログラムが実行できる状況であり、特に、runningは、現在プログラム実行中であることを示す。

また、waiting及びsuspended-waitingは、現在、プログラムが実行できない状態にあることを示す。

【0123】

non-existentは、現在、プログラムを実行する必要がない状態にあることを示す。つまり、これらwaiting、suspended-waiting及びnon-existentの状態であるプログラムは、プログラム実行のスケジューリングの対象範囲外となる。

また、プログラム実行中にプログラム実行ができない状態となり、これらコードがsuspended-waitingへと遷移した場合、その時点において割り当てられていたタイムスロットのTS_IDの値がこれらコードに付されることとなる。

【0124】

なお、これらのコードは、プログラム実行が可能となった時点で、それぞれsuspended-readyへと遷移するが、プログラムの実行がなされるまで上述のTS_IDの値が付されたままとなる。プログラムの属性を示すコードとしては、そのプログラムが、定常的に実行されることを原則とするプログラムである場合には「TD」が、また、イベントが発生した場合にのみに実行されるプログラムである場合には「ED」が示されている。

【0125】

起床タイミングを示すコードとしては、プログラムの実行要求が発生したときにスタンバイレジスタ群となっているレジスタ群が開放された後、コンテキストを格納して実行するプログラム、つまり、処理時間の制約が少ないプログラムの場合には、「n」が示され、また、既にスタンバイレジスタ群内に格納したコンテキストを入れ替えてまで早急に実行を必要とするプログラムの場合、「e」のコードが示されている。

【0126】

制御レジスタ420は、現在実行対象となっているタイムスロット及び次の実行対象とすべきタイムスロットを特定する情報を格納するためのレジスタである。より具体的には、制御レジスタ420は、現在実行対象となっているタイムスロットに対応するタイムスロット情報が格納されている場所を示すカレントアドレスの値と、次の実行対象とすべきタイムスロットに対応するタイムスロット情報が格納されている場所を示すネクストアドレスの値を格納している。

【0127】

仮想マルチプロセッサ400においては、16個タイムスロットが設定されており、これに伴い、タイムスロット情報の数も16存在する。上記16個のタイムスロットのうち、プログラムの実行に通常用いられるタイムスロットは、4つのみである。なお、この4つのタイムスロットに対応するタイムスロット情報のアドレスの値が、連続した値をとるように、タイムスロット情報が時間割付情報格納レジスタ402に格納されている。

【0128】

制御レジスタ420は、上述の連続したアドレス値における、その値が最も小さい先頭アドレスの値及びその値が最も大きい終端アドレスの値も格納している。各タイムスロット情報は、各タイムスロットを特定するためのTS_IDの値(0から15のいずれか)を含んでいる。

このTS_IDの値が小さいほど、時間割付情報格納レジスタ402において、タイムスロット情報が格納されている場所のアドレスの値も小さい。プログラムの実行に通常用いられる上記4つタイムスロットには、0から3までのTS_IDの値が対応づけられている。つまり、上述の先頭アドレスには、TS_IDの値0が示されたタイムスロット情報が格納されており、また、上述の終端アドレスには、TS_IDの値3が示されたタイムスロット情報が格納されている。

【0129】

上記4つタイムスロットのそれぞれには、優劣関係があり、先頭アドレスに格納されて

10

20

30

40

50

いるタイムスロット情報により特定されるタイムスロット(TS_ID=0)が最も優先度が高く、終端アドレスに向かってアドレスの値が大きくなるほど、そのアドレスに格納されているタイムスロットの優先度が低くなる。このタイムスロットの優先度とは、リアルタイム処理が必要とされる度合いを示し、特に、終端アドレスに格納されているタイムスロット情報(TS_ID=3)により特定されるタイムスロットにおいては、リアルタイム処理を保証しない。

【0130】

プロセッサ状態格納部425は、現在実行中のプログラムの状態に関する情報を格納するレジスタである。この情報は、1)現在においてレジスタ切替部412による切り替えが禁止されているか否かを示す切替禁止フラグの値と、2)現在、後述の共通割り込処理が実行されているか否かを示す共通割り込フラグの値とからなる。

10

前記切り替えが禁止及び前記共通割り込みが生じている場合には、それぞれのフラグの値は、1となり、それ以外の場合には0の値をとる。設定フラグレジスタ421は、プログラムに関する設定情報を格納するためのレジスタである。より具体的には、この設定情報とは、プログラム毎に用意されているフラグの集まり(以下、「設定フラグ群」という。)であり、これら設定フラグ群の各フラグの値が示す内容は、例えば、自プログラムが受付可能な固有割り込み処理要求又はイベント処理要求の種別などである。

【0131】

リクエスト検出部419は、プログラム毎に用意されているフラグの集まり(以下、「固有フラグ群」という。)と、各プログラムに共通して用意されているフラグの集まり(以下、「共通フラグ群」という。)を有する。この固有フラグ群の各フラグは、設定フラグレジスタ421に格納されている設定フラグ群の各フラグと1対1に対応しており、固有フラグ群を構成するビットデータのビット列長は、設定フラグ群を構成するビットデータのビット列長と同一であり、先頭からのビット位置が同じであれば同じ項目についての値を示している。

20

【0132】

なお、これらのフラグには、各フラグが示す項目に関連する情報、例えば、この項目に対応するプログラムを特定するIDなどの関連情報が付されているものがある。リクエスト検出部419は、外部I/F417から受け取った外部信号のビットデータの配列及び値にもとづいて、上述の固有フラグ群及び共通フラグ群の値を更新する。

30

つまり、リクエスト検出部419は、受け取った外部信号を理解可能なデータ形式へと変換する。リクエスト検出部419は、更新された共通フラグ群の各フラグの値、及び、設定フラグレジスタ421内の設定フラグ群の値と固有フラグ群の値とを照合した結果にもとづいて、この外部信号の内容を把握する。

【0133】

このような処理を、便宜的に、外部信号解読処理という。外部信号の内容としては、大別すると、割り込み処理要求とイベント処理要求の2つがある。割り込み処理要求には、さらに、固有割り込み処理要求と共通割り込み処理要求とがある。

ここで、固有割り込み処理要求とは、特定のプログラムのみにおいて実行可能な処理の実行要求であって、この特定のプログラムのみが固有割り込み処理の実行を必要としている。

40

【0134】

また、共通割り込み処理要求とは、全てのプログラムにおいて実行可能な処理の実行要求であって、これらプログラムが共通してこの共通割り込み処理の実行を必要としている。

上述の共通フラグ群は、この共通割り込み処理要求の内容について示すものである。なお、これら割り込み処理は、実行可能なプログラム上からサブルーチンプログラムにジャンプして実行される。イベント処理要求とは、実行できない状態となっているプログラム又は現在実行の必要がないプログラムにおいて、実行可能な状態となるイベント又は実行を必要とするイベントが発生したために生じるこれらプログラムの実行要求である。

50

【 0 1 3 5 】

より具体的には、リクエスト検出部 4 1 9 は、上述の外部信号解読処理の実行により、受け取った外部信号が、固有割り込み処理要求、共通割り込み処理要求及びイベント処理要求のいずれを示すのかを判別する。上記判別結果が、固有割り込み処理要求であった場合、リクエスト検出部 4 1 9 は、上述の関連情報により、この固有割り込み処理の割り込み先となるプログラムを特定する I D (以下、「固有 P __ I D」という。)及び固有割り込み処理を実行するサブルーチンプログラムを特定する I D (以下、「固有 S P __ I D」という。)も取得する。

【 0 1 3 6 】

また、上記判別結果が、共通割り込み処理要求であった場合、リクエスト検出部 4 1 9 は、上述の関連情報により、上述の外部信号から共通割り込み処理を実行するサブルーチンプログラムを特定する I D (以下、「共通 S P __ I D」という。)を取得する。

また、上記判別結果が、イベント処理要求であった場合、リクエスト検出部 4 1 9 は、上述の関連情報により、上述の外部信号から実行要求がなされたプログラムを特定する I D (以下、「イベント P __ I D」という。)を取得する。

【 0 1 3 7 】

リクエスト検出部 4 1 9 は、上述の取得に伴い、取得したこれら I D をスケジューリング制御部 4 1 3 に出力する。

また、リクエスト検出部 4 1 9 は、上述の固有割り込み処理要求、共通割り込み処理要求及びイベント処理要求に応じる処理の実行が完了した旨をスケジューリング制御部 4 1 3 から受け取ったとき、完了した処理に該当する項目のフラグの値をデフォルト値に変更する。

【 0 1 3 8 】

さらに、リクエスト検出部 4 1 9 は、上記判別結果が、イベント処理要求であった場合、先に述べたイベント P __ I D のスケジューリング制御部 4 1 3 への出力に先行して、L P 状態格納レジスタ 4 2 4 を参照する。このとき、このイベント P __ I D により特定されるプログラムの属性が「E D」を示す場合、つまり、このプログラムが非定常的に実行されるイベントドリブン型のプログラムである場合、リクエスト検出部 4 1 9 は、後述するサイクル先頭信号の直近の受信時点から現在までにおいて、プログラムの属性が「E D」であるプログラムを示す P __ I D をスケジューリング制御部 4 1 3 に出力したことがない場合に限り、この P __ I D をスケジューリング制御部 4 1 3 に出力するものとする。

【 0 1 3 9 】

リクエスト検出部 4 1 9 は、以上の処理を実施した結果、P __ I D の出力できなかったものについて、この P __ I D を記憶し、次にサイクル先頭信号を受信したときに、スケジューリング制御部 4 1 3 に出力するようにする。スケジューリング制御部 4 1 3 は、1 タイムスロット分のプログラム実行時間枠内において、プログラム実行のスケジューリングを実施し、コンテキストをスタンバイレジスタ群に復帰するための処理(以下、「復帰処理」という。)を実施し、レジスタ切り替えのための処理(以下、「レジスタ切り替え処理」という。)を実施し、スタンバイレジスタ群からコンテキストをコンテキストメモリ 5 0 0 上に退避するための処理(以下、「退避処理」という。)を実施する機能を有する。(1 - 1 . スケジューリング)まず、上述のスケジューリングについて説明する。

【 0 1 4 0 】

スケジューリングには、定常的な処理の実行についてのスケジューリングである 1) タイムドリブンスケジューリング、現在スケジューリングの対象外となっているプログラムの実行についてのスケジューリングである 2) イベントドリブンスケジューリング及びプログラムの実行中にプログラムの実行時間が放棄された場合におけるスケジューリングである 3) 放棄スケジューリングの 3 つがある。

【 0 1 4 1 】

スケジューリング制御部 4 1 3 は、これら 3 つのスケジューリング及びレジスタ切り替え処理の予備的処理として、(1) 目的タイムスロット検索処理、(2) 目的 T S 割り当

10

20

30

40

50

てプログラム検索処理、(3)優先度&モード情報検索処理、(4)アイドルオプション情報検索処理及び(5)タイムオプション情報検索処理を実施する。

(1-1-1. 目的タイムスロット検索処理)

以下、目的タイムスロット検索処理について説明する。

【0142】

スケジューリング制御部413は、制御レジスタ420内に格納されているカレントアドレスの値、ネクストアドレスの値、先頭アドレスの値及び終端アドレスの値のうち、目的とするいずれか1つのアドレス(以下、「目的アドレス」という。)の値を参照し、時間割付情報格納レジスタ402内の目的アドレスに格納されているタイムスロット情報内のTS_IDの値を参照し、当該タイムスロット(以下、「目的TS」という。)を特定し、さらに、このタイムスロットに割り当てられているプログラム実行時間枠、即ち、基本サイクルの値を取得する。

10

(1-1-2. 目的TS割り当てプログラム検索処理)

以下、目的TS割り当てプログラム検索処理について説明する。

【0143】

スケジューリング制御部413は、目的TSに対応づけられているLP情報を参照し、このLP情報が示す4つのプログラムを特定し、さらに、これらプログラムそれぞれに対応づけられている優先度(0から3のいずれか)を把握する。

(1-1-3. 優先度&モード情報検索処理) 以下、優先度&モード情報検索処理について説明する。

20

【0144】

スケジューリング制御部413は、目的TSに対応づけられている優先度&モード情報を参照し、優先度として0から3のいずれが示されているのか、さらに、モードとしてIndexモード及びLevelモードのいずれが示されているのかを把握する。上述のIndexモード及びLevelモードとは、タイムスロットへのプログラムの割り付け方を規定する設定情報であり、優先度&モード情報にIndexモードが示されている場合、ある1つのプログラムのみに対してタイムスロットへの割り付けを試み、また、優先度&モード情報にLevelモードが示されている場合、指定された優先度よりも上位の優先度を有するプログラムのみに対してタイムスロットへの割り付けを試みられることとなる。

(1-1-4. アイドルオプション情報検索処理)

30

以下、アイドルオプション情報検索処理について説明する。

【0145】

スケジューリング制御部413は、アイドルオプション情報検索処理として、目的TSに対応づけられているアイドルオプション情報を参照し、「SWITCH」及び「MICRO-SLEEP」のいずれが示されているのかを判別する。

(1-1-5. タイムオプション情報検索処理)

スケジューリング制御部413は、タイムオプション情報検索処理として、目的TSに対応づけられているタイムオプション情報の内容を参照し、「LP_SWITCH」、「TS_SWITCH」及び「MICRO-SLEEP」のいずれが示されているのかを判別する。

(1-2. タイムドリブンスケジューリング)

40

以下、タイムドリブンスケジューリングについて説明する。

【0146】

タイムドリブンスケジューリングとは、通常のプログラム実行に関するスケジューリングである。後述のイベントドリブンスケジューリング及び放棄スケジューリングが実行されない限り、スケジューリング制御部413は、タイムドリブンスケジューリングを実行し、これにもとづいて復帰処理、レジスタ切り替え処理及び退避処理を実施する。

スケジューリング制御部413は、タイムドリブンスケジューリングとして、以下のスケジューリング処理を実施する。

(1-2-1. スケジューリング処理)

以下、スケジューリング処理について説明する。スケジューリング制御部413は、ネ

50

クストアドレスを目的アドレスとして先に述べた目的タイムスロット検索処理を実施し、目的TS（以下、ここでは、「ネクストTS」という。）を特定した後、先に述べた目的TS割り当てプログラム検索処理を実施してこのネクストTSに対応づけられている4つのプログラムを特定し、さらに、これらプログラムそれぞれに対応づけられている優先度を把握する。

【0147】

次に、スケジューリング制御部413は、先に述べた優先度&モード情報検索処理を実行し、上述のネクストTSに対応づけられている優先度&モード情報の内容、即ち、優先度及びモードを検索する。検索されたモードがLevelモードであるとき、スケジューリング制御部413は、検索された優先度の値以下の値が対応づけられている全てのプログラムを上述の目的TSへと割り当てるプログラムの候補（以下、「候補プログラム」という。）とする。

10

【0148】

検索されたモードがIndexモードであるとき、スケジューリング制御部413は、優先度&モード情報検索処理により検索された優先度の値が対応づけられている1つのプログラムを上述の候補プログラムとする。さらに、スケジューリング制御部413は、LP状態格納レジスタ424内の各候補プログラムに対応するプログラム状態情報を参照し、各候補プログラムがそれぞれ実行可能であるか否かを検出する。

【0149】

より具体的には、スケジューリング制御部413は、候補プログラムのプログラム状態情報がreadyを示している場合に、そのプログラムが実行可能であると判断し、そのプログラムを上述のネクストTSへの割り付け候補として有効にする。このようにして有効とされた候補プログラムを、有効候補プログラムという。

20

スケジューリング制御部413は、1つのネクストTSに対して、複数の有効候補プログラムが存在する場合には、優先度の値が最も小さな値が対応づけられている有効候補プログラムをこのネクストTSへの割り付け対象とする。

【0150】

また、スケジューリング制御部413は、1つのネクストTSに対して、1つの有効候補プログラムが存在する場合には、その有効候補プログラムをこのネクストTSへの割り付け対象とする。

30

有効候補プログラムをネクストTSへの割り付け対象とするのに伴い、スケジューリング制御部413は、当該ネクストTSのタイムスロット情報に有効候補プログラムを特定するIDを対応づける。さらに、スケジューリング制御部413は、上述の有効候補プログラムの実行が完了したとき、上述のタイムスロット情報から当該有効候補プログラムを特定するIDを抹消する。

【0151】

また、1つのネクストTSに対して、有効候補プログラムが1つも存在しない場合には、スケジューリング制御部413は、アイドルオプション情報検索処理を実施することにより得られる情報の内容にもとづいて、その後の処理の内容を決定する。より具体的には、スケジューリング制御部413は、上述の情報の内容が、「MICRO-SLEEP」を示す場合、いずれのプログラムも上述のネクストTSへの割り当て対象としない。

40

【0152】

なお、このようにしてプログラムが割り当てられなかったタイムスロットをエンptyタイムスロットという。

また、上述の情報の内容が、「SWITCH」を示す場合、スケジューリング制御部413は、ネクストアドレスの値に1を加えた値（ただし、ネクストアドレスが終端アドレスであった場合に限り、先頭アドレスの値）を新たなネクストアドレスとし、上述のスケジューリング処理を繰り返し実施する。

【0153】

以上のようなタイムドリブンスケジューリングを実施した後、スケジューリング制御部

50

4 1 3 は、復帰処理、レジスタ切り替え処理及び退避処理を実施する。

(1 - 3 . イベントドリブンスケジューリング) 以下、イベントドリブンスケジューリングについて説明する。イベントドリブンスケジューリングとは、現在スケジューリングの対象外となっている以下のプログラムの実行要求が生じているとき、前記タイムドリブンスケジューリングに代えて実施されるスケジューリングである。

【 0 1 5 4 】

実行要求が生じている上述のプログラムとは、即ち、EDの属性を有するプログラムであって、waitingの状態からreadyへと遷移したもの、及び、TD又はEDの属性を有するプログラムであって、suspended- waiting 又はrunningの状態からsuspended-readyの状態となっているプログラムである。なお、スケジューリング制御部 4 1 3 は、タイムドリブンスケジューリングを実施しようとしたとき、その直前において、上述のプログラムの実行要求が生じているか否かを判断する。

10

【 0 1 5 5 】

スケジューリング制御部 4 1 3 は、リクエスト検出部 4 1 9 からイベント P __ I D を受け取ったとき、受け取ったイベント P __ I D によりプログラムを特定し、 L P 状態格納レジスタ 4 2 4 を参照し、このプログラムの状態と T S __ I D の値、プログラムの属性、起床タイミングを把握し、把握した内容に応じて以下の処理を実施する。

(1 - 3 - 1 . 回復イベント処理)

スケジューリング制御部 4 1 3 は、上記特定プログラムの状態がwaiting、属性が「 T D 」、起床タイミングが「 n 」を示す場合、 L P 状態格納レジスタ 4 2 4 におけるイベント P __ I D により特定される現在のプログラムの状態をreadyに遷移させる。

20

【 0 1 5 6 】

スケジューリング制御部 4 1 3 は、このような回復イベント処理の実施後、タイムドリブンスケジューリング後に実施する復帰処理、レジスタ切り替え処理及び退避処理を実施する。

(1 - 3 - 2 . 緊急回復イベント処理)

上記特定プログラムの状態がsuspended-waiting、属性が「 T D 」、起床タイミングが「 e 」を示す場合、スケジューリング制御部 4 1 3 は、この特定プログラムの状態をsuspended-readyに遷移させる。

【 0 1 5 7 】

30

そして、スケジューリング制御部 4 1 3 は、この特定プログラムの状態を示すコードに付されている T S __ I D の値からタイムスロット (以下、「特定 T S 」という。) を特定し、この特定 T S に上述の特定プログラムを割り当てる。このような緊急回復イベント処理を実施した後、スケジューリング制御部 4 1 3 は、後述の緊急復帰処理を実施し、また、タイムドリブンスケジューリング後に実施されるレジスタ切り替え処理に代えて、後述の緊急レジスタ切り替え処理を実施する。

(1 - 3 - 3 . イベント起床処理) また、上記特定プログラムの状態がsuspended-waiting、属性が「 T D 」、起床タイミングが「 n 」を示す場合、 L P 状態格納レジスタ 4 2 4 におけるイベント P __ I D により特定される現在のプログラムの状態をsuspended-readyに遷移させる。

40

【 0 1 5 8 】

さらに、上記特定プログラムの状態がwaiting又はsuspended-waiting、属性が「 E D 」、起床タイミングが「 n 」を示す場合、スケジューリング制御部 4 1 3 は、受け取ったイベント P __ I D により特定されるプログラムの状態がwaitingであればreadyに、suspended-waitingであればsuspended-readyに変更する。加えて、スケジューリング制御部 4 1 3 は、イベントドリブンスケジューリング後に実施される起床タイミング「 e 」の緊急レジスタ切り替え処理が発生した場合、その事象の発生によって、実行中のプログラムの状態を属性に関係なくrunningからsuspended-readyへと変更する。

【 0 1 5 9 】

このイベント起床処理を実施した場合、スケジューリング制御部 4 1 3 は、タイムドリ

50

ブンスケジュールリング後に実施される復帰処理に代えて、後述の早期復帰処理を実施し、カウント値 0 信号の到来をトリガとして、後述のレジスタ切り替え処理を実施する。このようなイベント処理をイベント起床処理という。

(1 - 3 - 4 . 緊急イベント起床処理) また、上記特定プログラムの状態がwaiting又はsuspended-waiting、属性が「 E D 」、起床タイミングが「 e 」を示す場合、スケジュールリング制御部 4 1 3 は、特定プログラムの状態がwaitingであればreadyに、suspended-waitingであればsuspended-readyに変更する。

【 0 1 6 0 】

この緊急イベント起床処理を実施した場合、スケジュールリング制御部 4 1 3 は、タイムドリブンスケジュールリング後に実施される復帰処理に代えて、後述の緊急復帰処理を実施し、また、タイムドリブンスケジュールリング後に実施されるレジスタ切り替え処理に代えて、後述の緊急レジスタ切り替え処理を実施する。このようなイベント処理を緊急イベント起床処理という。

(1 - 4 . 放棄スケジュールリング)

スケジュールリング制御部 4 1 3 が、処理制御部 4 1 1 を介して処理実行部 4 1 0 からウエイトコードを受け取った場合、もしくは、DMA管理部 4 1 8 から後に述べるキャッシュミス信号を受け取った場合に実施する放棄スケジュールリングについて説明する。

【 0 1 6 1 】

上述のウエイトコードとは、実行中のプログラムの継続が不可能となったとき、実行中のプログラムに割り当てられている時間が放棄されたことを示すコードであり、このウエイトコードには、「vmp_wait」と、「vmp_sus」とがある。「vmp_wait」コードは、上述の放棄時間が比較的長いことが見込まれる場合に、処理実行部 4 1 0 により発せられるコードであり、また、「vmp_sus」コードは、上述の放棄時間が比較的短いことが見込まれる場合に、処理実行部 4 1 0 により発せられるコードである。

【 0 1 6 2 】

スケジュールリング制御部 4 1 3 は、処理制御部 4 1 1 を介して処理実行部 4 1 0 から「vmp_wait」コード、「vmp_sus」コード及びキャッシュミス信号を受け取った場合、LP状態格納レジスタ 4 2 4 内に格納されている実行中のプログラムの状態を更新する。より具体的には、「vmp_wait」コードを受け取った場合、スケジュールリング制御部 4 1 3 は、実行中のプログラムの状態をrunningからwaitingに変更する。

【 0 1 6 3 】

また、「vmp_sus」コードを受け取った場合、又は、キャッシュミス信号を受け取った場合、スケジュールリング制御部 4 1 3 は、実行中のプログラムの状態をrunningからsuspended-waitingに変更する。そして、スケジュールリング制御部 4 1 3 は、カレントアドレスを目的アドレスとして、先に述べたタイムオプション情報検索処理を実施する。

受け取ったウエイトコードが「vmp_sus」の場合、スケジュールリング制御部 4 1 3 は、タイムオプション情報に関係なく、後述の緊急レジスタ切り替え処理を実施し、退避処理を実施して、さらに後述の制御レジスタ値更新処理を実施する。

【 0 1 6 4 】

また、受け取ったウエイトコードが「vmp_wait」であり、かつ、タイムオプション情報の内容が「TS_SWITCH」を示す場合においても、上記と同様にスケジュールリング制御部 4 1 3 は、後述の緊急レジスタ切り替え処理を実施し、退避処理を実施して、さらに後述の制御レジスタ値更新処理を実施する。

なお、前記緊急レジスタ切り替え処理が実施される直前のスタンバイレジスタ群では、タイムドリブンスケジュールリングによる復帰処理か、又は、イベントドリブンスケジュールリングで実施されるイベント起床処理後の早期復帰処理がなされている。

【 0 1 6 5 】

また、受け取ったウエイトコードが「vmp_wait」であって、かつ、タイムオプション情報の内容が「LP_SWITCH」を示すとき、スケジュールリング制御部 4 1 3 は、現在実行対象となっているタイムスロットに有効候補プログラムが存在するか否かを判断し、有効候補

10

20

30

40

50

プログラムが存在する場合には、有効候補プログラムのうち最も優先度が高いプログラム（有効候補プログラムが1つの場合には、そのプログラム）に対応するコンテキストのレジスタ値をスタンバイレジスタ群に格納、即ち、復帰を実施する。

【0166】

その後、スケジューリング制御部413は、後述の緊急レジスタ切り替え処理を実施し、退避処理を実施する。

また、有効候補プログラムが存在しない場合、及び、タイムオブション情報の内容が「MICRO-SLEEP」を示す場合、スケジューリング制御部413は、処理実行部410においてカレントレジスタ群内のレジスタ値の更新を伴わないプログラム実行である後述のNOP実行させるように処理制御部411に指示すると共に、以下の処理を実施する。

10

【0167】

即ち、スケジューリング制御部413は、直ちにマスク信号のクロックマスク部427への出力を開始し、さらに、スタンバイレジスタ群に格納されているコンテキストの内容を参照し、このコンテキストに対応するプログラムの実行に必要なデータが格納されているメモリのアドレスの値を取得し、このアドレスに格納されているデータをキャッシュメモリに書き込むようにDMA管理部418に指示する。

【0168】

さらに、スケジューリング制御部413は、次にレジスタ切り替え、又は緊急レジスタ切り替えが実施されるとき、マスク信号のクロックマスク部427への出力を停止する。スケジューリング制御部413は、サイクルカウンタ416からカウント値0信号を受け取ったとき、後述のレジスタ切り替え処理及び退避処理を実施する。

20

次に、復帰処理、緊急復帰処理及び早期復帰処理などの復帰に関する処理について説明する。

（1-5-1．復帰処理）

復帰処理とは、タイムドリブンスケジューリング実施後において、スケジューリング制御部413により実施される、レジスタ値をスタンバイレジスタ群に格納するための処理である。

【0169】

スケジューリング制御部413は、復帰処理として、スタンバイレジスタ群が開放されたとき、即ち、退避が完了したとき、上述のタイムドリブンスケジューリングにより割り当てられたプログラムのコンテキストのレジスタ値を、上述のスタンバイレジスタ群に格納するようにDMA管理部418に指示する。

30

（1-5-2．緊急復帰処理）

緊急復帰処理とは、イベントドリブンスケジューリングを実施した結果、先に述べた特定プログラムの起床タイミングが「e」を示す場合において、スケジューリング制御部413により実施される、レジスタ値をスタンバイレジスタ群に格納するための処理である。

【0170】

スケジューリング制御部413は、緊急復帰処理として、上述の緊急回復イベント処理又は緊急イベント起床処理を実行が完了したとき、スタンバイレジスタ群に格納されているレジスタ値を退避して、直ちに特定プログラムに対応するコンテキストのレジスタ値を上述のスタンバイレジスタ群に格納、即ち、緊急的な復帰を実施するようにDMA管理部418に指示する。

40

【0171】

なお、上述の退避及び復帰は、レジスタ群単位ではなく、各レジスタ単位で実行することが可能であり、レジスタ毎に退避、復帰を順次実施することにより、これらの退避及び復帰が擬似並行的に実施されることとなる。

（1-5-3．早期復帰処理）

早期復帰処理とは、イベントドリブンスケジューリングを実施した結果、先に述べた特定プログラムの起床タイミングが「n」を示す場合において、スケジューリング制御部4

50

13により実施される、レジスタ値をスタンバイレジスタ群に格納するための処理である。

【0172】

スケジューリング制御部413は、早期復帰処理として、スタンバイレジスタ群が開放されたとき、即ち、退避が完了したとき、直ちに特定プログラムに対応するコンテキストのレジスタ値を、上述のスタンバイレジスタ群に格納するようにDMA管理部418に指示する。なお、スケジューリング制御部413は、緊急復帰処理と同様に、上述の格納に先行して、スタンバイレジスタ群のレジスタ値の初期化を実施する。

【0173】

つづいて、レジスタ切り替え処理及び緊急レジスタ切り替え処理などのレジスタ切り替えに関する処理について説明する。

(1-6-1. レジスタ切り替え処理)

スケジューリング制御部413は、サイクルカウンタ416からカウント値0信号を受け取ったとき、後述の基本サイクル調整処理を実施し、さらに後述の切り替えタイミング確認処理を行った上でレジスタ切り替えを実施するようにレジスタ切替部412に指示する。

【0174】

また、スケジューリング制御部413は、レジスタ切り替えが完了した旨の通知をレジスタ切替部412から受け取ったとき、基本サイクルの値をサイクルカウンタ416に出力し、さらに、後述の実時間保存処理及び後述の制御レジスタ値更新処理を実施する。この時、実行対象となっているタイムスロットが先に述べたエンプティタイムスロットである場合、スケジューリング制御部413は、基本サイクルの値をサイクルカウンタ416に出力するのに伴い、処理実行部410においてNOP実行させるように処理制御部411に指示すると共に、クロックマスク部427へのマスク信号の出力を開始し、さらに、スタンバイレジスタ群に格納されているコンテキストの内容を参照し、このコンテキストに対応するプログラムの実行に必要なデータが格納されているメモリのアドレスの値を取得し、このアドレスに格納されているデータをキャッシュメモリに書き込むようにDMA管理部418に指示する。

【0175】

また、スケジューリング制御部413は、次にレジスタ切り替え又は緊急レジスタ切り替えが実施されたとき、このマスク信号のクロックマスク部427への出力を停止する。

(1-6-1-1. 基本サイクル調整処理)

以下、基本サイクル調整処理について説明する。

スケジューリング制御部413は、レジスタ切り替えの直前に、LP状態格納レジスタ424を参照し、スタンバイレジスタ群に格納されているレジスタ値群に対応するプログラムの状態がsuspended-readyであるか否かを判断し、suspended-readyである場合には、スタンバイレジスタ群内の後述の残余サイクル値記憶領域から後述の実カウント値を取得する。

【0176】

そして、スケジューリング制御部413は、次のプログラム実行時において、このプログラムが割り当てられているタイムスロットのタイムスロット情報に示されている正規の基本サイクルに代えて、この実カウント値をこのプログラムの実行時間、つまり、このプログラムが割り当てられているタイムスロットの基本サイクルとする。

さらに、このタイムスロットのTS_IDの値が3である場合、即ち、リアルタイム処理を保証しないタイムスロットである場合、スケジューリング制御部413は、TS_IDの値が3を示すタイムスロット情報内のタイムラグ値を取得し、このタイムラグ値を上述の基本サイクルの値から差し引いたものを、次のプログラム実行時における新たな基本サイクルとすると共に、このタイムスロット情報内の前記タイムラグ値を0に更新する。

【0177】

なお、スケジューリング制御部 4 1 3 は、処理制御部 4 1 1 からプログラム実行の開始の通知を受け取ったとき、以上のように求められた基本サイクルの値をサイクルカウンタ 4 1 6 に出力する。

(1 - 6 - 1 - 2 . 切り替えタイミング確認処理)

以下、切り替えタイミング確認処理について説明する。

【 0 1 7 8 】

プログラムを時分割実行する場合、例えば、プログラムの実行に用いられる共有資源などの都合によりプログラムの実行を任意の時間に終了することができない。つまり、任意の時間でレジスタ切り替えを実施することはできない。このため、スケジューリング制御部 4 1 3 は、まず、プログラムの終了を処理制御部 4 1 1 に依頼し、プログラムの実行の終了が可能な点でプログラムを終了させ、レジスタ切り替えが可能であるか否かを確認する。

10

【 0 1 7 9 】

より具体的には、上述の依頼後、スケジューリング制御部 4 1 3 は、プロセッサ状態格納部 4 2 5 内の切替禁止フラグの値及び共通割り込みフラグの値を参照し、この切替禁止フラグの値及び共通割り込みフラグの値が 0 である場合にはレジスタ切り替えが可能であり、切替禁止フラグの値及び共通割り込みフラグの値が 1 である場合にはレジスタ切り替えが不可能であると判断する。

(1 - 6 - 1 - 3 . 実時間保存処理)

以下、実時間保存処理について説明する。

20

【 0 1 8 0 】

スケジューリング制御部 4 1 3 は、レジスタ切り替えを実施するようにレジスタ切替部 4 1 2 に指示したとき、その時点においてサイクルカウンタ 4 1 6 が有するカウント値を取得し、このカウント値（以下、「実カウント値」という。）が正の値をとり、かつ、上述のレジスタ切り替えのトリガとなる「vmp_sus」コードを受け取っている場合、この実カウント値を、前記レジスタ切り替えが実行された結果、スタンバイレジスタ群となるレジスタ群の決められた領域（以下、「残余サイクル値記憶領域」という。）に書き込む。

【 0 1 8 1 】

このようにすることにより、実行を終了したプログラムの実カウント値が、このプログラムのコンテキストに付され、このコンテキストの退避に伴い、コンテキストメモリ 5 0 0 へと格納されるため、次回プログラム実行時において、前回のプログラム実行時における実カウント値が取得可能となる。

30

また、上記実カウント値が負の値をとる場合、もしくは、上記実カウント値が正の値をとり、かつ、上述のレジスタ切り替えのトリガとなる「vmp_wait」コードを受け取っている場合、スケジューリング制御部 4 1 3 は、終端アドレスを目的アドレスとして目的タイムスロット検索処理を実施し、これにより特定されるタイムスロット（TS_ID=3）のタイムスロット情報の格納場所を特定し、このタイムスロット情報に格納されているタイムラグ値からこの実カウント値を加えることにより、このタイムラグ値を更新する。

【 0 1 8 2 】

なお、上述の切り替えタイミング確認処理の説明において述べた理由により、切り替えのタイミングをタイムスロットの基本サイクルと厳密に一致させることはできず、通常、数サイクル程度オーバーして切り替えるため、上述の実カウント値は、負の値をとり、また、プログラムの実行が中断されることによりレジスタ切り替えが行なわれたときなどには、実カウント値は、実行対象となっているタイムスロットの基本サイクル未満となるため正の値をとる。

40

(1 - 6 - 1 - 4 . 制御レジスタ値更新処理)

以下、制御レジスタ値更新処理について説明する。

【 0 1 8 3 】

制御レジスタ値更新処理とは、以下に示すスケジューリングの実施に伴って実行される制御レジスタ 4 2 0 内のデータ更新である。上述のスケジューリングの実施とは、即ち、

50

１）タイムドリブンスケジューリングの実施、２）タイムオプション情報の内容が「TS_SWITCH」を示す場合における放棄スケジューリングの実施のことである。

スケジューリング制御部４１３は、現時点におけるネクストアドレスの値を新たなカレントアドレスの値とすると共に、当該ネクストアドレスの値に１を加えた値（ただし、ネクストアドレスが終端アドレスであった場合に限り、先頭アドレスの値）を新たなネクストアドレスとし、これに伴って、制御レジスタ４２０内に格納されているカレントアドレスの値及びネクストアドレスの値を更新する。

【０１８４】

また、スケジューリング制御部４１３は、カレントアドレスの値を、終端アドレスの値から先頭アドレスの値へと移行させたとき、サイクル先頭信号をリクエスト検出部４１９

10

（１－６－２．緊急レジスタ切り替え処理）

以下、緊急レジスタ切り替え処理について説明する。

【０１８５】

スケジューリング制御部４１３は、放棄スケジューリングのタイムオプション処理において、タイムオプション情報の内容が「TS_SWITCH」を示すとき、並びに、イベントドリブンスケジューリング時の緊急回復イベント処理及び緊急イベント起床処理後に緊急復帰処理が完了したときに、上述の基本サイクル調整処理を実施し、さらに上述の切り替えタイミング確認処理を行った上でレジスタ切り替えを実施するようにレジスタ切替部４１２

20

【０１８６】

そして、スケジューリング制御部４１３は、レジスタ切り替えが完了した旨の通知をレジスタ切替部４１２から受け取ったとき、基本サイクルの値をサイクルカウンタ４１６に出力し、さらに、上述の実時間保存処理を実施する。

（１－７－１．退避処理）

スケジューリング制御部４１３は、レジスタ切り替えが完了した旨の通知をレジスタ切替部４１２から受け取ったとき、このレジスタ切り替えによりスタンバイレジスタ群となるレジスタ群に格納されているコンテキストのレジスタ値を、もともとのコンテキストが格納されていたコンテキストメモリ５００上の記憶領域に上書きする、即ち、退避するようにDMA管理部４１８に指示する。

30

【０１８７】

なお、上述のレジスタ切り替えが実施される直前において、実行対象となっていたタイムスロットが先に述べたエンブティタイムスロットであるときにおいても上述の退避を行う。

（１－８－１．固有割込み処理）

また、スケジューリング制御部４１３は、リクエスト検出部４１９から、固有P__ID及び固有S P__IDを受け取った場合、以下の処理を実施する。

【０１８８】

即ち、スケジューリング制御部４１３は、LP状態格納レジスタ４２４を参照し、受け取った固有P__IDにより特定されるプログラムが、現在実行中であるか否かを判断し、現在実行中である場合、直ちに処理制御部４１１を介し、処理実行部４１０に受け取った固有S P__IDにより特定されるサブルーチンプログラムにジャンプして処理を実行するように指示する。

40

【０１８９】

また、受け取った固有P__IDにより特定されるプログラムが、現在実行中でない場合、スケジューリング制御部４１３は、リクエスト検出部４１９内のフラグを参照することによりこのサブルーチンプログラムの割込み先となるプログラムを特定し、この特定されたプログラムが実行される時に、処理制御部４１１を介して処理実行部４１０にこのサブルーチンプログラムにジャンプして処理を実行するように指示する。

（１－８－２．共通割り込み処理）

50

また、スケジューリング制御部 4 1 3 は、リクエスト検出部 4 1 9 から、共通 S P _ I D を受け取った場合、直ちに処理制御部 4 1 1 を介し、処理実行部 4 1 0 に受け取った共通 S P _ I D により特定されるサブルーチンプログラムにジャンプして処理を実行するように指示し、処理制御部 4 1 1 から当該サブルーチンプログラムの実行開始の通知を受け取ったとき、サイクルカウンタ 4 1 6 にカウントダウンを中断するように指示する。

【 0 1 9 0 】

また、スケジューリング制御部 4 1 3 は、処理制御部 4 1 1 から上述の共通 S P _ I D により特定されるサブルーチンプログラムの実行が完了したことを示す通知を受け取ったとき、サイクルカウンタ 4 1 6 にカウントダウンの再開を指示するカウント再開信号を出力する。以上のように、スケジューリング制御部 4 1 3 は、様々な処理を実行する機能を有する。

10

【 0 1 9 1 】

処理制御部 4 1 1 は、処理実行部 4 1 0 におけるプログラムの実行状況を管理する機能を有し、処理実行部 4 1 0 のプログラム実行状況を監視し、処理実行部 4 1 0 におけるプログラム実行の開始タイミング及び終了タイミングと、割込み処理の開始タイミング及び終了タイミングを把握する。処理制御部 4 1 1 は、これらタイミングが到来すること、これらタイミングが到来した旨をスケジューリング制御部 4 1 3 に通知する。

【 0 1 9 2 】

また、処理制御部 4 1 1 は、処理実行部 4 1 0 からウエイトコードを受け取ったとき、このウエイトコードをスケジューリング制御部 4 1 3 に出力する。さらに、処理制御部 4 1 1 は、処理実行部 4 1 0 におけるプログラム実行が中断できない状態にあるとき、プロセッサ状態格納部 4 2 5 内の切替禁止フラグの値を 1 (デフォルト値は 0) にし、また、処理実行部 4 1 0 において、共通 S P _ I D により特定されるサブルーチンプログラムが実行されている間、プロセッサ状態格納部 4 2 5 内の共通割り込みフラグの値を 1 (デフォルト値は 0) にする。

20

【 0 1 9 3 】

さらに、処理制御部 4 1 1 は、処理実行部 4 1 0 からレジスタ切替禁止命令を受け取ったとき、プロセッサ状態格納部 4 2 5 内の切替禁止フラグの値を 1 にし、また、処理実行部 4 1 0 からレジスタ切替禁止解除命令を受け取ったとき、上記切替禁止フラグの値を 0 にする。また、処理制御部 4 1 1 は、スケジューリング制御部 4 1 3 から NOP 実行させる指示を受け取った場合、処理実行部 4 1 0 を NOP 実行させる。

30

【 0 1 9 4 】

D M A 管理部 4 1 8 は、D M A 転送を円滑に実施し、管理する機能部であり、処理実行部 4 1 0、スケジューリング制御部 4 1 3 などの各機能部からの要求にもとづいて D M A 転送を実施する。

また、D M A 管理部 4 1 8 は、各機能部から要求された D M A 転送が完了したとき、その旨を示す信号 (以下、「転送完了信号」という。) を当該機能部及びスケジューリング制御部 4 1 3 に出力し、また、処理実行部 4 1 0 からの要求に応じ、キャッシュメモリからプログラム及びデータを取得しようとして、キャッシュミスが生じた場合、キャッシュミスが生じたことを示す信号 (以下、「キャッシュミス信号」という。) をスケジューリング制御部 4 1 3 に出力する。

40

【 0 1 9 5 】

外部 I / F 4 1 7 は、外部プロセッサなどの外部機器に接続するインターフェースであり、外部から受け取った信号をスケジューリング制御部 4 1 3 およびリクエスト検出部 4 1 9 に出力する。サイクルカウンタ 4 1 6 は、クロックカウンタを有し、スケジューリング制御部 4 1 3 から基本サイクルの値を受け取ったとき、この値をカウンタの初期値としてカウントダウンを開始する。

【 0 1 9 6 】

サイクルカウンタ 4 1 6 は、上述の初期値から 1 クロック毎に 1 ずつディクリメントすることにより上述のカウントダウンを行い、値が 0 となったとき、スケジューリング制御

50

部 4 1 3 にその旨を示す信号（以下、「カウント値 0 信号」という。）を出力する。

また、サイクルカウンタ 4 1 6 は、スケジューリング制御部 4 1 3 からの指示にもとづき、カウントダウンを中断し、再開し、さらに、指示された時点におけるカウンタの値である実カウント値をスケジューリング制御部 4 1 3 に出力する。

【 0 1 9 7 】

クロックマスク部 4 2 7 は、スケジューリング制御部 4 1 3 からマスク信号を受信している間、処理実行部 4 1 0 へのクロックパルスの信号経路を遮断する機能を有する。

（ 2 . データ ） 図 1 6 は、時間割付情報格納レジスタ 4 0 2 及び L P 状態格納レジスタ 4 2 4 に格納されている情報の内容を示す図である。

時間割付情報格納レジスタ 4 0 2 には、 1) タイムスロット情報と、 2) L P 情報と、 3) 優先度 & モード情報と、 4) アイドルオプション情報と、 5) タイムオプション情報とが格納されている。これらの情報は、タイムスロット毎に用意され、互いに関連付けられている。

（ 2 - 1 - 1 . タイムスロット情報 ） タイムスロット情報には、タイムスロットを特定するための T S _ I D の値と、基本サイクルの値とが含まれている。

【 0 1 9 8 】

また、 T S _ I D の値が 3 のタイムスロット情報には、さらに、タイムラグ値が含まれている。

また、スケジューリング制御部 4 1 3 によりスケジューリングが実施されると、プログラムが割り当てられたタイムスロットの T S _ I D の値に、このプログラムの I D が対応づけられることとなる。

（ 2 - 1 - 2 . L P 情報 ） L P 情報には、タイムスロットへの割り当て候補となるプログラムに関する情報が格納されている。

【 0 1 9 9 】

1 つのタイムスロットへの割り当て候補となるプログラムの属性及び数は、割り当て先のタイムスロットの属性により異なる。即ち、 T D の属性を有するプログラムは、必ず T D の属性を有するタイムスロットに割り当てられる。

また、 T D の属性を有する各タイムスロットには、 T D の属性を有する 4 つの候補プログラムが用意されている。

【 0 2 0 0 】

一方、 E D の属性を有するプログラムは、必ず E D の属性を有するタイムスロットに割り当てられる。

また、 E D の属性を有する各タイムスロットには、 E D の属性を有する 1 つの候補プログラムが用意されているのみである。各 L P 情報は、上述の割り当て先のタイムスロットに対応するタイムスロット情報に対応づけられている。

【 0 2 0 1 】

L P 情報には、プログラムを特定するための I D の値と、優先度を示す値と、このプログラムに対応するコンテキストが格納されているコンテキストメモリ 5 0 0 上のアドレスの値との組が、割り当て候補となるプログラムの数だけ格納されている。上述の優先度を示す値は、 0 、 1 、 2 、 3 のいずれかの値であって、割り当て候補となる各プログラム同士の優先度が重複しないように設定されている。

【 0 2 0 2 】

なお、割り当て候補となるプログラムの数が 1 つの場合の優先度を示す値は、 0 となる。

（ 2 - 1 - 3 . 優先度 & モード情報 ）

優先度 & モード情報には、タイムスロットへの割り当て候補となるプログラムの指定に関する情報が格納されている。

【 0 2 0 3 】

即ち、優先度 & モード情報には、優先度を示す 0 、 1 、 2 、 3 のいずれか 1 つの値と、モードを示すコード「 I 」又は「 L 」が格納されている。ここでコード「 I 」は、 Index

10

20

30

40

50

モードを示し、コード「L」は、Levelモードを示す。この優先度 & モード情報の内容を変更することにより、有効となる候補プログラムの範囲が変化するため、各候補プログラムの実行頻度を設計上好ましい値に近づけることができる。

(2 - 1 - 4 . アイドルオプション情報)

アイドルオプション情報には、タイムスロットに割り付けようとしたプログラムが全て実行不可能な状態となっている場合におけるプログラム実行動作を決定する情報が格納されている。

【 0 2 0 4 】

即ち、アイドルオプション情報には、省電力化を示すコード「MICRO-SLEEP」又は高効率なプログラム実行を示すコード「SWITCH」が格納されている。

10

(2 - 1 - 5 . タイムオプション情報)

タイムオプション情報には、実行中のプログラムが中断した場合における、中断以降のプログラム実行動作を決定する情報が格納されている。

【 0 2 0 5 】

即ち、タイムオプション情報には、実行中のプログラムが中断した場合、このプログラムに代えて、このプログラムが属しているタイムスロットに対応づけられている実行可能な別のプログラムをタイムスロットに割り付けることを示すコード「LP_SWITCH」、次のタイムスロットを実行対象とすることを示すコード「TS_SWITCH」、又は、省電力化を示すコード「MICRO-SLEEP」が格納されている。

【 0 2 0 6 】

20

LP状態格納レジスタ424には、1)プログラム状態情報と、2)タイムスロット設定情報とが格納されている。

(2 - 2 . プログラム状態情報)

プログラム状態情報には、プログラム毎に、プログラムを特定するためのIDと、プログラム状態を示すコード及びプログラムの属性ならびにプログラムを起床又は回復させるタイミングを示すコードが格納されている。

【 0 2 0 7 】

プログラム状態を示すコードとしては、waiting、suspended-waiting、ready、suspended-ready、running、nonexistentの6つがある。waitingコードは、現在、プログラムの実行時間を放棄しており、かつ、次回プログラム実行時に通常のタイムスロットのプログラム実行時間枠でプログラム実行することを示し、また、suspended-waitingコードは、現在、プログラムの実行時間を放棄しており、かつ、放棄して使用されなかった時間(最大1スロット分)を次回プログラム実行時におけるプログラム実行時間とすることを示す。

30

【 0 2 0 8 】

readyコードは、現在、プログラムが実行可能な状態であることを示し、suspended-readyコードは、suspended-waitingコードであったプログラムが、現在、プログラムが実行可能な状態であることを示す。

また、プログラム実行時間が放棄されたことにより、waitingコード、suspended-waitingコード及びsuspended-readyコードの状態に遷移した場合には、これらコードに、前記放棄が生じた時点において割り当て先となっていたタイムスロットのTS_IDの値が付されている。

40

【 0 2 0 9 】

なお、このTS_IDの値は、実行されるまでクリアされないため、readyコード及びsuspended-readyコードにも付されている場合がある。runningコードは、現在、プログラムが実行可能な状態であり、かつ、実行中であることを示す。nonexistentコードは、現在、プログラムがスケジューリングの対象範囲外にあることを示す。

ここでスケジューリングの対象範囲外とは、現時点において、候補プログラムにさえなり得ないプログラムの状態を意味する。ちなみに、候補プログラムであって、waiting又はsuspended-waitingの状態となっているものを、候補スケジューリングの対象外という

50

こととする。なお、先に述べたプログラムの回復とは、「TD」属性のプログラムが、waitingの状態からreadyになったこと、あるいは、「TD」属性のプログラムがsuspended-waitingの状態から上述したタイミング「e」で、suspended-readyになったことを意味する。

【0210】

また、プログラムの起床とは、「TD」属性のプログラムが、suspended-waitingの状態から上述したタイミング「n」で、suspended-readyになったこと、あるいは、「ED」属性のプログラムが、waiting又はsuspended-waitingの状態からそれぞれready又はsuspended-readyの状態となったことを意味する。なお、このようなプログラムを起床又は回復させるタイミングを総称して、本実施例では、先に述べたように単に起床タイミングと呼んでいる。

10

【0211】

プログラムの属性を示すコードには、イベントが発生すると候補スケジューリングの対象範囲内となりイベントドリブンスケジューリングにて、タイムドリブンスケジューリングで決定される所定の順番に関係なく、タイムスロットに割り付けられて実行されるイベントドリブン型のプログラムであることを示すコード「ED」と、イベントが発生すると、候補スケジューリングの対象範囲内となるが、タイムドリブンスケジューリングにより所定の順番まで待ってからタイムスロットに割り付けられて実行されるプログラム、即ち、タイムドリブン型のプログラムであることを示すコード「TD」がある。

【0212】

20

つまり、ED属性のプログラムは、waiting又はsuspended waitingの状態からそれぞれready又はsuspended readyに移行し、イベントドリブンスケジューリングによりタイムスロットに割り付けられ、一方、TD属性のプログラムはwaitingからreadyに遷移し、タイムドリブンスケジューリングにより、タイムスロットに割り付けられる。

ただし、例外として、タイムドリブン型のプログラムであっても、イベントが発生すると候補スケジューリングの対象範囲内となり、イベントドリブンスケジューリングにて、タイムドリブンスケジューリングで決定される所定の順番に関係なくタイムスロットに割り付けられ実行される場合がある。つまり、この例外とは、タイムドリブン型のプログラムがsuspended waitingの状態からsuspended readyに遷移し、イベントドリブンスケジューリングにより、タイムスロットに割り付けられる場合である。

30

【0213】

起床タイミングを示すコードには、既にスタンバイレジスタ群への格納が完了したコンテキストを消去してまでも、カウント値0信号を受け取るまで待つことなく緊急的にプログラムを実行する必要があることを示すコード「e」、及び、スタンバイレジスタ群から退避処理が完了した後、スタンバイレジスタ群への格納が完了し、カウント値0信号を受け取ってからプログラムを実行することを示すコード「n」がある。

【0214】

設定フラグレジスタ421には、プログラム毎に用意された設定フラグ群が格納されている。この設定フラグ群は、各プログラムが許容する割込みやイベントの内容を特定するためのものであり、プログラム毎に用意されている以外は、第1の実施形態において述べた設定フラグ群と同様であるため、ここでは詳細な説明を省略する。

40

リクエスト検出部419には、図15に示すように、プログラム毎に用意されたLP固有フラグ群及び1つの共通フラグ群が格納されている。LP固有フラグ群とは、固有割込みに関する情報であり、現在、どのプログラムに対してどのような固有割込みの実行要求が発生しているのかを示す。このLP固有フラグ群は、見かけ上プログラム毎に存在するが、その実態は1つのビット列データであって、各フラグは、割込みを許容するプログラム及び固有割込みの内容に対応するビット位置に存在している。

【0215】

つまり、或るプログラムにおいてその内容の割込みを許容する場合には、そのプログラム及び割込みの内容に対応するビット位置に存在するLP固有フラグの値(デフォルト値

50

は 0) が 1 となっている。そのデータ構成については、第 1 の実施形態において述べた完了通知フラグと同様であるため、ここでは詳細な説明を省略する。

共通フラグ群とは、共通割り込みの実行要求が発生しているか否かを示すフラグの集まりである。ここで、共通割り込みの実行要求とは、どのプログラムにおいても実行可能なサブルーチンプログラムの実行要求を意味する。共通フラグ群の実態は、1 つのビット列データであって、割り込んで実行すべきサブルーチンプログラム固有のビット位置が存在する。

【 0 2 1 6 】

つまり、共通割り込み要求が生じているサブルーチンプログラムがある場合に、そのサブルーチンプログラムに対応するビット位置のフラグの値が 1 (デフォルト値は 0) となっている。プロセッサ状態格納部 4 2 5 は、切替禁止フラグの値及び共通割り込みフラグの値を格納している。

10

切替禁止フラグは、例えば、プログラムの実行中において共有資源へアクセスしており、プログラムを即座に中断することができないなどの理由により、現在レジスタ切り替えを実行できない状況にあることを示し、レジスタ切り替えを実行できない間、1 (デフォルト値は 0) の値をとる。この切替禁止フラグの値は、実行中のプログラムが発する命令により、書き換えが可能である。

【 0 2 1 7 】

共通割り込みフラグは、現在、共通割り込みの処理が実行中であることを示し、共通割り込みの処理が実行されている間、1 (デフォルト値は 0) の値をとる。

20

(3 . 動作)

以下、仮想マルチプロセッサ 4 0 0 において、1 つのプロセッサにより擬似並行的にプログラムを実行する際、効率的なプログラムの実行を実現する処理について説明する。

【 0 2 1 8 】

図 1 7 は、仮想マルチプロセッサ 4 0 0 において実施される処理を示すフローチャートである。処理実行部 4 1 0 は、処理制御部 4 1 1 からの指示にもとづいて、プログラムの実行を開始する (ステップ S 1 0 0) 。このプログラム実行に並行して、スケジューリング制御部 4 1 3 は、スタンバイレジスタ群に格納されているレジスタ値を対象とする退避処理を実施し (ステップ S 1 0 1) 、リクエスト検出部 4 1 9 は、外部 I / F 4 1 7 から受け取った外部信号にもとづいて、イベント処理要求が発生しているか否かを判別する (ステップ S 1 0 2) 。

30

【 0 2 1 9 】

イベント処理要求が発生している場合、リクエスト検出部 4 1 9 は、イベント P __ I D をスケジューリング制御部 4 1 3 に出力する。スケジューリング制御部 4 1 3 は、このイベント P __ I D にもとづいて、イベントドリブンスケジューリングを実施する (ステップ S 1 0 3) 。なお、イベントドリブンスケジューリング実施後に実施される処理の内容は、イベントドリブンスケジューリングの実施結果により異なる。

【 0 2 2 0 】

イベント処理要求が発生していない場合、スケジューリング制御部 4 1 3 は、目的 T S をネクスト T S とし (ステップ S 1 0 4) 、タイムドリブンスケジューリングを実施し (ステップ S 1 0 5) 、さらに、復帰処理を実施する (ステップ S 1 0 6) 。そして、リクエスト検出部 4 1 9 は、割り込み処理要求が発生しているか否かを判別し (ステップ S 1 0 7) 、割り込み処理要求が発生している場合、割り込みの種別に応じて、固有 P __ I D 、固有 S P __ I D 及び共通 S P __ I D などをスケジューリング制御部 4 1 3 に出力する。

40

【 0 2 2 1 】

これら I D を受け取ったとき、スケジューリング制御部 4 1 3 は、割り込み処理を実行する (ステップ S 1 0 8) 。一方、割り込み処理要求が発生していない場合、スケジューリング制御部 4 1 3 は、ウエイトコード及びキャッシュミス信号にもとづいて、プログラム実行時間の放棄又はキャッシュミスが発生しているか否かを判断し (ステップ S 1 0 9) 、これらが発生している場合には、放棄スケジューリングを実施する (ステップ S 1 1 0)

50

。

【0222】

なお、放棄スケジューリング実施後に実施される処理の内容は、放棄スケジューリングの実施結果により異なる。一方、プログラム実行時間の放棄又はキャッシュミスが発生していない場合、スケジューリング制御部413は、カウント値0信号をサイクルカウンタ416から受け取ったとき、処理実行部410におけるプログラム実行を終了させ（ステップS111）、レジスタ切り替え処理を実施した後（ステップS112）、処理実行部410によるプログラム実行を開始するステップS100に戻る。

【0223】

図18は、スケジューリング制御部413により実施されるタイムドリブンスケジューリングの内容を示すフローチャートである。まず、目的TSをネクストTSとした場合の動作を説明する。スケジューリング制御部413は、図18(A)のステップS104の処理に引き続き、目的タイムスロット検索処理を実施する（ステップS120）。

より具体的には、スケジューリング制御部413は、時間割付情報格納レジスタ402内のネクストアドレスに格納されているタイムスロット情報を参照し、タイムスロットを特定するTS_IDによりネクストTSを特定し、さらに、このタイムスロットに割り当てられている基本サイクルの値を取得する。そして、スケジューリング制御部413は、目的TS割り当てプログラム検索処理を実施する（ステップS121）。

【0224】

より具体的には、スケジューリング制御部413は、ネクストTSに対応づけられているLP情報を参照し、このLP情報が示す候補プログラムを特定し、さらに、これらプログラムそれぞれに対応づけられている優先度を把握する。そして、スケジューリング制御部413は、優先度&モード情報検索処理を実施する（ステップS122）。

より具体的には、スケジューリング制御部413は、ネクストTSに対応づけられている優先度&モード情報を参照し、優先度として0から3のいずれが示されているのか、さらに、モードとしてIndexモード及びLevelモードのいずれが示されているのかを把握する。

。

【0225】

また、スケジューリング制御部413は、アイドルオプション情報検索処理を実施する（ステップS123）。

より具体的には、スケジューリング制御部413は、ネクストTSに対応づけられているアイドルオプション情報を参照し、「SWITCH」及び「MICRO-SLEEP」のいずれが示されているのかを判別する。つづいて、スケジューリング制御部413は、スケジューリング処理を実施する（ステップS124）。

【0226】

より具体的には、スケジューリング制御部413は、目的タイムスロット検索処理により得られたネクストTSに対応づけられている4つのプログラムを特定し、目的TS割り当てプログラム検索処理により、これらプログラムそれぞれに対応づけられている優先度を把握する。そして、スケジューリング制御部413は、優先度&モード情報検索処理により得られた、優先度及びモードにもとづいて、上記4つのプログラムから候補プログラムを選出する。

【0227】

さらに、スケジューリング制御部413は、LP状態格納レジスタ424内の各候補プログラムに対応するプログラム状態情報を参照し、各候補プログラムがそれぞれ実行可能であるか否かを検出し、候補プログラムから有効候補プログラムを選出する。このとき、有効候補プログラムが複数ある場合、優先度の値が最も小さな値に対応づけられている有効候補プログラムをこのネクストTSへの割り付け対象とする。

【0228】

また、有効候補プログラムが1つも存在しない場合には、スケジューリング制御部413は、アイドルオプション情報検索処理を実施することにより得られる情報の内容にもと

10

20

30

40

50

づいて、その後の処理の内容を決定する。即ち、上述の情報の内容が、「MICRO-SLEEP」を示す場合、スケジューリング制御部 4 1 3 は、いずれのプログラムも上述のネクスト T S への割り当て対象としない。

【 0 2 2 9 】

この場合、スケジューリング制御部 4 1 3 は、このネクスト T S が実行対象となっている期間、処理実行部 4 1 0 に N O P 実行させる共に、クロックマスク部 4 2 7 に処理実行部 4 1 0 へのクロックパルスの供給を遮断させる。

また、上述の情報の内容が、「SWITCH」を示す場合、スケジューリング制御部 4 1 3 は、ネクストアドレスの値に 1 を加えた値（ただし、ネクストアドレスが終端アドレスであった場合に限り、先頭アドレスの値）を新たなネクストアドレスとし、ステップ S 1 2 0 から S 1 2 4 のスケジューリング処理を繰り返し実施する。

10

【 0 2 3 0 】

以上の処理が終了したとき、割込み処理要求が発生しているか否かを判別する上述のステップ S 1 0 6 へと移行する。なお、目的 T S をカレント T S とする場合の動作については、後述する。図 1 9 は、スケジューリング制御部 4 1 3 により実施されるイベントドリブンスケジューリングの内容を示すフローチャートである。

スケジューリング制御部 4 1 3 は、リクエスト検出部 4 1 9 から受け取ったイベント P - I D によりプログラムを特定する。

【 0 2 3 1 】

また、スケジューリング制御部 4 1 3 は、実行中のプログラムがイベントドリブンスケジューリングにより緊急レジスタ切り替えすることで中断させられるとき、中断させられるこのプログラムを特定する（ステップ S 1 3 0 ）。

20

さらに、スケジューリング制御部 4 1 3 は、L P 状態格納レジスタ 4 2 4 を参照し、このプログラムの属性、状態、起床タイミングを判別する。そして、スケジューリング制御部 4 1 3 は、以下に示す（ 1 ）から（ 3 ）のいずれかに該当しているか否かを判断する（ステップ S 1 3 1 ）。

（ 1 ）プログラムの属性が E D である。

（ 2 ）プログラムの属性が T D であって、プログラム状態が suspended waiting であり、かつ、起床タイミングが「 n 」である。

（ 3 ） running 状態であったプログラムがイベントドリブンスケジューリングにより緊急レジスタ切り替えされた。

30

【 0 2 3 2 】

この判定の結果、該当しなかった場合、つまり、「いいえ」のとき、スケジューリング制御部 4 1 3 は、L P 状態格納レジスタ 4 2 4 を参照し、このプログラムの起床タイミングが「 e 」であるか否かを判別し（ステップ S 1 3 2 ）、「 e 」でない場合、即ち、「 n 」である場合、回復イベント処理を実施する。より具体的には、スケジューリング制御部 4 1 3 は、L P 状態格納レジスタ 4 2 4 に示されているこのプログラムの状態が waiting であれば ready に変更する。

【 0 2 3 3 】

この後、タイムドリブンスケジューリングを実施する上述のステップ S 1 0 4 へと移行する。

40

また、ステップ S 1 3 2 における判別結果が、「 e 」である場合、スケジューリング制御部 4 1 3 は、先に述べた緊急回復イベント処理を実施し（ステップ S 1 3 4 ）、先に述べた緊急復帰処理を実施し（ステップ S 1 3 5 ）、さらに、先に述べた緊急レジスタ切り替え処理を実施した後（ステップ S 1 3 6 ）、上述のステップ S 1 0 0 へと移行させる。

【 0 2 3 4 】

また、上述のステップ S 1 3 1 において、判定の結果、該当した場合、即ち、「はい」の場合、スケジューリング制御部 4 1 3 は、L P 状態格納レジスタ 4 2 4 を参照し、このプログラムの起床タイミングが「 e 」であるか否かを判別し（ステップ S 1 3 7 ）、「 e 」でない場合、即ち、「 n 」である場合、先に述べたイベント起床処理を実施し（ステッ

50

プ S 1 4 1)、先に述べた早期復帰処理を実施し (ステップ S 1 4 2)、上述のステップ S 1 0 7 へと移行させる。

【 0 2 3 5 】

また、上述のステップ S 1 3 7 における判別において、起床タイミングが「 e 」である場合、スケジューリング制御部 4 1 3 は、先に述べた緊急イベント起床処理を実施し (ステップ S 1 3 8)、先に述べた緊急復帰処理を実施し (ステップ S 1 3 9)、さらに、先に述べた緊急レジスタ切り替え処理を実施した後 (ステップ S 1 4 0)、上述のステップ S 1 0 0 へと移行させる。

【 0 2 3 6 】

図 2 0 は、上述のステップ S 1 0 8 において実施される割込み処理の内容を示すフローチャートである。スケジューリング制御部 4 1 3 は、リクエスト検出部 4 1 9 から受け取る固有 P __ I D、固有 S P __ I D 及び共通 S P __ I D にもとづいて、共通割込み処理の要求が発生しているか否かを判別する (ステップ S 1 5 0)。

共通割込み処理の要求が発生している場合、即ち、リクエスト検出部 4 1 9 から共通 S P __ I D を受け取った場合、スケジューリング制御部 4 1 3 は、この共通 S P __ I D により特定されるサブルーチンプログラムが実行されている間、サイクルカウンタ 4 1 6 におけるカウントダウンを中断させる (ステップ S 1 5 1 ~ ステップ S 1 5 2) 一方、共通割込み処理の要求が発生しておらず、固有割込み処理の要求が発生している場合、つまり、リクエスト検出部 4 1 9 から固有 P __ I D 及び固有 S P __ I D を受け取った場合、スケジューリング制御部 4 1 3 は、この固有 S P __ I D により特定されるサブルーチンプログラムが実行されている間、サイクルカウンタ 4 1 6 におけるカウントダウンについては特に何もしない (ステップ S 1 5 3 ~ ステップ S 1 5 4)。

【 0 2 3 7 】

以上の処理が実行された後、上述のステップ S 1 0 9 へと移行する。図 2 1 は、上述のステップ S 1 1 0 において実施される放棄スケジューリングの内容を示すフローチャートである。スケジューリング制御部 4 1 3 は、DMA 管理部 4 1 8 からキャッシュミス信号を受け取ったか否かを判別し (ステップ S 1 6 0)、キャッシュミス信号を受け取っている場合、実行中のプログラムのプログラム状態を suspended-waiting に変更し (ステップ S 1 6 1)、先に述べたタイムオプション情報検索処理を実施する (ステップ S 1 6 2)。

【 0 2 3 8 】

また、キャッシュミス信号を受け取っていない場合、スケジューリング制御部 4 1 3 は、さらに、「vmp_sus」コードを受け取っているか否かを判別し (ステップ S 1 6 3)、「vmp_sus」コードを受け取っている場合、実行中のプログラムのプログラム状態を suspended-waiting に変更し (ステップ S 1 6 1)、先に述べたタイムオプション情報検索処理を実施する (ステップ S 1 6 2)。

【 0 2 3 9 】

上述のステップ S 1 6 3 における判別において、「vmp_sus」コードを受け取っていない場合、必然的に「vmp_wait」コードを受け取っているので、スケジューリング制御部 4 1 3 は、実行中のプログラムのプログラム状態を waiting に変更し (ステップ S 1 6 6)、先に述べたタイムオプション情報検索処理を実施する (ステップ S 1 6 2)。

スケジューリング制御部 4 1 3 は、このタイムオプション情報検索処理を実行することにより、タイムオプション情報の内容を認識する。そして、スケジューリング制御部 4 1 3 は、タイムオプション情報の内容が「MICRO-SLEEP」を示すか否かを判別し、(ステップ S 1 6 5) タイムオプション情報の内容が「MICRO-SLEEP」である場合、処理制御部 4 1 1 を介して処理実行部 4 1 0 に nop 実行させ (ステップ S 1 6 6)、上述のステップ S 1 0 2 へと移行させる。

【 0 2 4 0 】

また、タイムオプション情報の内容が「MICRO-SLEEP」でない場合、スケジューリング制御部 4 1 3 は、さらに、タイムオプション情報の内容が「TS_SWITCH」を示すか否かを

10

20

30

40

50

判別し、「TS_SWITCH」を示す場合、緊急レジスタ切り替え処理を実施し（ステップS 168）、上述のステップS 100へと移行させる。

また、タイムオプション情報の内容が「TS_SWITCH」を示していない場合、タイムオプション情報の内容は、必然的に「LP_SWITCH」となるので、スケジューリング制御部413は、ここでは、目的TSをカレントTSとし（ステップS 169）、先に述べたタイムドリブンスケジューリングを実施する（ステップS 170）。

【0241】

ここで、ステップS 170において実施するタイムドリブンスケジューリングの内容について、ステップS 105において実施するタイムドリブンスケジューリングの内容との相違点を挙げつつ説明する。上記2つのタイムドリブンスケジューリングの内容について、以下の2つの違いがある。

10

即ち、ステップS 170において実施するタイムドリブンスケジューリングでは、目的TSがネクストTSではなくカレントTSであるという点と、スケジューリング処理を実施した結果、有効候補プログラムが存在しないときにアイドルオプション情報とは無関係にNOP実行へと移行するという点である。より具体的には、図18（B）ステップS 169の処理に引き続き、スケジューリング制御部413は、目的タイムスロット検索処理を実施する（ステップS 120）。

【0242】

つまり、スケジューリング制御部413は、時間割付情報格納レジスタ402内のカレントアドレスに格納されているタイムスロット情報を参照し、タイムスロットを特定するTS-IDによりカレントTSを特定し、さらに、このタイムスロットに割り当てられている基本サイクルの値を取得する。そして、スケジューリング制御部413は、目的TS割り当てプログラム検索処理を実施し（ステップS 121）、優先度&モード情報検索処理を実施する（ステップS 122）。

20

【0243】

つづいて、スケジューリング制御部413は、アイドルオプション情報検索処理を実施し（ステップS 123）、スケジューリング処理を実施する（ステップS 124）。より具体的には、スケジューリング制御部413は、目的タイムスロット検索処理により得られたカレントTSに対応付けられている4つのプログラムを特定し、目的TS割り当てプログラム検索処理により、これらプログラムそれぞれに対応づけられている優先度を把握する。

30

【0244】

そして、スケジューリング制御部413は、優先度&モード情報検索処理により得られた、優先度及びモードにもとづいて、上記4つのプログラムから候補プログラムを選出する。さらに、スケジューリング制御部413は、LP状態格納レジスタ424内の各候補プログラムに対応するプログラム状態情報を参照し、各候補プログラムがそれぞれ実行可能であるか否かを検出し、候補プログラムから有効候補プログラムを選出する。

【0245】

このとき、有効候補プログラムが複数ある場合、優先度の値が最も小さな値が対応付けられている有効候補プログラムをこのカレントTSへの割り付け対象とする。もし、有効候補プログラムが1つも存在しない場合には、取得したアイドルオプションに関わりなく有効候補プログラムが存在しないことを記憶する。

40

以上の処理が、目的TSをカレントTSとするステップS 170におけるタイムドリブンスケジューリングである。このようなタイムドリブンスケジューリングを実行した後、スケジューリング制御部413は、有効候補プログラムがあるか否かを判断する（ステップS 171）。

【0246】

有効候補プログラムがない場合、スケジューリング制御部413は、処理制御部411を介して処理実行部410にnop実行させると共に（ステップS 166）、nop実行させている間、クロックマスク部427に処理実行部410へのクロック供給を遮断する

50

ように指示し、上述のステップ S 1 0 2 へと移行させる。

この指示の結果、処理実行部 4 1 0 へのクロックの供給は、遮断される。一方、有効候補プログラムがある場合、復帰処理を実施し（ステップ S 1 7 2 ）、緊急レジスタ切り替え処理を実施して（ステップ S 1 6 8 ）、上述のステップ S 1 0 0 へと移行させる。

（４．プロセッサ駆動時におけるプログラム実行状況）以下、仮想マルチプロセッサ 4 0 0 の駆動時におけるプログラム実行状況について説明する。

（４－１．基本的なプログラム実行状態）

仮想マルチプロセッサ 4 0 0 では、先に述べたように、複数のタイムスロット及び各タイムスロットに割り当てられたプログラムにもとづいてプログラム実行される。

【 0 2 4 7 】

10

図 2 2 は、このタイムスロットについて説明する図である。以下、便宜上、各タイムスロットを、T S n（n は、0 から 1 5 までの整数）と記し、この n の値は、タイムスロット毎に固有であるものとする。設定上 1 6 個のタイムスロット用意されており、このうち 4 つのタイムスロット（T S 0 ～ T S 3 ）が通常のスケジューリングに用いられるタイムドリブン型のタイムスロットである。

【 0 2 4 8 】

つまり、T S 0 から T S 3 までの各タイムスロットに割り当てられている 1 つのプログラムが順次実行され、T S 3 のタイムスロットに割り当てられているプログラムの実行が終了すると、また、T S 0 のタイムスロットに割り当てられているプログラムから実行されることとなる。T S 0 から T S 2 までのタイムスロットは、映像や音声などを対象とするリアルタイムでの処理を前提とするプログラム実行用のタイムスロットである。

20

【 0 2 4 9 】

また、T S 3（TS_ID=3）は、リアルタイムでの処理を前提としないプログラム実行用のタイムスロットである。その他のタイムスロット（T S 4 ～ T S 1 5 ）は、イベントが発生したときに起床するイベントドリブン型のタイムスロットである。図 2 2（a）に示すように、各タイムスロットは、基本サイクルの値が対応づけられており、この基本サイクルの値に相当する時間が、プログラム実行時間枠となる。

【 0 2 5 0 】

図 2 2（b）に示すように、切り替えのタイミングをタイムスロットの基本サイクルと厳密に一致させることはできず、通常、数サイクル程度オーバーして切り替わる。仮想マルチプロセッサ 4 0 0 では、先に述べた基本サイクル調整処理を実施することにより、T S 0 から T S 3 までのタイムスロットにおいて、実行時間枠からオーバーした時間を T S 3 のタイムスロットの実行時間枠から差し引き、実行時間に破綻を来たさないようにする。

30

【 0 2 5 1 】

また、仮想マルチプロセッサ 4 0 0 では、実行予定のプログラムに割り当てられているプログラム実行時間が放棄されている場合、効率的なプログラム実行又は省電力化を図るための措置の内容を決定するものとして、タイムオプション情報及びアイドルオプション情報の 2 つの情報がある。

（４－２．放棄時におけるタイムオプション情報にもとづくプログラム実行状態）タイムオプション情報とは、先に述べたように、プログラム実行中にこのプログラムが実行不可能となり、このプログラムの実行時間が放棄された以後のプログラム実行の内容を決定するタイムスロットについての情報である。

40

【 0 2 5 2 】

図 2 3 は、プログラム実行時間が放棄された場合のタイムオプション情報にもとづくプログラム実行状態、即ち、タイムオプション情報検索処理によるプログラム実行状態について説明する図である。以下、便宜上、各プログラムを、L P m（m は、0 から 3 1 までの整数）と記し、この m の値は、プログラム毎に固有であるものとする。

図 2 3（a）に示すように、タイムドリブン型のタイムスロットである T S 0、T S 1、T S 2 及び T S 3 において、有効な候補プログラムがある場合、各タイムスロットの実

50

行時間枠内で順次プログラムが実行される。ここで、TS0にはLP0が、TS1にはLP1が、TS2にはLP2が、TS3にはLP3がそれぞれ割り当てられている。

【0253】

図23(b)に示すように、TS1の実行時間枠においてLP1の実行中に、処理実行部410による「vmp_wait」コード出力をトリガとして、プログラム実行時間の放棄が行なわれ、かつ、このTS1のタイムオプション情報が「LP_SWITCH」を示す場合には、先に述べた各機能部における処理が実行されることにより、以下のようにプログラム実行される。

【0254】

即ち、このTS1の実行時間枠を使いきるまで、LP1に代わって、同じTS1に対応づけられ実行可能な状態となっている有効候補プログラムであって、最も優先度が高い候補プログラムであるLP4が実行される。このとき、スタンバイレジスタ群に格納されていたLP2に対応するコンテキストのレジスタ値は一旦破棄され、LP4に対応するコンテキストのレジスタ値が格納される。

【0255】

このような破棄やLP4の復帰を行い、LP4の実行が開始されるまでにt2の時間を要する。通常、このt2は、サイクルカウンタ416によるパルス出力のおよそ30サイクル分に相当する時間である。ところで、TS1に有効候補プログラムが1つもないとき、もはやこのタイムスロットにおいてのプログラム実行は不可能となり、このような場合、スケジューリング制御部413は、後述のTS1のタイムオプション情報が「MICRO-SLEEP」を示す場合におけるプログラム実行と同様のプログラム実行を行うように制御するものとする。

【0256】

なお、有効候補プログラムが1つもないために、実行対象範囲外となったTS1のようなタイムスロットをスリーピングタイムスロットという。上述のプログラム実行時間の放棄が行なわれ、かつ、このTS1のタイムオプション情報が「TS_SWITCH」を示す場合には、以下のようにプログラム実行される。

即ち、図23(c)に示すように、速やかにレジスタ切り替えが行われ、次のTS2に割り当てられているLP2が繰り上げられて実行される。このとき、スタンバイレジスタ群には、既に、LP2が格納されているため、LP2が速やかに実行される。

【0257】

また、上述のプログラム実行時間の放棄が行なわれ、かつ、このTS1のタイムオプション情報が「MICRO-SLEEP」を示す場合には、図23(d)に示すように、このTS1の実行時間枠を使いきるまで、本来実行するプログラムとは無関係な意味のないプログラムの実行、即ち、nop実行される。

なお、上述のnop実行されている間、処理実行部410へのクロックの出力が遮断され、その結果、駆動電力が低減される。

(4-3. 放棄時におけるアイドルオプション情報にもとづくプログラム実行状態) アイドルオプション情報とは、先に述べたように、あるタイムスロットにおいて有効な候補プログラムが1つも存在しない場合におけるプログラム実行の内容を決定するタイムスロットについての情報である。

【0258】

図24(a)、(b)、(c)は、タイムドリブン型のタイムスロットにおいて、有効な候補プログラムが1つもいない場合のアイドルオプション情報にもとづくプログラム実行の状況を説明する図である。図24(a)は、タイムドリブン型の各タイムスロットにおいて、有効な候補プログラムが少なくとも1つある場合のプログラム実行状況を示す。

このような状況を前提とし、以下、TS1において、有効な候補プログラムが1つも存在しないこととなった場合のプログラム実行について説明する。TS1において、有効な候補プログラムが1つも存在せず、かつ、このTS1のアイドルオプション情報が「SWITCH」を示す場合には、先に述べた各機能部における処理が実行されることにより、図24

10

20

30

40

50

(b) に示すように、TS1 に代わり、TS2 に割り当てられているLP2 が実行される。

【0259】

また、TS1 において、有効な候補プログラムが1つも存在せず、かつ、このTS1 のアイドルオプション情報が「MICRO-SLEEP」を示す場合には、図24(c) に示すように、このTS1 の実行時間枠を使いきるまでnop実行される。なお、上述のnop実行されている間、処理実行部410、即ち、CPUへのクロックパルスの出力が遮断され、その結果、駆動電力が低減される。

(4-4. ウェイト状態解消時におけるプログラム実行状態) vmp_susコード出力をトリガとして、実行中であったプログラムは実行不可能な中断状態(suspended waiting)となる。

10

【0260】

また、緊急レジスタ切り替えをともなつて他のプログラムにより中断されたときに、実行中であったプログラムは実行可能な中断状態(suspended ready)となる。これらのプログラムが実行中に割り付けられていたタイムスロットを中断タイムスロットといい、上述したとおりプログラムのP_IDがTS_IDに関連付けられて記憶されている。

上述の実行不可能な中断状態となったプログラムは、ウェイト状態が解消したときに有効候補プログラムとなる。

【0261】

また、他のプログラムにより中断させられたプログラムは、実行可能な中断状態となったときに有効候補プログラムとなる。以下、一旦中断させられたプログラムが有効候補プログラムとなり、中断タイムスロットに割り付けられてタイムスロットの途中から実行を再開する状況について説明する。

20

図25(a)、(b)は、このような状況におけるプログラムの実行状況を説明する図である。図25(a)は、TS0 からTS3 の4つのタイムスロットのうち、既にTS3 が、中断タイムスロットとなっており、処理実行部410からの「vmp_sus」コード出力をトリガとして、TS0 に割り当てられているLP0 が有効でなくなり、新たにTS0 が中断タイムスロットとなった場合のプログラムの実行状況を示す。

【0262】

このとき、TS0 においては、設定されている実行時間枠に相当する時間t3 が使いきられていない。「vmp_sus」コードを受け取った場合、スケジューリング制御部413は、LP0 のプログラム状態をsuspended-waitingに更新し、実行時間枠内において使われずに残った時間t4を実カウント値としてコンテキスト内に保存する。

30

そして、図25(a)に示すように、TS1 及びTS2 のみが繰り返し実行対象となる。図25(b)は、中断タイムスロットとなったTS0 に割り当てられているLP0 にウェイト状態の解消、つまり、実行不可能な中断状態(suspended-waiting)の解消が生じ、かつ、LP状態格納レジスタ424においてLP0 の起床タイミングが「n」、即ち通常に起床させる旨を示す場合におけるプログラムの実行状況を示す。

【0263】

この場合、スケジューリング制御部413は、図25(b)に示すように、実行不可能な中断状態の解消が生じたとき、LP0 のプログラム状態をsuspended-readyに更新するが、即座にLP0 を実行させようとせず、カウント値0 信号を待って、他のTSでのプログラム実行を中断させないタイミングで、LP0 を実行するように制御する。

40

これにより、実行不可能な中断状態の解消が生じた時点において、既に、スタンバイレジスタ群に格納されているレジスタ値が使用され、この格納が無駄とならない。このとき、スケジューリング制御部413は、コンテキスト内に保存されている実カウント値分の時間t4だけ、LP0 を実行するように制御する。

【0264】

なお、このLP0 の実行は、イベントドリブンスケジューリングによりなされる。そして、これにつづいて、通常のタイムドリブンスケジューリングにより実行対象となったT

50

S 0 に割り当てられている L P 0 が t 3 だけ実行されることとなる。

以下、上述のケースにおいて、L P 状態格納レジスタ 4 2 4 における L P 0 の起床タイミングが「e」を示している場合における、ウェイトの解消時、つまり、実行不可能な中断状態が解消したときのプログラム実行状況について説明する。

(4 - 5 . 緊急的な復帰イベント発生時におけるプログラム実行状態) 図 2 6 (a) は、図 2 5 (a) と同様に、T S 0 から T S 3 の 4 つのタイムスロットのうち、既に T S 3 が、中断タイムスロットとなっており、処理実行部 4 1 0 からの「vmp_sus」コード出力をトリガとして、T S 0 に割り当てられている L P 0 が有効でなくなり、新たに T S 0 が中断タイムスロットとなった場合のプログラムの実行状況を示す。

【 0 2 6 5 】

このとき、T S 0 においては、設定されている実行時間枠 t 5 が全て使い切られていない。

また、図 2 5 (a) と同様に、「vmp_sus」コードを受け取った場合、スケジューリング制御部 4 1 3 は、L P 0 のプログラム状態をsuspended-waitingに更新し、実行時間枠内において使われずに残った時間t 6 を実カウント値としてコンテキスト内に保存する。

【 0 2 6 6 】

図 2 6 (b) は、スリーピングタイムスロットとなった T S 0 に割り当てられている L P 0 にウェイトの解消、つまり、実行不可能な中断状態の解消が生じ、かつ、L P 状態格納レジスタ 4 2 4 において L P 0 の起床タイミングが「e」、即ち緊急的に起床させる旨を示している場合におけるプログラムの実行状況を示す。

この場合、スケジューリング制御部 4 1 3 は、図 2 6 (b) に示すように、ウェイトの解消、つまり、実行不可能な中断状態の解消が生じたとき、L P 0 のプログラム状態をsuspended-readyに更新し、即座に L P 0 を実行させようとする。ここで上述の即座とは、復帰処理やレジスタ切り替えなどに要するオーバーヘッド (t 2) でのプログラム実行を意味する。

【 0 2 6 7 】

より具体的には、スケジューリング制御部 4 1 3 は、ウェイトの解消、つまり、実行不可能な中断状態の解消が生じた時点において、既に、スタンバイレジスタ群に格納されているレジスタ値を破棄し、このスタンバイレジスタ群に L P 0 に対応するコンテキストのレジスタ値を格納するという緊急復帰処理を実施し、緊急レジスタ切り替え処理を実行し、直ちに L P 0 を上述の t 6 の時間分実行させると共に、L P 1 のプログラム状態をsuspended-readyに更新し、T S 1 の実行時間枠内において使われずに残った時間t 7 を実カウント値として L P 1 のコンテキスト内に保存する。

【 0 2 6 8 】

上述のレジスタ切り替え処理を実行によりスタンバイレジスタ群となったレジスタ群には、それまで実行されていた L P 1 に対応するコンテキストのレジスタ値が存在するが、スケジューリング制御部 4 1 3 は、この L P 1 のレジスタ値を退避し、L P 0 の実行に伴い復帰処理を実行し、L P 0 の実行が終了後、レジスタ切り替え処理を実施して、直ちに L P 1 を上述の t 7 の時間分実行させる。

【 0 2 6 9 】

なお、この L P 1 の実行も、イベントドリブンスケジューリングによりなされ、具体的には、イベント起床処理 (起床タイミング「n」) により実行再開される。

(4 - 6 . イベント処理要求が発生した場合におけるプログラム実行状態) イベント処理要求が発生しない場合、図 2 7 (a) に示すように、T S 0、T S 1、T S 2 及び T S 3 が順次実行対象となり、これらタイムスロットそれぞれに割り当てられたプログラム L P 0、L P 1、L P 2 及び L P 3 が逐次実行される。

【 0 2 7 0 】

図 2 7 (b) は、上述の状態を前提として、イベント用のタイムスロットである T S 4 の L P 4 を実行するイベント処理要求が発生し、かつ、L P 状態格納レジスタ 4 2 4 において L P 4 の起床タイミングが「n」、即ち通常に起床させる旨を示す場合におけるプロ

10

20

30

40

50

グラムの実行状況を示す。この場合、スケジューリング制御部 413 は、図 27 (b) に示すように、イベント処理要求が発生したとき、LP4 のプログラム状態を waiting から ready に更新するが、即座に LP4 を実行させようとせず、イベント処理要求が発生したときにカレントレジスタ群となっているレジスタ群が、スタンバイレジスタ群となり、かつ、このレジスタ群が開放されたとき、このレジスタ群に LP4 に対応するコンテキストのレジスタ値を格納し、このレジスタ群がカレントレジスタ群となったときに、LP4 を実行するようにする。

【0271】

これにより、イベント処理要求が発生した時点において、既に、スタンバイレジスタ群に格納されているレジスタ値が使用され、この格納が無駄とならない。図 27 (c) は、イベント用のタイムスロットである TS4 の LP4 を実行するイベント処理要求が発生し、かつ、LP 状態格納レジスタ 424 において LP4 の起床タイミングが「e」、即ち緊急的に起床させる旨を示している場合におけるプログラムの実行状況を示す。

【0272】

この場合、スケジューリング制御部 413 は、図 27 (c) に示すように、イベント処理要求が発生したとき、LP4 のプログラム状態を ready に更新し、即座に、即ち、上述の t2 で LP4 を実行させようとする。より具体的には、スケジューリング制御部 413 は、イベント処理要求が発生した時点において、既に、スタンバイレジスタ群に格納されているレジスタ値を破棄し、このスタンバイレジスタ群に LP4 に対応するコンテキストのレジスタ値を格納し、緊急レジスタ切り替え処理を実行し、直ちに LP4 を実行させると共に、実行が中断された LP1 のプログラム状態を suspended-ready に更新し、TS1 の実行時間枠内において使われずに残った時間、即ち、実カウント値を LP1 のコンテキスト内に保存する。

【0273】

上述のレジスタ切り替え処理を実行によりスタンバイレジスタ群となったレジスタ群には、それまで実行されていた LP1 に対応するコンテキストのレジスタ値が存在し、スケジューリング制御部 413 は、このレジスタ値を一旦退避するそして、再度このレジスタ値をスタンバイレジスタ群に復帰して、LP4 の実行が終了後、直ちにレジスタ切り替え処理を実施して、直ちに LP1 を上述の使われずに残った時間分実行させるように制御する。

【0274】

なお、この LP1 の実行も、イベントドリブンスケジューリングによりなされ、具体的には、イベント起床処理（起床タイミング「n」）により実行再開される。

（4-7．共通割り込み処理要求が発生した場合におけるプログラム実行状況）図 28 (a)、(b) は、共通割り込み処理要求が発生した場合におけるプログラム実行状況を説明する図である。

【0275】

図 28 (a) に示すように、共通割り込み処理要求が発生していない場合、TS0 から TS3 の 4 つのタイムスロットが順次実行対象のタイムスロットとなり、これらタイムスロットそれぞれに割り当てられている LP0、LP1、LP2 及び LP3 が逐次実行される。TS2 における LP2 実行中に共通割り込み処理要求が発生した場合、図 28 (b) に示すように、スケジューリング制御部 413 は、LP2 のプログラム実行中にこの共通割り込み処理専用サブルーチンにジャンプして共通割り込み処理を実行させ、この共通割り込み処理が完了した後、LP2 に戻ってプログラム実行させる。

【0276】

このとき、スケジューリング制御部 413 は、上述のサブルーチンにおけるプログラム実行時間を、割り込まれた LP2 の実行時間に含めないように制御する。より具体的には、スケジューリング制御部 413 は、上述のサブルーチンにおいて、共通割り込み処理が実行されている間、サイクルカウンタ 416 におけるカウントダウンをストップさせる。

この結果、TS2 の実行時間枠 t9 に上述のサブルーチンにおいてプログラムが実行さ

10

20

30

40

50

れていた時間 t_{10} を加えた時間が $LP2$ の実行時間となる。このように、共通割り込み処理の実行時間を、割り込まれた $LP2$ の本来の実行時間に含めないようにするのは、全てのプログラムが共通割り込み処理の実行を必要としているにもかかわらず、偶然に割り込まれたある1つのタイムスロットの実行時間枠を消費することにより、このタイムスロットに割り当てられているプログラムだけ実行時間が減少するという不公平が生じることを是正するためである。

(4-8. 固有割り込み処理要求が発生した場合におけるプログラム実行状況)

図29(a)、(b)、(c)は、固有割り込み処理要求が発生した場合におけるプログラム実行状況を説明する図である。

【0277】

図29(a)に示すように、固有割り込み処理要求が発生していない場合、 $TS0$ から $TS3$ の4つのタイムスロットが順次実行対象のタイムスロットとなり、これらタイムスロットそれぞれに割り当てられている $LP0$ 、 $LP1$ 、 $LP2$ 及び $LP3$ が逐次実行される。図29(b)に示すように、 $TS2$ における $LP2$ 実行中に $LP3$ に対応する固有割り込み処理要求が発生した場合、スケジューリング制御部413は、次に $LP3$ が実行されるときに、この固有割り込み処理専用サブルーチンにジャンプして固有割り込み処理を実行させ、この固有割り込み処理が完了した後、 $LP3$ に戻ってプログラム実行させる。

【0278】

このとき、スケジューリング制御部413は、固有割り込み処理専用サブルーチンにおけるプログラム実行時間を、割り込まれた $LP3$ の実行時間に含める。つまり、スケジューリング制御部413は、上述のサブルーチンにおいて、固有割り込み処理が実行されている間、サイクルカウンタ416におけるカウントダウンをストップさせない。

このように、 $LP3$ に対応する固有割り込み処理の実行時間を、割り込まれた $LP3$ の本来の実行時間に含めるのは、 $LP3$ のみがこの固有割り込み処理の実行を必要としており、固有割り込み処理の実行を必要とするプログラムの実行時間から固有割り込み処理の実行時間が消費されるとしても、共通割り込み処理の場合のような実行時間についてのプログラム間における不公平が生じないためである。

【0279】

また、図29(c)に示すように、 $TS3$ における $LP3$ 実行中に $LP3$ に対応する固有割り込み処理要求が発生した場合、スケジューリング制御部413は、直ちにこの固有割り込み処理専用サブルーチンにジャンプして固有割り込み処理を実行させ、この固有割り込み処理が完了した後、 $LP3$ に戻ってプログラム実行させる。

このときも上述と同様に、スケジューリング制御部413は、固有割り込み処理実行用サブルーチンにおけるプログラム実行時間を、割り込まれた $LP3$ の実行時間に含める。以上のように、第2の実施形態によれば、1つの仮想マルチプロセッサ400において、擬似並行的に複数のプログラムを実行しても、コンテキストの退避、判断及び復帰は、プログラムの実行と並行して実施するため、プログラムの実行時間にコンテキストの退避、判断及び復帰に要する時間が介在することがないので、効率的にタスクを実行することができる。

【0280】

また、タイムスロット毎に複数のプログラムが用意されているため、1つのプログラムが実行できない状態となっても、そのプログラムに代わって別のプログラムが実行することができ、プログラムの実行時間を有効に使用することができる。さらに、プログラムが実行できない状態となった場合、そのプログラムの実行時間において、処理実行部410にNOP実行させている状態で駆動を停止させることにより消費電力を低減することができる。

【0281】

なお、本実施形態では、スケジューリング制御部413は、固有割り込み処理実行用サブルーチンにおけるプログラム実行時間を、割り込まれたプログラムの実行時間に含めるとしたが、これに限らず、例えば、以下のようにしてもよい。ここで、図30(a)に示す

10

20

30

40

50

ように、固有割込み処理が発生していない場合には、TS0からTS3の4つのタイムスロットが順次実行対象のタイムスロットとなり、これらタイムスロットそれぞれに割り当てられているLP0、LP1、LP2及びLP3が逐次実行されているものとする。

【0282】

各タイムスロットに割り当てられているLP0からLP3までのプログラムが実行されたとき、1巡目のプログラム実行が終了する。図30(b)に示すように、k巡目のLP2の実行中において、LP2に対応する固有割込み処理が発生した場合、スケジューリング制御部413は、このときに限り、固有割込み処理実行用サブルーチンにおけるプログラム実行時間t13を、割り込まれたプログラム(LP)の実行時間に含めない。

【0283】

つまり、スケジューリング制御部413は、固有割込み処理実行用サブルーチンにおいてプログラム実行している時間t13の間、サイクルカウンタ416におけるカウントダウンをストップさせる。そして、スケジューリング制御部413は、図30(c)に示すように、k+1順目において、TS2に割り当てられているLP2が実行されるとき、このTS2の実行時間枠t12から上述におけるプログラム実行時間t13を差し引いた時間分LP2を実行させる。

【0284】

より具体的には、スケジューリング制御部413は、サイクルカウンタ416に出力する基本サイクルの値に代えて、上述のt12からt13を差し引いた時間に相当する値をサイクルカウンタ416に出力する。なお、k+2巡目では、図30(a)に示すような通常の実行がなされる。

また、本実施形態では、図31(a)に示すように、クロックマスク部427が、スケジューリング制御部413からマスク信号を受信している間、処理実行部410へのクロックパルスの信号経路を遮断することにより、処理実行部410の電力消費を低減しているが、これに限らず、以下のようにしてもよい。

【0285】

即ち、図31(b)に示すように、処理実行部410への電力供給経路上に配置される電力制御部430が、スケジューリング制御部413から上述のマスク信号に代わる電力制御信号を受信している間、この電力供給経路を遮断することにより、処理実行部410における電力消費を0にしてもよい。

また、図31(c)に示すように、発振器600と処理実行部410との間にあるクロックパルスの信号経路上に配置される分周器などからなる分周部431が、スケジューリング制御部413から上述のマスク信号に代わる分周信号を受信している間、例えば、4分周することにより、処理実行部410の駆動周波数を下げて電力消費を低減してもよい。

【0286】

さらに、図31(d)に示すように、発振器600と仮想マルチプロセッサ400との間にあるクロックパルスの信号経路上に配置される分周器などからなる分周部431が、スケジューリング制御部413から上述のマスク信号に代わる分周信号を受信している間、このクロックパルスの周波数を、例えば、4分周することにより、処理実行部410の駆動周波数を下げて電力消費を低減してもよい。

【0287】

このとき、サイクルカウンタ416は、従来の機能を発揮する通常処理部432に加え、逓倍器などからなる逓倍部433を有し、クロックパルスの周波数を、例えば、4逓倍するなどして上述の分周による影響を相殺し、カウントダウンを行うとしてもよい。なお、上述の分周及び逓倍の値は、無論、4に限るものではない。

また、第2の実施形態では、リクエスト検出部419は、イベントP_IDの出力を行おうとすると、現在を起点として所定時間分過去に遡った期間における記憶された前記イベントP_IDの出力回数が、予め決められているしきい値を越えることとなる場合、前記出力回数が前記しきい値以下となるように上述の出力を遅延するとしたが、前記出力

10

20

30

40

50

回数が当該しきい値を越えることとなる場合、前記イベント P __ I D の出力を取り止める、即ち、イベント処理要求を無視するとしてもよい。

【 0 2 8 8 】

また、第 2 の実施形態では、スケジューリング制御部 4 1 3 は、スタンバイレジスタ群に格納されているコンテキストに対応するプログラムの実行に必要なデータをキャッシュメモリに書き込むように D M A 管理部 4 1 8 に指示するとしたが、これに代えて、ガーベジコレクションを実行する機能部に対し、ガーベジコレクションの実行を指示するとしてもよい。

【 0 2 8 9 】

また、本第 2 の実施形態では、或る時点において、2 以上のプログラムの状態が新たに ready 又は suspended-ready となった場合、即ち、イベントドリブンスケジューリングの候補内、つまり、スケジューリングの対象範囲内に、複数のプログラムが存在するとき、いずれのプログラムを先に実行するのかについては、記載していないが、プログラムに付与された優先度にもとづいて、優先度が最も高いものから先に実行対象としてもよい。

【 0 2 9 0 】

例えば、プログラム I D の値が小さいものほど、優先度が高いとする場合、プログラムの I D の小さなものから先に実行対象とすることとなる。

また、第 2 の実施形態では、プログラム (L P) の数は、レジスタ群の数を上回っていることを前提として説明していたが、これに限らず、プログラムの数がレジスタ群の数以下であってもよい。

【 0 2 9 1 】

その場合、コンテキストメモリ 5 0 0 を備えなくても良く、退避、復帰のステップが不要となる。プログラムの数が 2 つの場合、L P 0 は、第 1 レジスタ群に、L P 1 は、第 2 レジスタ群にそれぞれ対応付けられているものとする。例えば、スケジューリング制御部 4 1 3 は、L P 0 を実行するときは、コンテキストメモリからレジスタ値群を復帰する動作なしで、第 1 レジスタ群をカレントレジスタ群として選択する。

【 0 2 9 2 】

仮に、プログラム及びレジスタ群の数が 3 つの場合であっても、カレントレジスタ群として選択されているものが実行対象レジスタ群であって、それ以外がスタンバイレジスタ群となる。ここで、L P 2 は第 3 レジスタ群に対応付けられているものとする。L P 2 を実行するときは、第 3 レジスタ群をカレントレジスタ群として選択する。

選択先を切り替えた後も、コンテキストメモリにレジスタ値群を退避する動作は不要である。

【 0 2 9 3 】

また、コンテキストメモリ 5 0 0 は、S R A M により構成されたとしたが、D R A M であってもよく、さらに、これらメモリに限らず、高速にアクセスされるレジスタや低速にアクセスされる外部記憶装置であってもよい。

【図面の簡単な説明】

【 0 2 9 5 】

【図 1】第 1 の実施形態における仮想マルチプロセッサ 1 0 0 の機能ブロック図である。

【図 2】コンテキストメモリに格納されているテーブル及びコンテキストのデータ構成を示す図である。

【図 3】完了通知フラグ群及び設定フラグ群の各フラグについて説明する図である。

【図 4】タスクとプレートとの関係を示す図である。

【図 5】仮想マルチプロセッサにおいて実施される処理を示すフローチャートである。

【図 6】フラグ判定処理を示すフローチャートである。

【図 7】仮想マルチプロセッサにおいて実施される処理を時系列的に説明する図である。

【図 8】ポインタを用いて実行順序を示すテーブルのデータ構成を示す図である。

【図 9】スリープしているプレートのコンテキストに対して優先的にフラグ判定処理を実施する場合のプレート実行状況を時系列的に示す図である。

10

20

30

40

50

【図 10】遅くスリープされたプレートのコンテキストに対して優先的にフラグ判定処理を実施する例を時系列的に示す図である。

【図 11】コンテキストアドレスにカウント値の初期値が付加された実行順序テーブルのデータ構成を示す図である。

【図 12】サイクルカウンタの機能ブロック図である。

【図 13】4 段のパイプライン処理を実施する例を示す図である。

【図 14】スリープとなった直後から 1 回分のプレート実行までの間については、スリープしたプレートの実行順序を最先の順序とし、このスリープしたプレートを優先的に実行する場合のプレート実行の流れを示す図である。

【図 15】第 2 の実施形態における仮想マルチプロセッサの機能ブロック図である。

【図 16】時間割付情報格納レジスタ及び LP 状態格納レジスタに格納されている情報の内容を示す図である。

【図 17】第 2 の実施形態における仮想マルチプロセッサにおいて実施される処理を示すフローチャートである。

【図 18】スケジューリング制御部により実施されるタイムドリブンスケジューリングの内容を示すフローチャートである。

【図 19】スケジューリング制御部により実施されるイベントドリブンスケジューリングの内容を示すフローチャートである。

【図 20】割込み処理の内容を示すフローチャートである。

【図 21】放棄スケジューリングの内容を示すフローチャートである。

【図 22】(a) は、各タイムスロットの基本サイクルの値を示す図である。(b) は、基本サイクル調整処理による基本サイクルの値の見直しについて説明する図である。

【図 23】(a) は、プログラム実行時間が放棄されない通常のプログラム実行状態を示す図である。(b) は、プログラム実行時間が放棄され、タイムオプション情報が「LP_SWITCH」を示す場合におけるプログラム実行状態を示す図である。(c) は、プログラム実行時間の放棄が行なわれ、かつ、この LP 1 のタイムオプション情報が「TS_SWITCH」を示す場合におけるプログラム実行状態を示す図である。(d) は、プログラム実行時間の放棄が行なわれ、かつ、この LP 1 のタイムオプション情報が「MICRO-SLEEP」を示す場合におけるプログラム実行状態を示す図である。

【図 24】(a) は、タイムドリブン型のタイムスロットにおいて、有効な候補プログラムが 1 つもない場合のアイドルオプション情報にもとづくプログラム実行の状況を説明する図である。(b) は、有効な候補プログラムが 1 つも存在せず、かつ、この LP 1 のアイドルオプション情報が「SWITCH」を示す場合におけるプログラム実行状態を示す図である。(c) は、TS 1 において、有効な候補プログラムが 1 つも存在せず、かつ、この LP 1 のアイドルオプション情報が「MICRO-SLEEP」を示す場合におけるプログラム実行状態を示す図である。

【図 25】(a) は、スリーピングタイムスロットがある場合のプログラムの実行状況を示す。(b) は、スリーピングタイムスロットに割り当てられているプログラムにウェイト状態の解消が発生し、かつ、このプログラムの起床タイミングが「n」を示す場合におけるプログラムの実行状況を示す。

【図 26】(a) は、スリーピングタイムスロットがある場合のプログラムの実行状況を示す。(b) は、スリーピングタイムスロットに割り当てられているプログラムにウェイト状態の解消が発生し、かつ、このプログラムの起床タイミングが「e」を示す場合におけるプログラムの実行状況を示す。

【図 27】(a) は、イベント処理要求が発生しない場合におけるプログラム実行状態を示す図である。(b) は、イベント処理要求が発生し、かつ、このイベント処理を実行するプログラムの起床タイミングが「n」を示す場合におけるプログラムの実行状況を示す。(c) は、イベント処理要求が発生し、かつ、このイベント処理を実行するプログラムの起床タイミングが「e」を示す場合におけるプログラムの実行状況を示す。

【図 28】(a) は、共通割り込み処理要求が発生していない場合におけるプログラム実

10

20

30

40

50

行状態を示す図である。(b)は、共通割り込み処理要求が発生している場合におけるプログラム実行状態を示す図である。

【図29】(a)は、固有割り込み処理要求が発生していない場合におけるプログラム実行状態を示す図である。(b)は、固有割り込み処理要求が発生したとき、この固有割り込み処理を実行するプログラム以外が実行されている場合における固有割り込み処理の実行状態を示す図である。(c)は、固有割り込み処理要求が発生したとき、この固有割り込み処理を実行するプログラムが実行されている場合における固有割り込み処理の実行状態を示す図である。

【図30】(a)は、固有割り込み処理要求が発生していない場合におけるプログラム実行状態を示す図である。(b)は、k巡目に固有割り込み処理が発生した場合における、第1及び2の実施形態以外の方法によるk巡目の固有割り込み処理の実行状態を示す図である。(c)は、k巡目に固有割り込み処理が発生した場合における、第1及び2の実施形態以外の方法によるk+1巡目の固有割り込み処理の実行状態を示す図である。

10

【図31】(a)は、第2の実施形態における電力消費の低減化する機能部の構成を示す図である。(b)は、第2の実施形態以外の方法による電力消費の低減化する機能部の構成の一例を示す図である。(c)は、第2の実施形態以外の方法による電力消費の低減化する機能部の構成の一例を示す図である。(d)は、第2の実施形態以外の方法による電力消費の低減化する機能部の構成の一例を示す図である。

【図32】従来のプロセッサの機能ブロック図である。

【図33】従来のプロセッサにおけるタスク実行の流れ及びレジスタ群とコンテキストメモリとの間におけるコンテキストの入出力の状況を示す図である。

20

【図34】第1の実施形態における仮想マルチプロセッサ100と外部プロセッサ30の接続関係を示す図である。

【符号の説明】

【0296】

- 10 主メモリ
- 30 外部プロセッサ
- 100 仮想マルチプロセッサ
- 110 処理実行部
- 111 処理制御部
- 112 レジスタ切替部
- 113 切替入替制御部
- 114 レジスタ群
- 115 レジスタ群
- 116 サイクルカウンタ
- 117 外部I/F
- 118 DMA管理部
- 119 完了通知レジスタ
- 120 制御レジスタ
- 121 設定フラグレジスタ
- 200 コンテキストメモリ
- 201 コンテキスト格納領域
- 202 実行順序テーブル格納領域
- 301 入出力部
- 302 第1メモリ
- 303 第2メモリ
- 304 カウントダウン部
- 400 仮想マルチプロセッサ
- 402 時間割付情報格納レジスタ
- 410 処理実行部

30

40

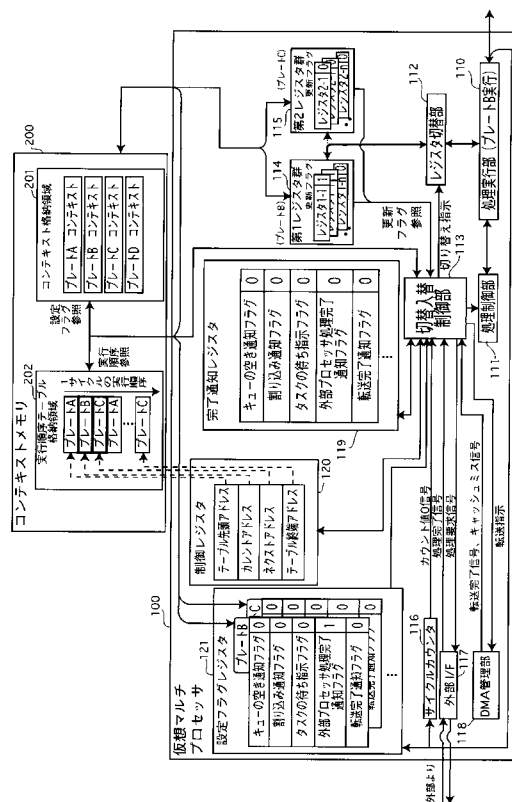
50

- | | |
|-------|----------------|
| 4 1 1 | 処理制御部 |
| 4 1 2 | レジスタ切替部 |
| 4 1 3 | スケジュールリング制御部 |
| 4 1 4 | レジスタ群 |
| 4 1 5 | レジスタ群 |
| 4 1 6 | サイクルカウンタ |
| 4 1 7 | 外部 I / F |
| 4 1 8 | D M A 管理部 |
| 4 1 9 | リクエスト検出部 |
| 4 2 0 | 制御レジスタ |
| 4 2 1 | 設定フラグレジスタ |
| 4 2 4 | L P 状態格納レジスタ |
| 4 2 5 | プロセッサ状態格納部 |
| 4 2 7 | クロックマスク部 |
| 4 3 0 | 電力制御部 |
| 4 3 1 | 逡倍部 |
| 4 3 2 | 通常処理部 |
| 4 3 3 | 分周部 |
| 5 0 0 | コンテキストメモリ |
| 5 0 1 | コンテキスト格納領域 |
| 6 0 0 | 発振器 |
| 6 0 1 | P L L |
| 7 0 0 | バス |
| 7 0 1 | 外部プロセッサ処理要求信号線 |
| 7 0 2 | 外部プロセッサ処理完了信号線 |

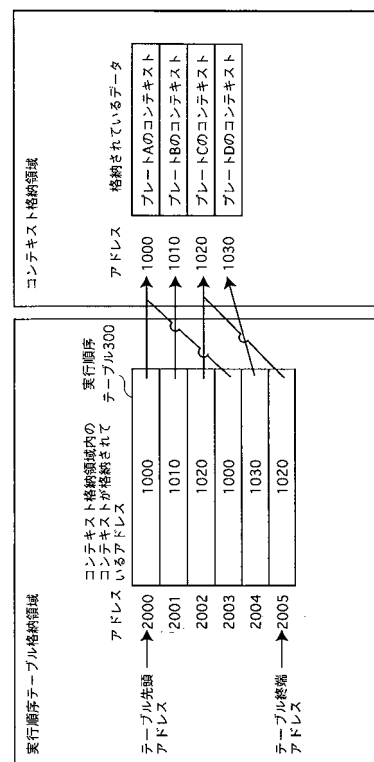
10

20

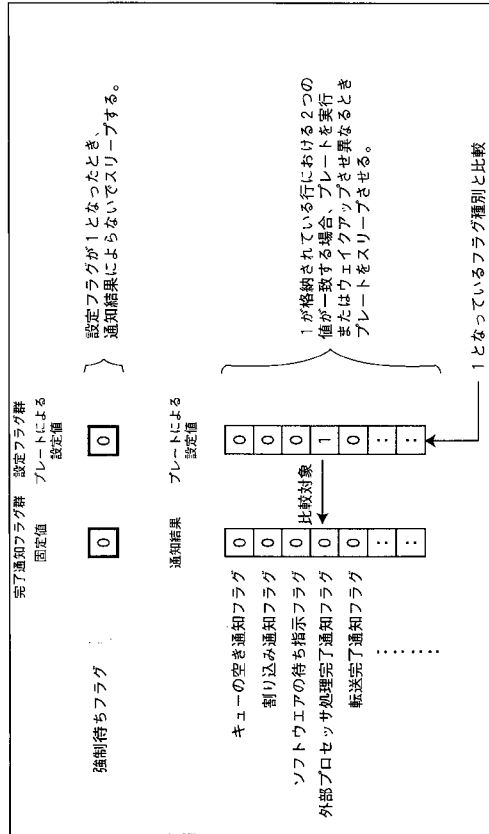
【 図 1 】



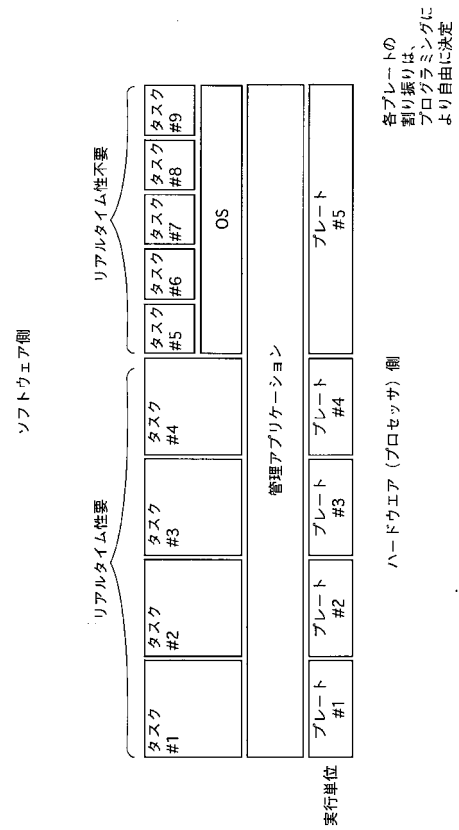
【圖 2】



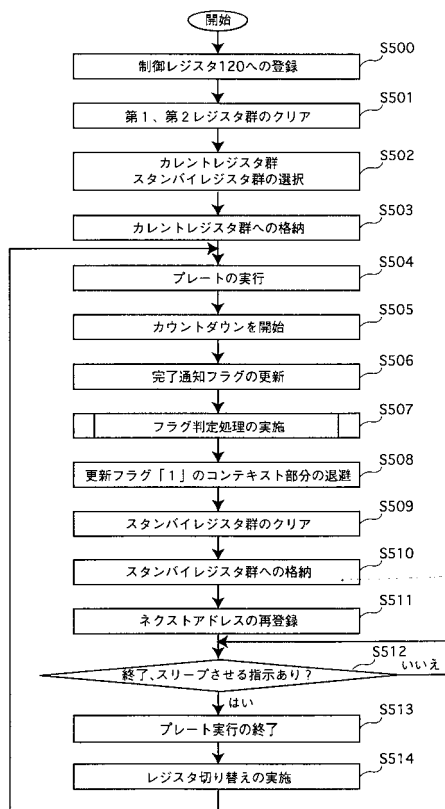
【図 3】



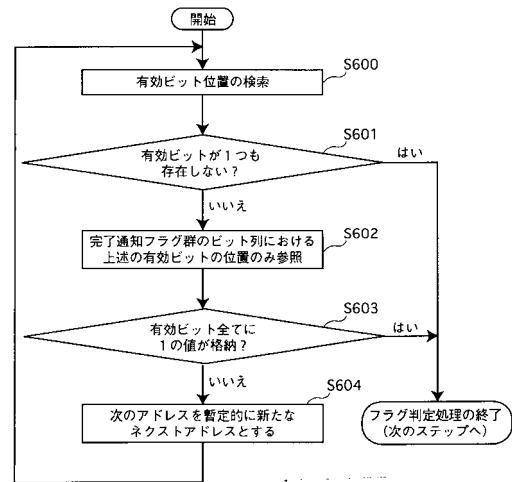
【図 4】



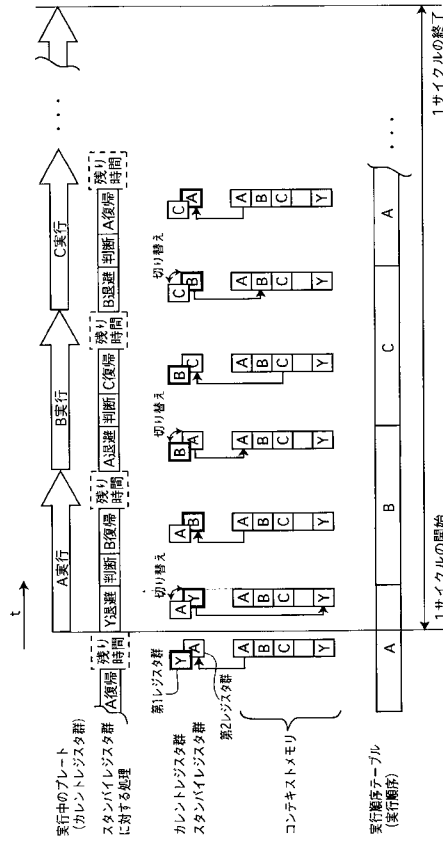
【図 5】



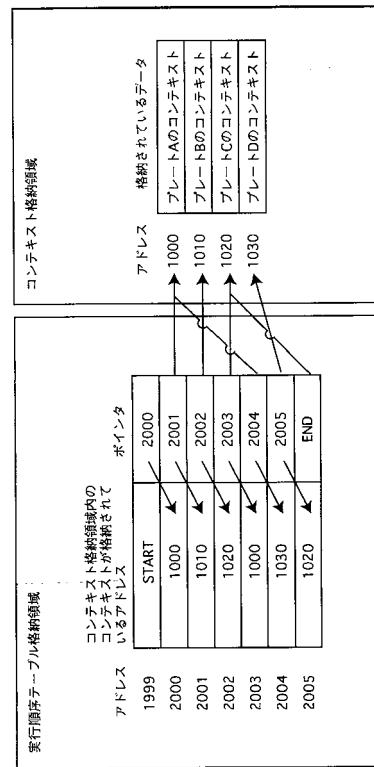
【図 6】



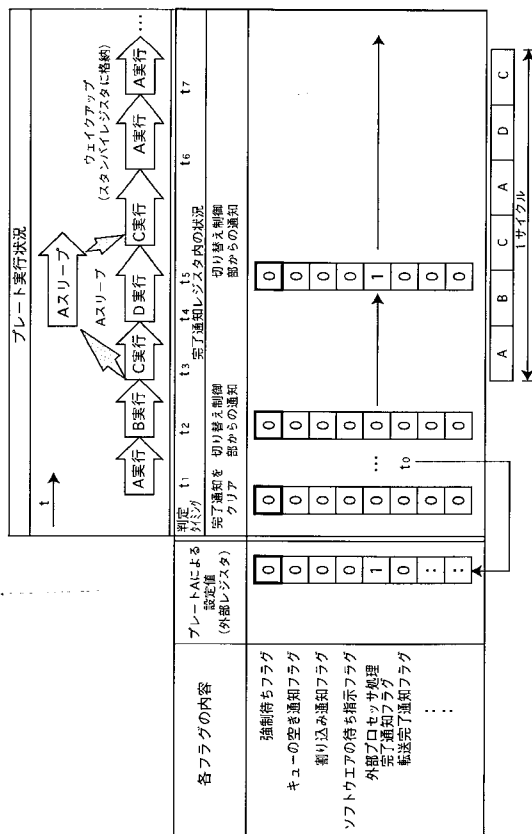
【図 7】



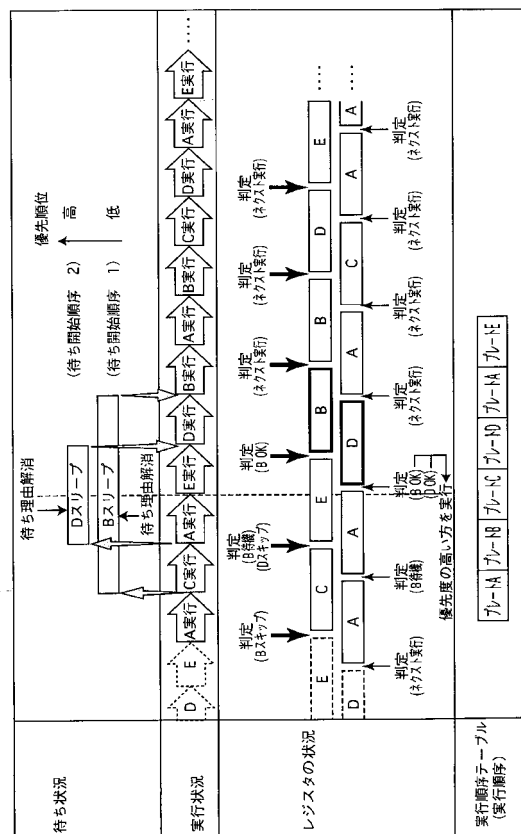
【図 8】



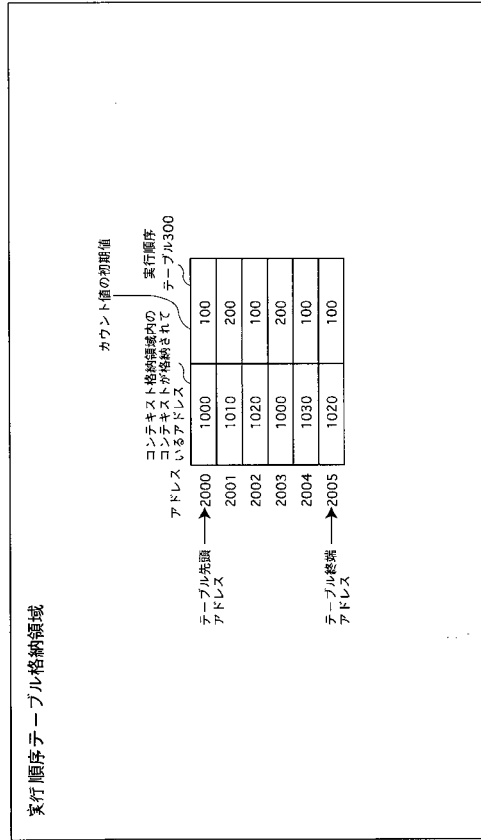
【図 9】



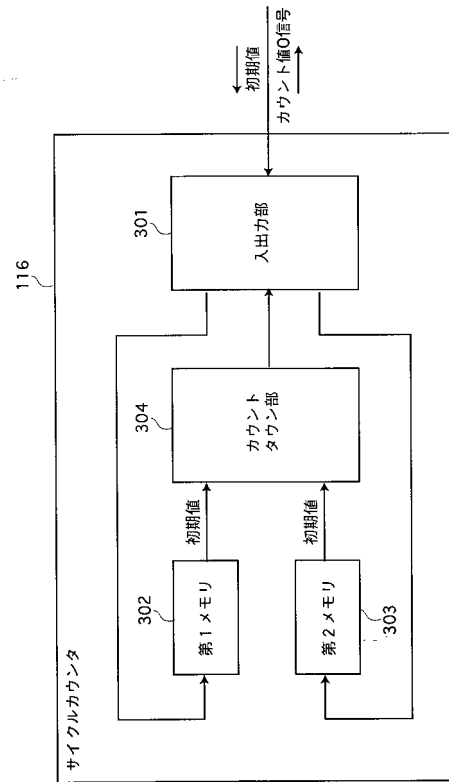
【図 10】



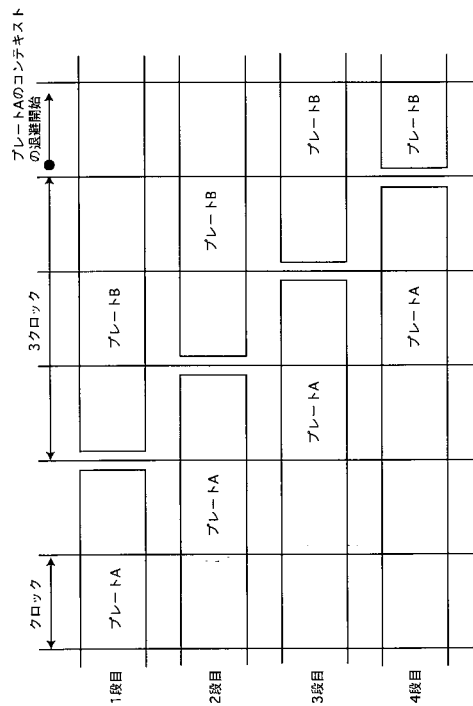
【図 1 1】



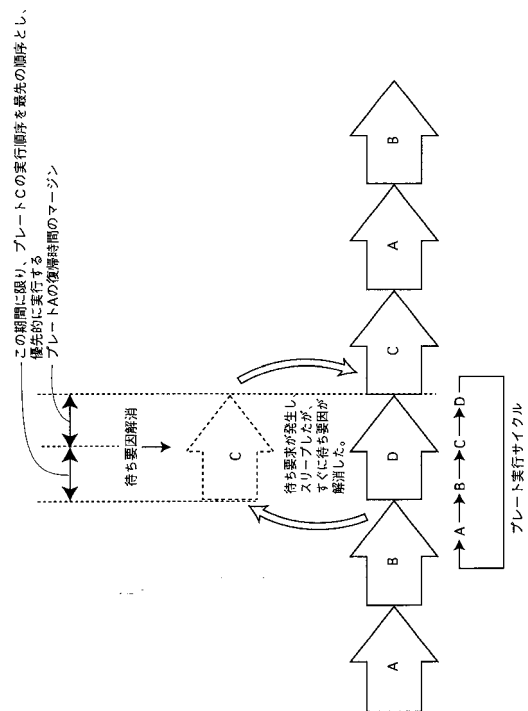
【図 1 2】



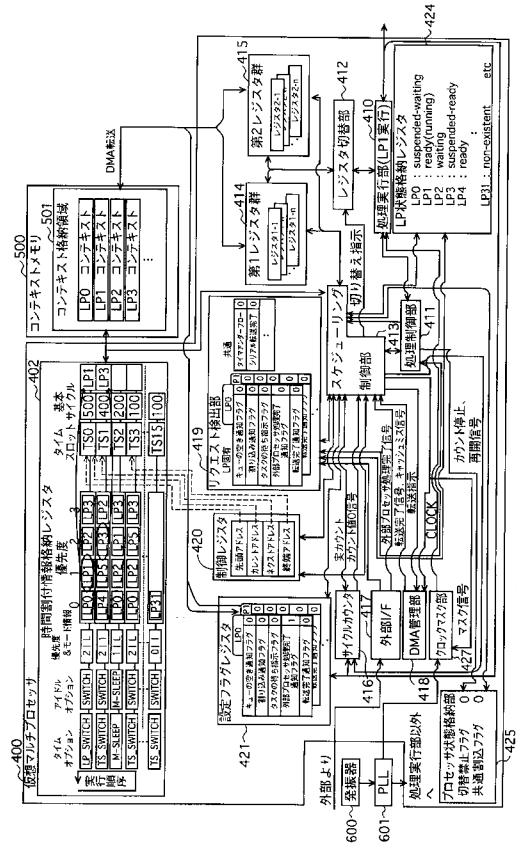
【図 1 3】



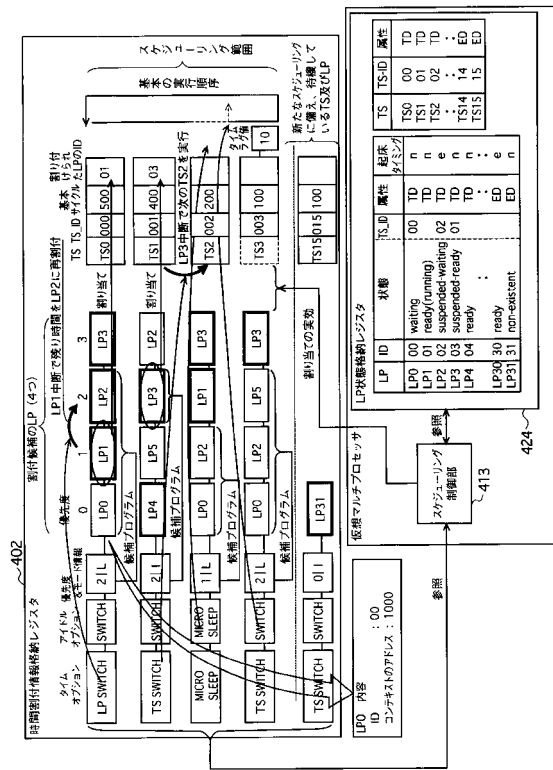
【図 1 4】



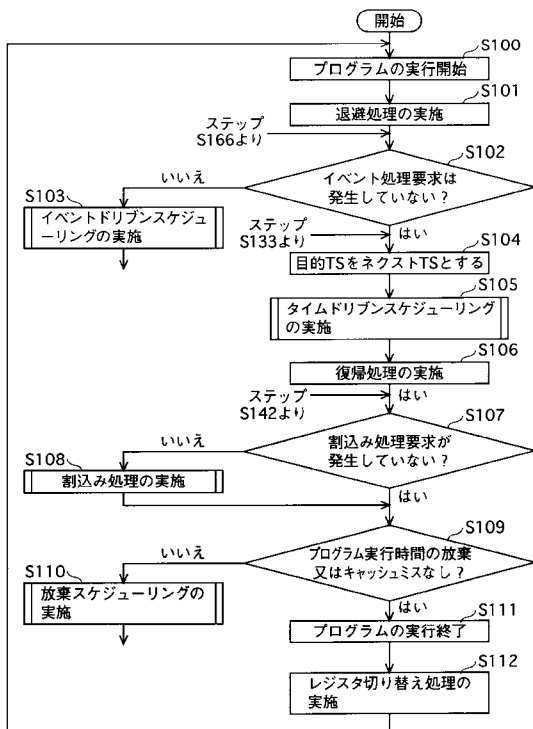
【図 15】



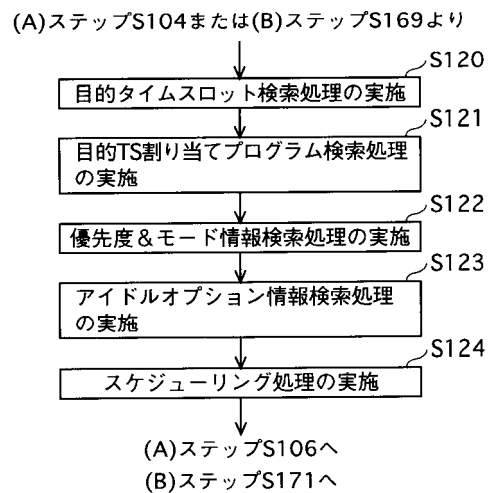
【図 16】



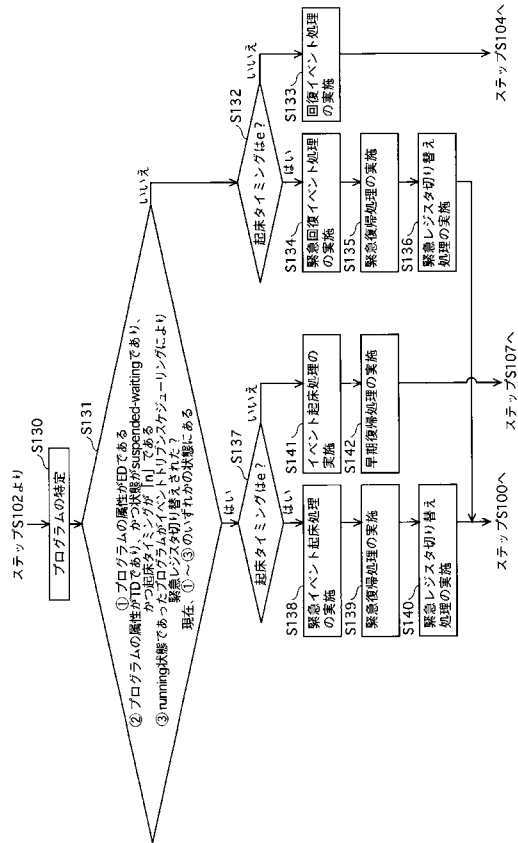
【図 17】



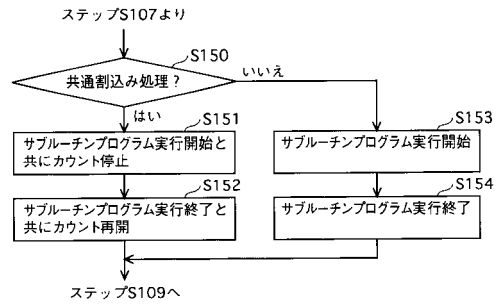
【図 18】



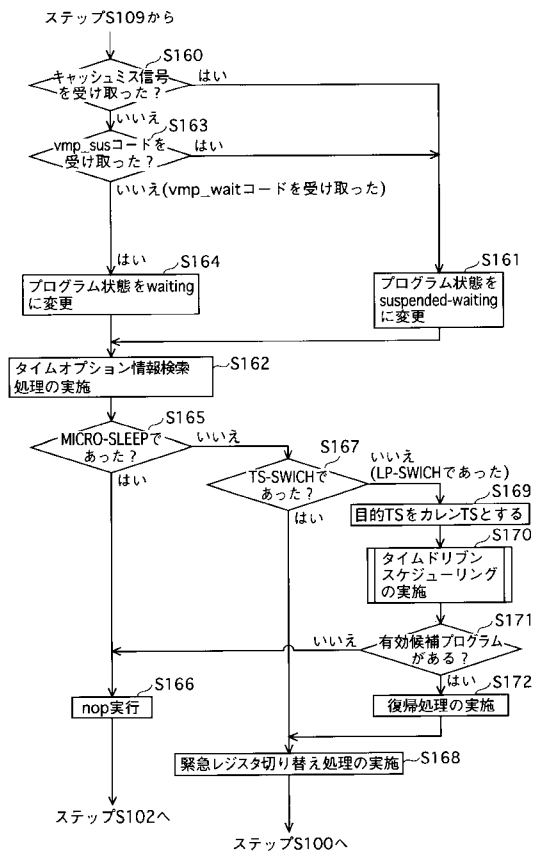
【図 19】



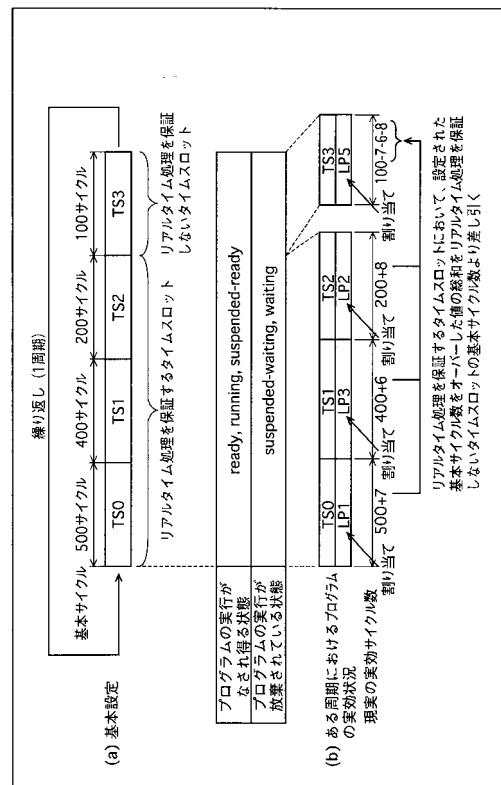
【図 20】



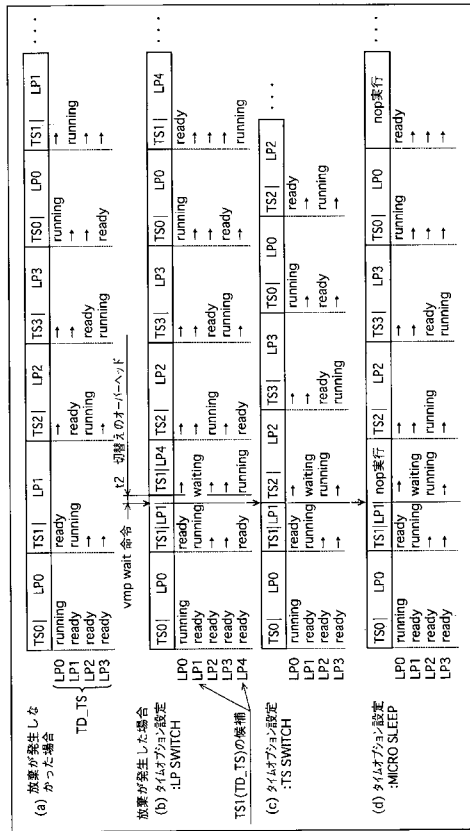
【図 21】



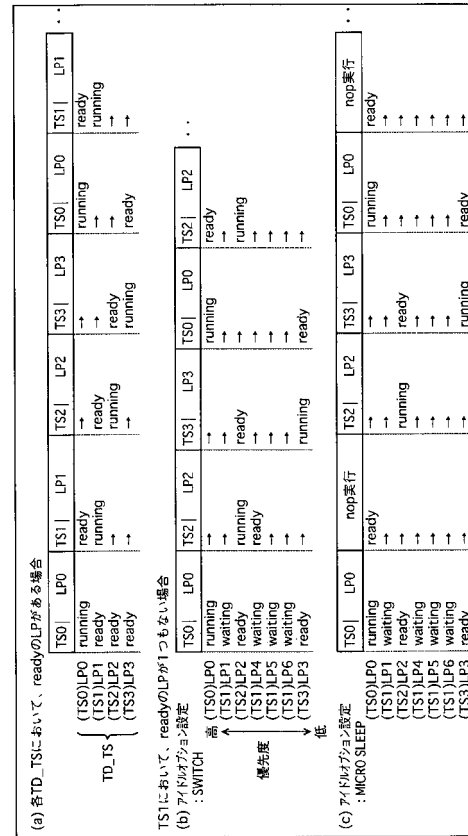
【図 22】



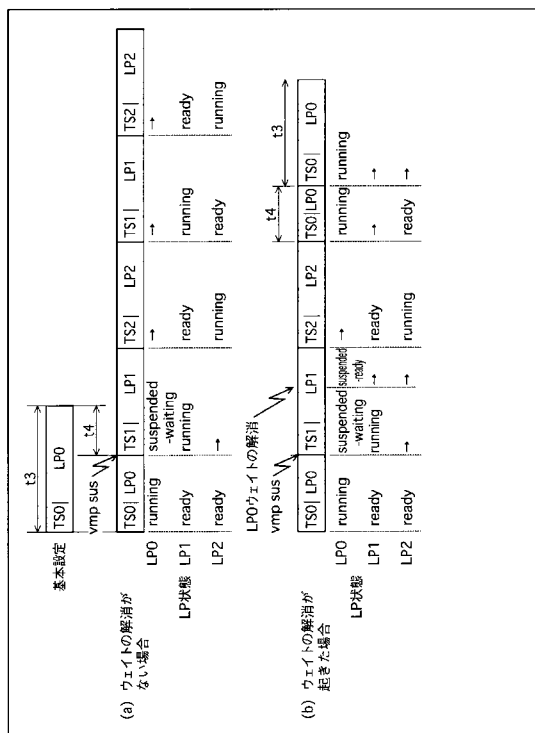
【図 2 3】



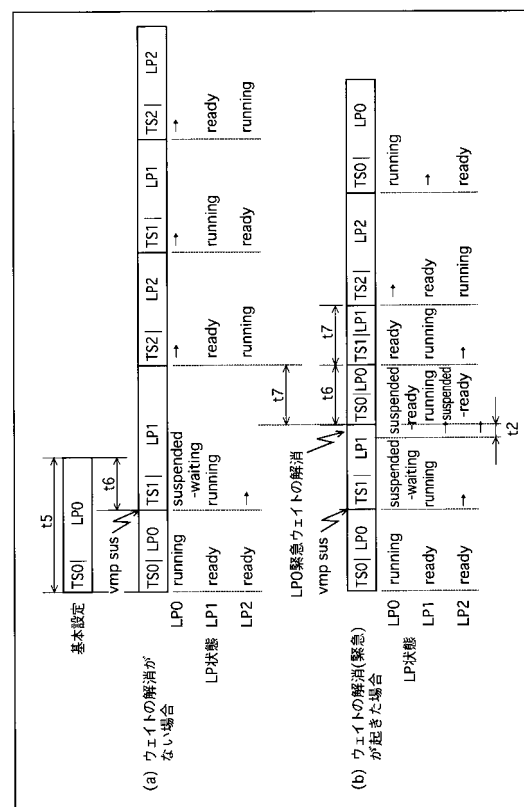
【図 2 4】



【図 2 5】



【図 2 6】



フロントページの続き

- (72)発明者 檜垣 信生
大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
- (72)発明者 清原 督三
大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
- (72)発明者 木村 浩三
大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
- (72)発明者 西田 英志
大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
- (72)発明者 杉村 敏夫
大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
- (72)発明者 廉田 浩
大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
- (72)発明者 古川 量也
大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
- (72)発明者 林 邦彦
大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

審査官 殿川 雅也

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

(58)調査した分野(Int.Cl. , D B 名)

G 0 6 F 9 / 4 6 - 9 / 5 4
G 0 6 F 1 / 0 4
G 0 6 F 1 / 3 2
G 0 6 F 9 / 4 2