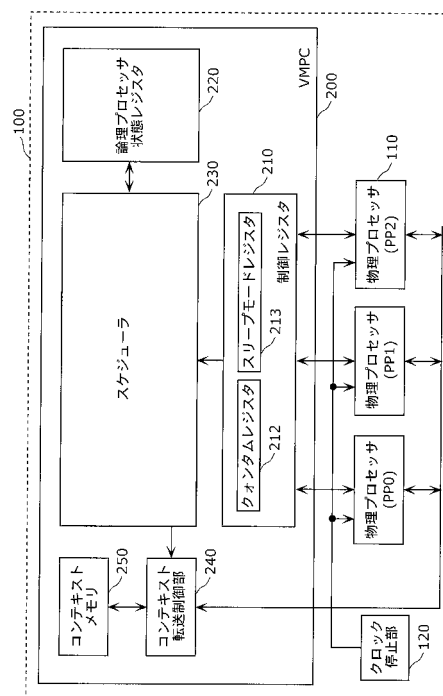




【特許請求の範囲】

【請求項 1】

複数のプログラムを割り当て時間ごとに切り替えながら実行する 1 以上のプロセッサと

、

前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定するスケジューリング部と、

前記複数のプログラムのそれぞれに対する、前記割り当て時間を保持する割り当て時間レジスタと、

第 1 モード又は第 2 モードが設定されるモードレジスタとを備え、

前記スケジューリング部は、前記モードレジスタに前記第 1 モードが設定されている場合、前記 1 以上のプロセッサにより実行されているプログラムの前記割り当て時間に依存するタイミングで、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定し、前記モードレジスタに前記第 2 モードが設定されている場合、前記割り当て時間に依存しないタイミングで、前記 1 以上のプロセッサのうち少なくとも 1 つのプロセッサがプログラムを実行しないように、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定する

ことを特徴とする仮想マルチプロセッサ。

【請求項 2】

前記仮想マルチプロセッサは、さらに、

前記モードレジスタに前記第 2 モードが設定されている場合、前記 1 以上のプロセッサのうち、プログラムを実行していないプロセッサへのクロックの供給を停止するクロック停止部を備える

ことを特徴とする請求項 1 に記載の仮想マルチプロセッサ。

【請求項 3】

前記 1 以上のプロセッサは、複数のプロセッサであり、

前記スケジューリング部は、前記モードレジスタに前記第 2 モードが設定されている場合、前記複数のプロセッサの一部のみがプログラムを実行するように、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定する

ことを特徴とする請求項 1 又は 2 に記載の仮想マルチプロセッサ。

【請求項 4】

前記スケジューリング部は、前記モードレジスタの設定が変更されたことを検知し、前記第 1 モードから前記第 2 モードに変更を検知した際に、前記割り当て時間に依存しないタイミングで、前記 1 以上のプロセッサのうち少なくとも 1 つのプロセッサがプログラムを実行しないように、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定する

ことを特徴とする請求項 3 に記載の仮想マルチプロセッサ。

【請求項 5】

前記スケジューリング部は、前記モードレジスタの設定が前記第 1 モードから前記第 2 モードに変更された際、前記複数のプロセッサに実行されているプログラムのうち一部のプログラムの実行を一時中断し、前記モードレジスタの設定が再度前記第 1 モードに変更された後に、一時中断した前記プログラムを優先して前記プロセッサに実行させる

ことを特徴とする請求項 4 に記載の仮想マルチプロセッサ。

【請求項 6】

前記複数のプログラムは、1 以上の第 1 プログラム及び 1 以上の第 2 プログラムを含み

、

前記スケジューリング部は、前記モードレジスタに前記第 1 モードが設定されている場合、前記 1 以上のプロセッサのそれぞれに、前記第 1 プログラム又は前記第 2 プログラムを実行させるように、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定し、前記モードレジスタの設定が前記第 1 モードから前記第 2 モードに変更された際、前記複数のプロセッサのうち前記第 1 プログラムを実行しているプロセッサに

当該第 1 プログラムの実行を継続させ、前記複数のプロセッサのうち前記第 2 プログラムを実行しているプロセッサに当該第 2 プログラムの実行を一時中断させる

ことを特徴とする請求項 4 又は 5 に記載の仮想マルチプロセッサ。

【請求項 7】

前記モードレジスタの設定は、前記複数のプロセッサにより実行される前記第 1 プログラムにより変更され、

前記スケジューリング部は、前記モードレジスタの設定が前記第 1 モードから前記第 2 モードに変更された際、当該変更を行った第 1 プログラムを実行しているプロセッサに当該第 1 プログラムの実行を継続させ、当該第 1 プログラム以外のプログラムを実行しているプロセッサに当該プログラムの実行を一時中断させる

10

ことを特徴とする請求項 6 に記載の仮想マルチプロセッサ。

【請求項 8】

前記仮想マルチプロセッサは、さらに、

複数の前記第 1 プログラムのそれぞれに対し第 1 の値又は第 2 の値が設定されるプログラム設定レジスタを備え、

前記スケジューリング部は、前記モードレジスタに前記第 2 モードが設定されている場合、前記複数のプロセッサの一部のみに、前記複数の第 1 プログラムのうち前記第 1 の値が設定されている前記プログラム設定レジスタに対応する複数の第 1 プログラムを切り替えながら実行させる

20

ことを特徴とする請求項 6 に記載の仮想マルチプロセッサ。

【請求項 9】

前記仮想マルチプロセッサは、さらに、

複数の前記第 1 プログラムのそれぞれに対し第 1 の値又は第 2 の値が設定されるプログラム設定レジスタと、

前記複数のプロセッサのうち前記第 2 モード時にプログラムを実行するプロセッサの数が設定されるプロセッサ数レジスタとを備え、

前記スケジューリング部は、前記モードレジスタに前記第 2 モードが設定されている場合、前記プロセッサ数レジスタに設定された前記数のプロセッサに、前記複数の第 1 プログラムのうち前記第 1 の値が設定されている前記プログラム設定レジスタに対応する複数の第 1 プログラムを実行させる

30

ことを特徴とする請求項 6 に記載の仮想マルチプロセッサ。

【請求項 10】

前記 1 以上のプロセッサは、1 個のプロセッサであり、

前記スケジューリング部は、前記モードレジスタに前記第 2 モードが設定されている場合、前記 1 個のプロセッサにプログラムを実行させない

ことを特徴とする請求項 1 又は 2 に記載の仮想マルチプロセッサ。

【請求項 11】

請求項 1 ～ 10 のいずれか 1 項に記載の仮想マルチプロセッサを備える

ことを特徴とするシステム L S I。

【請求項 12】

40

請求項 11 記載のシステム L S I を備える

ことを特徴とする携帯電話機器。

【請求項 13】

複数のプログラムを割り当て時間ごとに切り替えながら実行する 1 以上のプロセッサと、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定するスケジューリング部と、前記複数のプログラムのそれぞれに対する、前記割り当て時間を保持する割り当て時間レジスタと、第 1 モード又は第 2 モードが設定されるモードレジスタとを備える仮想マルチプロセッサの制御方法であって、

前記スケジューリング部は、前記モードレジスタに前記第 1 モードが設定されている場合、前記 1 以上のプロセッサにより実行されているプログラムの前記割り当て時間に依存

50

するタイミングで、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定し、

前記スケジューリング部は、前記モードレジスタに前記第2モードが設定されている場合、前記割り当て時間に依存しないタイミングで、前記1以上のプロセッサのうち少なくとも1つのプロセッサがプログラムを実行しないように、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定する

ことを特徴とする仮想マルチプロセッサの制御方法。

【発明の詳細な説明】

【技術分野】

【0001】

10

本発明は、仮想マルチプロセッサ、システムLSI、携帯電話機器、及び仮想マルチプロセッサの制御方法に関し、特に、複数のプログラムを切り替えながら実行する仮想マルチプロセッサに関する。

【背景技術】

【0002】

近年、携帯電話機器などの移動体通信機器の需要が急速に高まっている。これにより、移動体通信機器用のLSIの需要もますます高まっている。移動体通信機器は、主にバッテリーから電力が供給されるので、移動体通信機器用のLSIでは、消費電力を削減することが重要である。

【0003】

20

一方、プロセッサ開発の分野において、近年、特に仮想化技術が注目されている。プロセッサの仮想化を実現する手段は数多く開発されている。中でもIntel社により開発されたCPUの仮想化支援機能であるIntel Virtualization Technologyがよく知られている。また、本発明に関連する技術としては、特許文献1記載の技術が知られている。特許文献1には、スレッドを作成及び破棄する際のオーバーヘッドを低減する技術が記載されている。

【特許文献1】特表2007-504541号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

30

しかしながら、上記の従来の技術には下記の問題点がある。

【0005】

仮想マルチプロセッサ上で複数のプログラムを並列に実行する場合、通常、個々のプログラムを複数個実装されている物理プロセッサの中のどのプロセッサで実行させるかはハードウェアで構成されるスケジューラによって決定される。また、プログラムの実行の並列度を高めるという観点から、プログラムをどの物理プロセッサで実行するかは通常は指定しない。

【0006】

ところで、例えば仮想マルチプロセッサ上で携帯電話用アプリケーションプログラムを実行する場合、当該アプリケーションプログラムは、ホスト処理プログラムとメディア処理プログラムとに分類される。

40

【0007】

ホスト処理プログラムとは、主にシステム制御を行うプログラムであり、メディア処理プログラムとは、主に音声通話処理及び画像処理を行うプログラムである。

【0008】

仮想マルチプロセッサは、メディア処理プログラムを休止する期間が発生した場合、ホスト処理プログラムだけを実行し続け、メディア処理プログラムを実行しない制御を行う。このメディア処理プログラムを休止する期間においては、仮想マルチプロセッサの処理量が少なくなるため、複数の物理プロセッサのうち一部の物理プロセッサの動作を停止する。具体的には、動作を停止する物理プロセッサへのクロックの供給が停止される。これ

50

によって、L S I の消費電力が削減される。

【0009】

しかしながら、ホスト処理プログラム及びメディア処理プログラムが実行される物理プロセッサがハードウェアで構成されるスケジューラにより決定されるので、クロックの供給を停止する制御を低粒度で行うことが困難である。また、スケジューラは所定の時間毎にスケジューリングを行うので、一部の物理プロセッサの動作を停止する低電力モードに移行する際に、直ちにスケジューリングを行うことができない。つまり、従来の仮想マルチプロセッサでは、低電力モードに移行する際のオーバーヘッドが大きいという問題がある。

【0010】

そこで、本発明は、低電力モードに移行する際のオーバーヘッドを削減できる仮想マルチ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】

【0011】

上記目的を達成するために、本発明に係る仮想マルチプロセッサは、複数のプログラムを割り当て時間ごとに切り替えながら実行する1以上のプロセッサと、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定するスケジューリング部と、前記複数のプログラムのそれぞれに対する、前記割り当て時間を保持する割り当て時間レジスタと、第1モード又は第2モードが設定されるモードレジスタとを備え、前記スケジューリング部は、前記モードレジスタに前記第1モードが設定されている場合、前記1以上のプロセッサにより実行されているプログラムの前記割り当て時間に依存するタイミングで、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定し、前記モードレジスタに前記第2モードが設定されている場合、前記割り当て時間に依存しないタイミングで、前記1以上のプロセッサのうち少なくとも1つのプロセッサがプログラムを実行しないように、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定する。

【0012】

この構成によれば、本発明に係る仮想マルチプロセッサが備えるスケジューリング部は第2モード（低電力モード）時には、プロセッサの一部にプログラムを実行させない。これにより、プログラムを実行していないプロセッサに対してクロックの供給を停止することで、消費電力を削減できる。例えば、本発明に係る仮想マルチプロセッサ上で携帯電話用アプリケーションプログラムを実行する場合、第2モードにおいては、スケジューリング部は、ホスト処理プログラムのみをプロセッサに実行させ、メディア処理を休止させることができる。

【0013】

さらに、本発明に係る仮想マルチプロセッサは、第2モードへの移行時には、プログラムの割り当て時間に依存しないタイミングで、スケジューリングを行う。これにより、本発明に係る仮想マルチプロセッサは、第2モードへの移行時に直ちにスケジューリングを行えるので、第2モードに移行する際のオーバーヘッドを削減できる。

【0014】

また、前記仮想マルチプロセッサは、さらに、前記モードレジスタに前記第2モードが設定されている場合、前記1以上のプロセッサのうち、プログラムを実行していないプロセッサへのクロックの供給を停止するクロック停止部を備えてもよい。

【0015】

この構成によれば、本発明に係る仮想マルチプロセッサは、動作していないプロセッサに対してクロックの供給を停止することで、消費電力を削減できる。

【0016】

また、前記1以上のプロセッサは、複数のプロセッサであり、前記スケジューリング部は、前記モードレジスタに前記第2モードが設定されている場合、前記複数のプロセッサの一部のみがプログラムを実行するように、前記複数のプログラムの実行順序、及びプロ

10

20

30

40

50

グラムを実行するプロセッサを決定してもよい。

【0017】

また、前記スケジューリング部は、前記モードレジスタの設定が変更されたことを検知し、前記第1モードから前記第2モードに変更を検知した際に、前記割り当て時間に依存しないタイミングで、前記1以上のプロセッサのうち少なくとも1つのプロセッサがプログラムを実行しないように、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定してもよい。

【0018】

また、前記スケジューリング部は、前記モードレジスタの設定が前記第1モードから前記第2モードに変更された際、前記複数のプロセッサに実行されているプログラムのうち一部のプログラムの実行を一時中断し、前記モードレジスタの設定が再度前記第1モードに変更された後に、一時中断した前記プログラムを優先して前記プロセッサに実行させてもよい。

10

【0019】

この構成によれば、本発明に係る仮想マルチプロセッサは、第1モードから第2モードに遷移した後に、再び第1モードに遷移した場合に、元の第1モードで実行されていたプログラムを引き続き実行できる。

【0020】

また、前記複数のプログラムは、1以上の第1プログラム及び1以上の第2プログラムを含み、前記スケジューリング部は、前記モードレジスタに前記第1モードが設定されている場合、前記1以上のプロセッサのそれぞれに、前記第1プログラム又は前記第2プログラムを実行させるように、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定し、前記モードレジスタの設定が前記第1モードから前記第2モードに変更された際、前記複数のプロセッサのうち前記第1プログラムを実行しているプロセッサに当該第1プログラムの実行を継続させ、前記複数のプロセッサのうち前記第2プログラムを実行しているプロセッサに当該第2プログラムの実行を一時中断させてもよい。

20

【0021】

この構成によれば、第2モードが設定されている場合、スケジューリング部は、ホスト処理プログラム（第1プログラム）のみをプロセッサに実行させ、メディア処理プログラム（第2プログラム）を休止させることができる。

30

【0022】

また、前記モードレジスタの設定は、前記複数のプロセッサにより実行される前記第1プログラムにより変更され、前記スケジューリング部は、前記モードレジスタの設定が前記第1モードから前記第2モードに変更された際、当該変更を行った第1プログラムを実行しているプロセッサに当該第1プログラムの実行を継続させ、当該第1プログラム以外のプログラムを実行しているプロセッサに当該プログラムの実行を一時中断させてもよい。

【0023】

この構成によれば、本発明に係る仮想マルチプロセッサは、第2モードにおいて、ホスト処理プログラムのみを実行できる。

40

【0024】

また、前記仮想マルチプロセッサは、さらに、複数の前記第1プログラムのそれぞれに対し第1の値又は第2の値が設定されるプログラム設定レジスタを備え、前記スケジューリング部は、前記モードレジスタに前記第2モードが設定されている場合、前記複数のプロセッサの一部のみに、前記複数の第1プログラムのうち前記第1の値が設定されている前記プログラム設定レジスタに対応する複数の第1プログラムを切り替えながら実行させてもよい。

【0025】

この構成によれば、本発明に係る仮想マルチプロセッサは、第2モードにおいて、複数のホスト処理プログラムを実行できる。

50

【0026】

また、前記仮想マルチプロセッサは、さらに、複数の前記第1プログラムのそれぞれに対し第1の値又は第2の値が設定されるプログラム設定レジスタと、前記複数のプロセッサのうち前記第2モード時にプログラムを実行するプロセッサの数が設定されるプロセッサ数レジスタとを備え、前記スケジューリング部は、前記モードレジスタに前記第2モードが設定されている場合、前記プロセッサ数レジスタに設定された前記数のプロセッサに、前記複数の第1プログラムのうち前記第1の値が設定されている前記プログラム設定レジスタに対応する前記複数の第1プログラムを実行させてもよい。

【0027】

この構成によれば、本発明に係る仮想マルチプロセッサは、第2モードにおいて、複数の

10

のホスト処理プログラムを並列に複数のプロセッサで実行できる。

【0028】

また、前記1以上のプロセッサは、1個のプロセッサであり、前記スケジューリング部は、前記モードレジスタに前記第2モードが設定されている場合、前記1個のプロセッサにプログラムを実行させなくてもよい。

【0029】

また、本発明に係るシステムLSIは、前記仮想マルチプロセッサを備える。

【0030】

また、本発明に係る携帯電話機器は、前記システムLSIを備える。

【0031】

また、本発明に係る仮想マルチプロセッサの制御方法は、複数のプログラムを割り当て時間ごとに切り替えながら実行する1以上のプロセッサと、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定するスケジューリング部と、前記複数のプログラムのそれぞれに対する、前記割り当て時間を保持する割り当て時間レジスタと、第1モード又は第2モードが設定されるモードレジスタとを備える仮想マルチプロセッサの制御方法であって、前記スケジューリング部は、前記モードレジスタに前記第1モードが設定されている場合、前記1以上のプロセッサにより実行されているプログラムの前記割り当て時間に依存するタイミングで、前記複数のプログラムの実行順序、及びプログラムを実行するプロセッサを決定し、前記スケジューリング部は、前記モードレジスタに前記第2モードが設定されている場合、前記割り当て時間に依存しないタイミングで、前

20

30

【0032】

なお、本発明は、このような仮想マルチプロセッサとして実現できるだけでなく、仮想マルチプロセッサに含まれる特徴的な手段をステップとする仮想マルチプロセッサの制御方法として実現したり、そのような特徴的なステップをコンピュータに実行させるプログラムとして実現したりすることもできる。そして、そのようなプログラムは、CD-ROM等の記録媒体及びインターネット等の伝送媒体を介して流通させることができるのは言うまでもない。

40

【発明の効果】

【0033】

以上より、本発明は、低電力モードに移行する際のオーバーヘッドを削減できる仮想マルチプロセッサ、システムLSI、携帯電話機器、及び仮想マルチプロセッサの制御方法を提供できる。

【発明を実施するための最良の形態】

【0034】

以下、本発明に係る仮想マルチプロセッサの実施の形態について、図面を参照しながら詳細に説明する。

【0035】

50

(実施の形態 1)

本発明の実施の形態 1 に係る仮想マルチプロセッサでは、通常モード時には、スケジューラが複数のプロセッサの全てに、ホスト処理プログラム又はメディア処理プログラムを実行させ、低電力モード時には、スケジューラが複数のプロセッサの一部にホスト処理プログラムのみを実行させる。

【0036】

これにより、本発明の実施の形態 1 に係る仮想マルチプロセッサは、低粒度でクロック制御を行えるので、低電力モードに移行する際のオーバーヘッドを削減できる。

【0037】

まず、本発明の実施の形態 1 に係る仮想マルチプロセッサの構成を説明する。

10

【0038】

図 1 は、本発明の実施の形態 1 に係る仮想マルチプロセッサの構成を示す機能ブロック図である。

【0039】

図 1 に示す仮想マルチプロセッサ 100 は、複数のプログラムを並列に実行可能である。仮想マルチプロセッサ 100 は、例えば、携帯電話用アプリケーションプログラムを実行する。また、仮想マルチプロセッサ 100 は、消費電力を抑制した低電力モードであるスリープモードと、通常の消費電力で動作を行う通常モードとを有する。

【0040】

仮想マルチプロセッサ 100 は、複数の物理プロセッサ 110 と、仮想マルチプロセッサコントローラ（以下、VMPC）200 と、クロック停止部 120 とを備える。

20

【0041】

複数の物理プロセッサ 110（以下、PPとも記す。また、仮想マルチプロセッサ 100 が備える 3 つの PP をそれぞれ PP0 ～ PP2 と記す。）は、複数のプログラムを並列に実行する。また、複数の物理プロセッサ 110 は、複数のプログラムを一定時間ごとに切り替えながら実行する。

【0042】

このとき、各プログラムからはそれぞれが仮想的に 1 つの独立したプロセッサで実行されるように見える。この仮想的なプロセッサを論理プロセッサ（以下、LP）という。1 つのプログラムは 1 つの LP 上で実行される。1 つのプログラムが PP 上で実行されるとき、当該プログラムに対応する LP が PP にアサインされるという。

30

【0043】

LP は、実行状態（`running` 状態）と、実行可能状態（`ready` 状態）と、イベント待ち状態（`waiting` 状態）と、実行可能な中断状態（`suspended ready` 状態）と、イベント待ちかつ中断状態（`suspended ready` 状態）との 5 つの動作状態をもつ。

【0044】

`running` 状態は、プログラムが PP 上で実行されている状態である。`ready` 状態は、プログラムを実行可能であるが、当該プログラムを実行する PP が存在しないために待機している状態である。`waiting` 状態は、なんらかの外部イベント（DMA 転送の完了など）を待つ状態である。`suspended ready` 状態は、プログラムの実行が外部イベントにより一時的に中断されている状態である。`suspended waiting` 状態はプログラムの実行が外部イベントにより一時的に中断され、かつ外部イベントを待つ状態である。

40

【0045】

VMPC 200 は、物理プロセッサ 110 によるプログラムの実行を制御する。VMPC 200 は、制御レジスタ 210 と、論理プロセッサ状態レジスタ 220 と、スケジューラ 230 と、コンテキスト転送制御部 240 と、コンテキストメモリ 250 とを備える。

【0046】

論理プロセッサ状態レジスタ 220 は、LP の状態を管理するためのレジスタである。

50

図 2 は、論理プロセッサ状態レジスタ 220 の構成を示す図である。

【0047】

論理プロセッサ状態レジスタ 220 は、32 個のレジスタ LP0SR、LP1SR、・ ・ ・、LP31SR を含む。それぞれのレジスタ LPnSR (n = 0 - 31。また n を LP の ID という) は、LP がアサインされた PP を特定する情報と、当該 LP の動作状態とを保持する。

【0048】

レジスタ LPnSR の第 0 から第 2 ビットは、LP の動作状態を示す。レジスタ LPnSR の第 3 ビットから第 4 ビットは、対応する LP がどの PP にアサインされているかを示す。また、LP の ID により優先度が管理される。例えば、ID が小さい LP ほど優先度が高く、かつ実行が完了したプログラムに対応する LP の優先度は最低に設定される。

【0049】

論理プロセッサ状態レジスタ 220 は、スケジューラ 230 により更新される。

【0050】

制御レジスタ 210 は、スケジューラ 230 を制御するためのレジスタである。制御レジスタ 210 は、クオンタムレジスタ 212 と、スリープモードレジスタ 213 とを備える。

【0051】

クオンタムレジスタ 212 は、32 個のレジスタ LPnQTM R (n = 0 - 31) を含む。

【0052】

図 3 は、クオンタムレジスタ 212 の構成を示す図である。各レジスタ LPnQTM R は、対応する LP のクオンタム時間を保持する。クオンタム時間は、ユーザが各プログラムに付与した固有の時間であり、各 LP に対応するプログラムが PP で実行される時間である。言い換えると、クオンタム時間は、PP に LP が割り当てられる時間である。また、クオンタムレジスタ 212 により保持されるクオンタム時間は、PP で実行されるプログラムにより設定される。

【0053】

スケジューラ 230 は、LP に対応するプログラムの実行時間が、当該 LP に対応するクオンタム時間に等しくなったときに、予めスケジューリングした他の LP を PP にアサインすることにより、実行するプログラムを切り替える。具体的には、スケジューラ 230 は、プログラムの実行時間をカウントし、カウント値がクオンタム時間に等しくなったときに LP を切り替える。

【0054】

スリープモードレジスタ 213 は、スリープモード及び通常モードのうちいずれかが設定されるレジスタである。スリープモードレジスタ 213 は、PP で実行されるプログラムにより更新される。つまり、スリープモードレジスタ 213 は、PP で実行されるプログラムが、通常モードからスリープモードに遷移することを要求するためのレジスタである。

【0055】

図 4 は、スリープモードレジスタ 213 の構成を示す図である。例えば、スリープモードレジスタ 213 の第 0 ビットに 1 が保持されているとスリープモードであることを示し、0 が保持されていると通常モードであることを示す。スリープモードレジスタ 213 の第 0 ビットに 1 が書き込まれると、スケジューラ 230 はスリープモードに遷移するためのスケジューリングを行う。

【0056】

スケジューラ 230 は、複数のプログラムの優先度に従い、複数のプログラムの実行順序、及びプログラムを実行する PP を決定するスケジューリングを行う。

【0057】

通常モード時には、スケジューラ 230 は、複数の PP により実行されているプログラ

10

20

30

40

50

ムのうちいずれかのクォンタム時間に依存するタイミングで、スケジューリングを行う。具体的には、スケジューラ 230 は、プログラムの実行時間がクォンタム時間に等しくなる時刻から所定の時間前のタイミングで、スケジューリングを行い、当該プログラムの実行時間がクォンタム時間に等しくなった後に、PP により実行されるプログラムを切り替える。スケジューラ 230 は、動作状態が実行可能状態である LP のうち優先度が高い LP を、プログラムの実行時間がクォンタム時間に等しくなった PP に実行させるように、スケジューリングを行う。

【0058】

通常モード時には、スケジューラ 230 は、PP のそれぞれに、プログラムを実行させる。つまり、スケジューラ 230 は、PP にアサインされた全ての LP の動作状態を実行状態に設定する。

10

【0059】

スリープモード時には、スケジューラ 230 は、複数の PP の一部のみにホスト処理プログラムを実行させる。また、スケジューラ 230 は、PP にメディア処理プログラムを実行させない。つまり、スケジューラ 230 は、複数の PP のうち少なくとも 1 つの PP がプログラムを実行しないようにスケジューリングを行う。

【0060】

スケジューラ 230 は、PP にアサインされているホスト処理プログラムを実行する LP の動作状態を実行状態に設定し、PP にアサインされているメディア処理プログラムに対応する LP の動作状態を `suspended ready` 状態に設定する。ここで、スケジューラ 230 は、メディア処理プログラムを実行する LP のうちで `suspended ready` 状態又は `ready` 状態であるものが存在し、かつ、LP がアサインされていない PP が存在する場合でも、メディア処理プログラムを実行する LP を PP にアサインしない。

20

【0061】

また、スケジューラ 230 は、動作モードが通常モードからスリープモードに変更された際に、クォンタム時間に依存しないタイミングで、スケジューリングを行う。言い換えると、スケジューラ 230 は、スリープモードレジスタ 213 の設定が変更されたタイミングに依存するタイミングで、スケジューリングを行う。

【0062】

コンテキストメモリ 250 は、LP で用いられるコンテキストを記憶する。コンテキストとは、プログラムを実行するために必要な制御情報及びデータ情報等である。

30

【0063】

コンテキスト転送制御部 240 は、実行が完了したプログラムのコンテキストをコンテキストメモリ 250 に書き込む。また、コンテキスト転送制御部 240 は、実行されるプログラムのコンテキストをコンテキストメモリ 250 から読み出し、読み出したコンテキストを当該プログラムに対応する LP がアサインされた PP に転送する。

【0064】

クロック停止部 120 は、スリープモードにおいて、複数の PP のうち LP がアサインされていない PP へのクロックの供給を停止する。

40

【0065】

図 5 は、LP の状態遷移を示す図である。当該状態遷移はスケジューラ 230 により行われる。

【0066】

次に、本発明の実施の形態 1 に係る仮想マルチプロセッサ 100 の動作を説明する。

【0067】

まず、通常モード時の仮想マルチプロセッサ 100 の動作を説明する。

【0068】

図 6 A は、通常モード時の各 PP にアサインされる LP の一例を示す図である。図 6 B は、LP の状態の一例を示す図である。

50

【0069】

時刻T1より前において、PP0、PP1及びPP2にはそれぞれLP0、LP1及びLP2がアサインされている。なお、LP0～LP2で実行されるプログラムは、ホスト処理プログラムであっても、メディア処理プログラムであってもよい。

【0070】

時刻T1においてLP0で実行されていたプログラムの実行が終了する。スケジューラ230は、時刻T1より所定の時間前の時刻T0において、スケジューリングを開始する。

【0071】

この実施例ではスケジューリングの優先度は、ラウンドロビン方式を用いて決定されるものとする。つまり、IDが小さいLPほど優先度が高く、かつ実行が完了したプログラムに対応するLPの優先度は最低に設定される。

【0072】

したがって、LP0の割り当て時間が終了したのち、ready状態のLPの中で次に最も優先度が高いLPはLP3である。よって、スケジューラ230は、LP0の次にPP0にアサインするLPをLP3と決定する。また、スケジューラ230は、次にPP0にアサインするLPがLP3であることをコンテキスト転送制御部240に通知する。

【0073】

コンテキスト転送制御部240は、コンテキストメモリ250からLP3のコンテキストを読み出し、読み出したコンテキストをPP0に転送する。

【0074】

ここで、PP0～PP2は、それぞれ2つのコンテキストを保持可能である。これにより、コンテキスト切り替えの際のオーバーヘッドが低減される。

【0075】

また、時刻T1で、VMPC200は、PP0にLP0の実行時間が割り当て時間に等しくなったことを通知する。PP0は、コンテキスト転送制御部240により転送されたLP3のコンテキストを用いてLP3に対応するプログラムの実行を開始する。

【0076】

次に、通常モードからスリープモードに遷移するときの仮想マルチプロセッサ100の動作を説明する。

【0077】

図7Aは、通常モード及びスリープモードの遷移時において、各PPにアサインされるLPを示す図である。図7Bは、スリープモードに遷移するときの各LPの状態を示す図である。ここでは、LP0上で実行されるプログラムはホスト処理プログラムであり、LP1及びLP2上で実行されるプログラムはメディア処理プログラムであるとする。

【0078】

ホスト処理プログラム上でスリープモードに遷移することが決定されたとき、LP0で実行中のプログラムの特権レベルは、ユーザレベルからシステムレベルに遷移する。

【0079】

PP0上のシステムレベルのプログラムにより、時刻T2において、スリープモードレジスタ213の第0ビットに1が書き込まれる。スケジューラ230は、スリープモードレジスタ213の設定が変更されたことを検出し、スケジューリングを開始する。また、このとき、スケジューラ230はスリープモードレジスタ213にアクセスしたPPを検出し（この例ではPP0）、PP0以外のPPにアサインされているLPすなわちLP1とLP2とをsuspended ready状態にする。また、スケジューラ230は、PP0にLP0をアサインし、実行を続けさせる。

【0080】

スケジューリング開始時刻に到達したとき（時刻T3及び時刻T4）、スケジューラ230は、PP0にLP0を再度アサインし、PP1及びPP2にはready状態及びsuspended ready状態のLPが存在するにもかかわらず、アサインしない。

10

20

30

40

50

【0081】

時刻T5で、PP0によりスリープモードレジスタ213の第0ビットに0が書き込まれ、スケジューラ230が起動する。スケジューラ230は、PP1及びPP2に、直前にアサインされていたLP1とLP2とをそれぞれアサインし、PP0～PP2にプログラムの実行を開始させる。つまり、スケジューラ230は、スリープモードレジスタ213の設定が再度通常モードに変更された後に、スリープモードの移行時に一時中断したLPを優先してPPに実行させる。

【0082】

一方、時刻T2から時刻T5において、クロック停止部120は、PP1及びPP2へのクロック供給を停止する。

10

【0083】

図8は、通常モード時及びスリープモード時のスケジューリング動作の流れを示すフローチャートである。

【0084】

まず、スケジューラ230が起動される(S101)。具体的には、PPで実行されているプログラムの実行時間がクォンタム時間に一致する時刻より所定の時間前の時刻、及びスリープモードレジスタ213に保持される値が変更された際に、スケジューラ230が起動される。

【0085】

次に、スケジューラ230は、スリープモードレジスタ213に保持される値が1であるか0であるかを判定する(S102)。スリープモードレジスタ213に保持される値が0の場合(S102でNo)、つまり通常モード時には、ready状態、及びsuspended ready状態のLPのうち優先度の高いLPを、割り当て時間が終了するプログラムを実行していたPPにアサインする(S103)。

20

【0086】

次に、スケジューラ230が停止される(S104)。

【0087】

一方、スリープモードレジスタ213に保持される値が1の場合(S102でYes)、つまり通常モードからスリープモードに変更された際には、スケジューラ230は、当該変更を行ったプログラム以外のプログラムの実行を一時中断する。つまり、スケジューラ230は、PPに実行されているプログラムのうちメディア処理プログラムの実行を一時中断する。

30

【0088】

具体的には、まず、スケジューラ230は、スリープモードレジスタ213に1を書き込んだLPのID及びPPのIDを取得し、保持する(S105)。つまり、スケジューラ230は、ホスト処理プログラムを実行しているLPのID及び当該LPがアサインされているPPのIDを取得し、保持する。

【0089】

次に、スケジューラ230は、ステップS105で取得したIDのLP以外のrunning状態のLPの動作状態をsuspended ready状態にする(S106)。このとき、ステップS105で取得したIDのLP以外のrunning状態のLPが仮想マルチプロセッサ100の外部にアクセス中の場合は、スケジューラ230は、アクセスが完了するまで待ち、アクセス完了後に、当該LPの動作状態をsuspended ready状態に変更する。

40

【0090】

また、スケジューラ230は、ステップS105で取得したPPのIDのPPに、同じく取得したLPのIDのLPをアサインし、実行を続けさせる(S107)。

【0091】

次に、スケジューラ230が停止される(S104)。

【0092】

50

以上より、本発明の実施の形態１に係る仮想マルチプロセッサ１００では、スリープモード時には、スケジューラ２３０は、複数のＰＰのうち一部のＰＰにホスト処理プログラムを実行させ、他のＰＰにはメディア処理プログラムの実行を一時中断させる。これにより、プログラムを実行していないＰＰに対してクロックの供給を停止することで、消費電力を削減できる。

【００９３】

また、仮想マルチプロセッサ１００では、通常モードからスリープモードへの移行時には、クオンタム時間に依存しないタイミングで、スリープモード時のスケジューリングを行う。これにより、仮想マルチプロセッサ１００は、スリープモードへの移行時に直ちにスケジューリングを行えるので、スリープモードに移行する際のオーバーヘッドを削減できる。また、仮想マルチプロセッサ１００は、物理プロセッサ１１０へのクロックの供給又は停止の制御を低粒度で行えるので、極めて高い消費電力削減効果を実現できる。

10

【００９４】

以上、本発明の実施の形態１に係る仮想マルチプロセッサ１００について説明したが、本発明は、この実施の形態に限定されるものではない。

【００９５】

例えば、上記説明において、スリープモード時には、ホスト処理プログラムを実行する１つのＬＰのみが１つのＰＰにアサインされ、実行される例を示したが、ホスト処理プログラムを実行する２以上のＬＰが切り替えられながら１つのＰＰにアサインされてもよい。

20

【００９６】

この場合、VMPC２００は、図１に示す構成に加えて、さらに、ＬＰ制御レジスタ２６０を備えればよい。図９は、ＬＰ制御レジスタ２６０の構成を示す図である。

【００９７】

ＬＰ制御レジスタ２６０は、スリープモード時に実行される複数のホスト処理プログラムが設定されるレジスタである。ＬＰ制御レジスタ２６０は、３２個のレジスタLPnCTL R (n=0-31)を含む。各レジスタLPnCTL Rの第０ビットには、グループIDが設定される。グループIDは、当該ＬＰをスリープモード時に実行するか否かを示す情報である。具体的には、ホスト処理プログラムを実行するＬＰに対応するレジスタLPnCTL RにはグループID=0が設定される。一方、メディア処理プログラムを行う

30

【００９８】

スリープモード時には、スケジューラ２３０は、グループID=0が設定された複数のＬＰに対応するプログラムを、１つのＰＰに切り替えながら実行させる。また、スケジューラ２３０は、グループID=1が設定されたＬＰをsuspended ready状態にし、当該ＬＰに対するプログラムをＰＰに実行させない。

【００９９】

また、複数のＰＰがホスト処理プログラムを並列に実行しているときにスリープモードレジスタ２１３に１が書き込まれた場合、スケジューラ２３０は、当該ホスト処理プログラムの処理を完了させたのちに、グループID=0が設定された複数のＬＰに対応するプログラムを、１つのＰＰに切り替えながら実行させる。

40

【０１００】

これにより、ホスト処理プログラムが複数のＬＰ上で実行される場合においても同様の効果を得られる。

【０１０１】

また、上記説明では、スリープモード時には、１つのＰＰのみがプログラムを実行するとしたが、２以上のＰＰがプログラムを実行してもよい。つまり、仮想マルチプロセッサ１００が備える複数のＰＰのうち一部のみのみがプログラムを実行すればよい。

【０１０２】

この場合、VMPC２００は、さらに、ＰＰ制御レジスタ２７０を備えればよい。図１

50

0 は、P P 制御レジスタ 2 7 0 の構成を示す図である。

【0 1 0 3】

P P 制御レジスタ 2 7 0 は、スリープモード時に動作させる P P の数が設定されるレジスタである。

【0 1 0 4】

スリープモード時には、スケジューラ 2 3 0 は、P P 制御レジスタ 2 7 0 に設定された P P の数分の P P にそれぞれ L P をマッピングする。

【0 1 0 5】

これにより、スリープモード時に、複数のホスト処理プログラムを並列に複数の P P 上で実行できる。

【0 1 0 6】

また、上記説明において、スリープモード時に実行するプログラムは携帯電話用アプリケーションプログラムに含まれるホスト処理プログラムとしたが、本発明はこれに限定されるものでない。

【0 1 0 7】

また、上記説明において、仮想マルチプロセッサ 1 0 0 は、3 つの P P を備えるとしたが、1 以上の P P を備えればよい。また、仮想マルチプロセッサ 1 0 0 が 1 つの P P のみを備える場合には、スリープモードにおいては、スケジューラ 2 3 0 は、当該 1 つの P P の L P を `suspended ready` 状態にし、プログラムを実行させない。また、クロック停止部 1 2 0 は、当該 1 つの P P へのクロックの供給を停止する。

【0 1 0 8】

(実施の形態 2)

本発明の実施の形態 2 では、上述した実施の形態 1 に係る仮想マルチプロセッサ 1 0 0 を備える携帯電話機器について説明する。

【0 1 0 9】

まず、本発明の実施の形態 2 に係る携帯電話機器 5 0 0 の構成を説明する。

【0 1 1 0】

図 1 1 は、本発明の実施の形態 2 に係る携帯電話機器 5 0 0 の構成を示す機能ブロック図である。携帯電話機器 5 0 0 は、音楽再生機能付き携帯電話機器である。

【0 1 1 1】

図 1 1 に示す携帯電話機器 5 0 0 は、システム L S I 5 1 0 と、アンテナ 5 2 1 と、高周波信号送受信部 5 2 2 と、外部メモリ 5 2 3 と、入出力部 5 2 4 と、S D R A M 5 2 5 とを備える。

【0 1 1 2】

高周波信号送受信部 5 2 2 は、アンテナ 5 2 1 を介して、通話データ等を送受信する。

【0 1 1 3】

外部メモリ 5 2 3 は、音楽データを蓄積する。

【0 1 1 4】

入出力部 5 2 4 は、スピーカを内蔵した受話器である。

【0 1 1 5】

S D R A M 5 2 5 は、再生を行う音楽データ及び再生処理後の音楽データ等を一時格納する。

【0 1 1 6】

システム L S I 5 1 0 は、例えば、1 チップの半導体集積回路である。システム L S I 5 1 0 は、仮想マルチプロセッサ 1 0 0 と、バス制御部 5 3 0 と、メモリ I / F 5 1 1 と、I / O 制御部 5 1 2 とを備える。なお、仮想マルチプロセッサ 1 0 0 のみを 1 チップの半導体集積回路として形成してもよいし、仮想マルチプロセッサ 1 0 0 と、仮想マルチプロセッサ 1 0 0、バス制御部 5 3 0、及びメモリ I / F 5 1 1 のうちいずれか 1 以上とを 1 チップの半導体集積回路として形成してもよい。

【0 1 1 7】

仮想マルチプロセッサ１００は、実施の形態１で述べた仮想マルチプロセッサ１００である。

【０１１８】

メモリＩ／Ｆ５１１は、ＳＤＲＡＭ５２５へのデータ書き込み及び読み出しを行う。

【０１１９】

Ｉ／Ｏ制御部５１２は、高周波信号送受信部５２２、外部メモリ５２３及び入出力部５２４とのデータの入出力を制御する。

【０１２０】

バス制御部５３０は、仮想マルチプロセッサ１００、メモリＩ／Ｆ５１１及びＩ／Ｏ制御部５１２の間を接続するバスを制御する。

【０１２１】

次に、携帯電話機器５００において、外部メモリ５２３に蓄積される音楽データを再生する動作を説明する。

【０１２２】

まず、Ｉ／Ｏ制御部５１２は、外部メモリ５２３に蓄積される音楽データを読み出す。Ｉ／Ｏ制御部５１２は、メモリＩ／Ｆ５１１を介して、読み出した音楽データをＳＤＲＡＭ５２５に格納する。

【０１２３】

仮想マルチプロセッサ１００は、メモリＩ／Ｆ５１１及びバス制御部５３０を介して、ＳＤＲＡＭ５２５から音楽データを読み出す。仮想マルチプロセッサ１００は、読み出した音楽データに再生処理を行う。仮想マルチプロセッサ１００は、再生処理後のデータをＳＤＲＡＭ５２５に格納する。

【０１２４】

Ｉ／Ｏ制御部５１２は、ＳＤＲＡＭ５２５に格納された再生処理後のデータを、メモリＩ／Ｆ５１１を介して読み出す。Ｉ／Ｏ制御部５１２は、読み出した再生処理後のデータを入出力部５２４に送る。

【０１２５】

入出力部５２４は、Ｉ／Ｏ制御部５１２から送られた再生処理後のデータをスピーカから出音する。

【０１２６】

ここで、音楽データの再生処理を行っている間、仮想マルチプロセッサ１００は常に全てのＰＰを用いて処理を実行しているわけではなく、一部のプログラム（実施の形態１で説明したホスト処理プログラム）以外のプログラムを実行する必要がない期間がある。この期間、仮想マルチプロセッサ１００は稼動するＰＰを１個にし、当該ＰＰ上でホスト処理プログラムのみを実行させる。これにより、携帯電話機器５００の消費電力を削減できる。

【産業上の利用可能性】

【０１２７】

本発明は、仮想マルチプロセッサ、システムＬＳＩ、携帯電話機器、及び仮想マルチプロセッサの制御方法に適用でき、特に仮想マルチプロセッサを備える携帯電話機器等の移動体通信機器に適用できる。

【図面の簡単な説明】

【０１２８】

【図１】本発明の実施の形態１に係る仮想マルチプロセッサの構成を示すブロック図である。

【図２】本発明の実施の形態１に係る論理プロセッサ状態レジスタの構成を示す図である。

【図３】本発明の実施の形態１に係るクォンタムレジスタの構成を示す図である。

【図４】本発明の実施の形態１に係るスリープモードレジスタの構成を示す図である。

【図５】本発明の実施の形態１に係る論理プロセッサの状態遷移を示す図である。

10

20

30

40

50

【図 6 A】通常モード時に、本発明の実施の形態 1 に係る物理プロセッサにアサインされる論理プロセッサの一例を示す図である。

【図 6 B】通常モード時に、本発明の実施の形態 1 に係る論理プロセッサの状態の一例を示す図である。

【図 7 A】スリープモード時に、本発明の実施の形態 1 に係る物理プロセッサにアサインされる論理プロセッサの一例を示す図である。

【図 7 B】スリープモード時に、本発明の実施の形態 1 に係る論理プロセッサの状態の一例を示す図である。

【図 8】本発明の実施の形態 1 に係る仮想マルチプロセッサの動作の流れを示すフローチャートである。

10

【図 9】本発明の実施の形態 1 に係る L P 制御レジスタの構成を示す図である。

【図 10】本発明の実施の形態 1 に係る P P 制御レジスタの構成を示す図である。

【図 11】本発明の実施の形態 2 に係る携帯電話機器の構成を示すブロック図である。

【符号の説明】

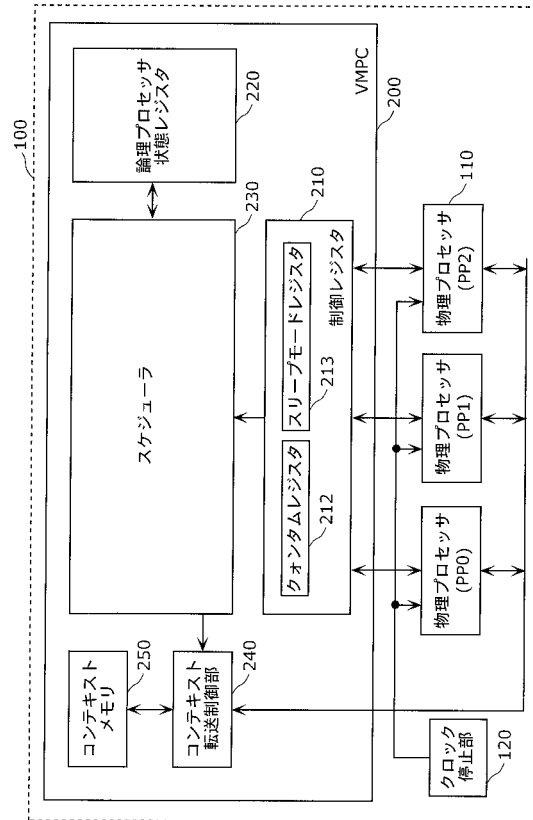
【0129】

- 100 仮想マルチプロセッサ
- 110 物理プロセッサ
- 120 クロック停止部
- 200 VMPC
- 210 制御レジスタ
- 212 クォンタムレジスタ
- 213 スリープモードレジスタ
- 220 論理プロセッサ状態レジスタ
- 230 スケジューラ
- 240 コンテキスト転送制御部
- 250 コンテキストメモリ
- 260 LP 制御レジスタ
- 270 PP 制御レジスタ
- 500 携帯電話機器
- 511 メモリ I/F
- 512 I/O 制御部
- 521 アンテナ
- 522 高周波信号送受信部
- 523 外部メモリ
- 524 入出力部
- 525 SDRAM
- 530 バス制御部

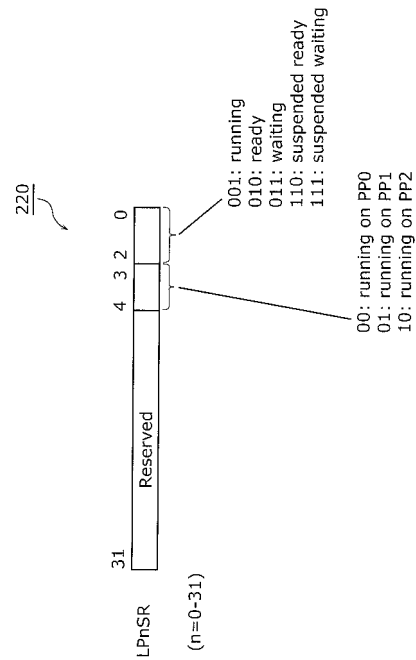
20

30

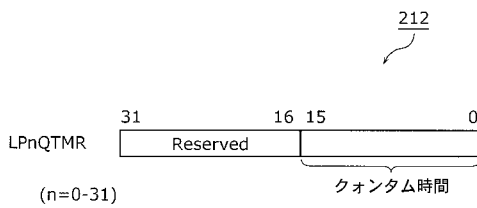
【図 1】



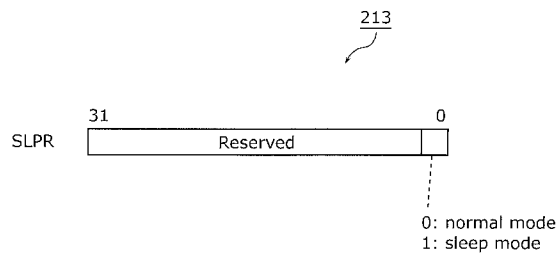
【図 2】



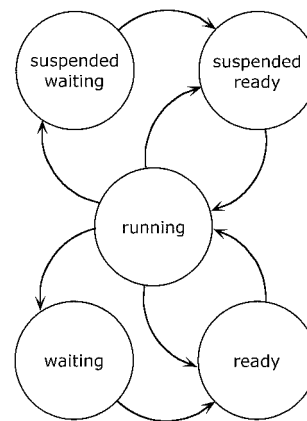
【図 3】



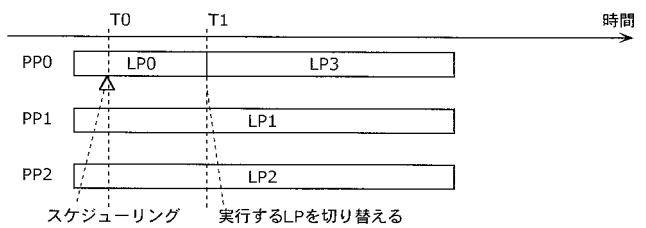
【図 4】



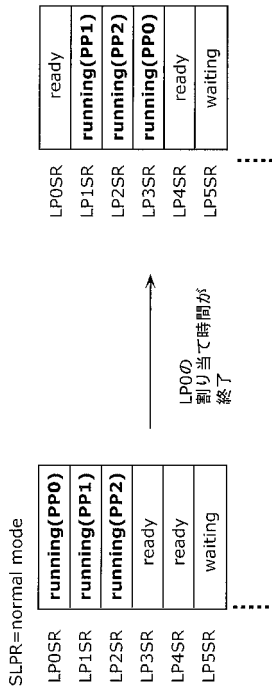
【図 5】



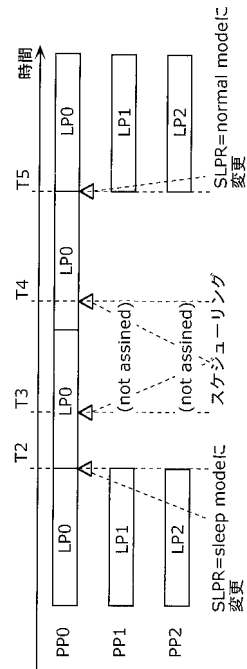
【図 6 A】



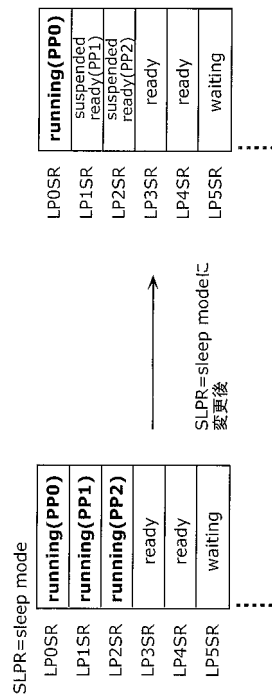
【図 6 B】



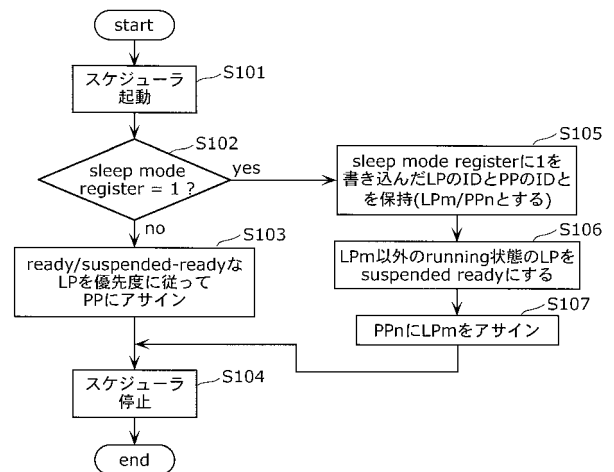
【図 7 A】



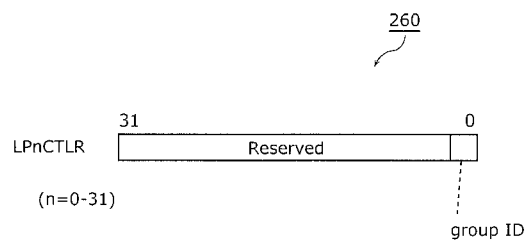
【図 7 B】



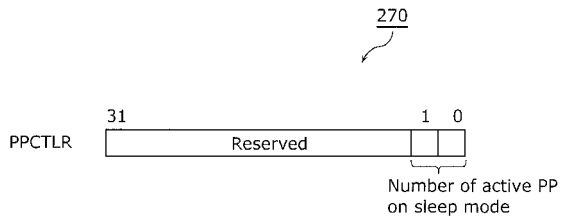
【図 8】



【図 9】



【図 10】



【図 11】

