

(11) 特許出願公開番号

特開2013-191202

(P2013-191202A)

(43) 公開日 平成25年9月26日(2013.9.26)

(51) Int.Cl.

G06F 15/167 (2006.01)

G O 6 F 9/52 (2006.01)

F 1

G06F 15/167 610B

G O 6 F 9/46 4 7 2 Z

テーマコード (参考)

5 B 0 4 5

審査請求 未請求 請求項の数 3 O L (全 8 頁)

(21) 出願番号 特願2013-23797 (P2013-23797)

(22) 出願日 平成25年2月8日 (2013.2.8)

(31) 優先權主張番号 特願2012-29674 (P2012-29674)

(32) 優先日 平成24年2月14日 (2012. 2. 14)

(33) 優先権主張国 日本国 (JP)

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(74) 代理人 100089118

弁理士 酒井 宏明

(72) 発明者 今田 萌

東京都中央区晴海1-8-10晴海アイランドトリトンスクエアオフィスタワーX
リコー1Tソリューションズ株式会社内

(72) 癸明者 大泉 敬弘

東京都中央区晴海1-8-10晴海アイランドトリトンスクエアオフィスタワーX
リコー1Tソリューションズ株式会社内

最終頁に続く

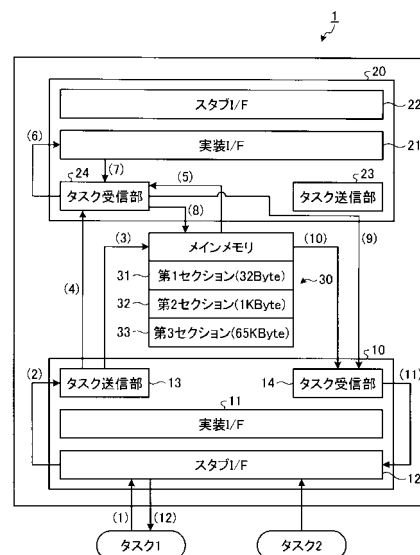
(54) 【発明の名称】 マルチコアプロセッサ

(57) 【要約】

【課題】複数のタスクを処理するマルチコアプロセスシステムの処理速度を向上させる。

【解決手段】マルチコアプロセッサ１は、複数のコア１０、２０と、コア１０、２０それぞれが共有して利用し、書き込み可能なデータ容量の範囲毎に区分された複数のセクション３１～３３に分割されたメインメモリ３０と、を有する。コア１０、２０に要求されるタスクは、当該タスクのデータ容量によって、書き込まれるセクション３１～３３が予め設定されている。

【選択図】 図 1



【特許請求の範囲】**【請求項 1】**

複数のコアと、

前記コアそれぞれが共有して利用し、書き込み可能なデータ容量が予め定められた複数の記憶領域に区分された共有メモリと、

前記コアに対して要求されるタスクを受け付ける受付部と、

受け付けた前記タスクを、当該タスクのデータ容量に応じて予め設定された前記記憶領域に対して書き込む書き込み部と、

を備えることを特徴とするマルチコアプロセッサ。

【請求項 2】

10

前記共有メモリは、同一の前記データ容量の範囲に対応する前記記憶領域をそれぞれ少なくとも 2 以上有している

ことを特徴とする請求項 1 に記載のマルチコアプロセッサ。

【請求項 3】

前記書き込み部は、前記記憶領域に対して前記タスクを書き込む際に、他のタスクによる書き込みを禁止する排他処理を行うとともに、前記タスクの共有メモリにおける書き込み位置を交換先の他の前記コアへと通知し、

前記排他処理は、前記他のコアによる当該タスクの処理が完了した結果の戻り値が前記記憶領域から読み出された際に、解除される

ことを特徴とする請求項 1 に記載のマルチコアプロセッサ。

20

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、複数のコアを備えたマルチコアプロセッサに関する。

【背景技術】**【0002】**

従来、複数のコアがメインメモリを共有する密結合型のマルチコアプロセッサシステムが知られている。このようなマルチコアプロセッサシステムの一例としては特許文献 1 に記載されているように、コア毎のメッセージ交換バッファをメインメモリに設け、この交換バッファを介してデータ交換が行われる構成が採用されている。

30

【0003】

具体的には、まず送信側のコアがデータを共有メモリ上のメッセージ交換バッファにデータをセットしたのちに、受信側のコアに対して割り込み要求を出す。受信側のコアは、メッセージ交換バッファからデータを取り込み、受信バッファにデータをセットする。受信側のコアは、受信データによる要求処理完了後、処理完了のメッセージをメッセージ交換バッファにセットする。受信側のコアは、送信側のコアに対して割り込み要求を出し、送信側のコアはメッセージ交換バッファから処理完了のメッセージを受理する。

【発明の概要】**【発明が解決しようとする課題】****【0004】**

40

しかしながら、上記のような方式におけるコア間通信においては、あるタスクにおいてコア間の通信が発生しているときには、他のタスクに対してメモリへの書き込みが排他状態となっている。このため、処理待ちが発生してしまい、処理速度の低下を招いてしまう可能性があった。

【0005】

本発明は、上記に鑑みてなされたものであって、複数のタスクを処理するマルチコアプロセッサの処理速度を向上させることを目的とする。

【課題を解決するための手段】**【0006】**

上述した課題を解決し、目的を達成するために、本発明は、複数のコアと、前記コアそ

50

れぞれが共有して利用し、書き込み可能なデータ容量が予め定められた複数の記憶領域に区分された共有メモリと、を有するマルチコアプロセッサであって、前記コアに要求されるタスクは、当該タスクのデータ容量によって、書き込まれる前記記憶領域が予め設定されていることを特徴とする。

【発明の効果】

【0007】

本発明によれば、複数のタスクを処理するマルチコアプロセッサの処理速度を向上させることができるという効果を奏する。

【図面の簡単な説明】

【0008】

10

【図1】図1は、マルチコアプロセッサの全体構成を示す概念図である。

【図2】図2は、マルチコアプロセッサにおけるメインメモリへの書き込み処理の概要を示す図である。

【図3】図3は、マルチコアプロセッサによる、コア間におけるメインメモリへの書き込みが可能な際の処理の流れを示すシーケンス図である。

【図4】図4は、マルチコアプロセッサによる、コア間におけるメインメモリへの書き込みができない場合の処理の流れを示すシーケンス図である。

【発明を実施するための形態】

【0009】

20

以下に添付図面を参照して、マルチコアプロセッサの実施の形態を詳細に説明する。図1は、実施の形態にかかるマルチコアプロセッサの構成を示すブロック図である。マルチコアとは、1つのプロセッサパッケージ内に複数のプロセッサコアを封入したものであり、本実施形態においては2つのコアを実装した実施形態を示すが、コアの数が増えた場合であっても本発明は適用可能である。

【0010】

30

図1に示されるマルチコアプロセッサ1は、第1コア10、第2コア20、及びメインメモリ30を備えている。第1コア10には、実装I/F11、スタブI/F12、タスク送信部13、及びタスク受信部14が設けられている。同じく、第2コア20には、実装I/F21、スタブI/F22、タスク送信部23、及びタスク受信部24が設けられている。タスクとは、様々なプログラムやライブラリにより実行を要求される処理命令である。

【0011】

40

実装I/F11、22は、受理したタスクをプロセッサにおいて実行するための処理命令として受け付けるインタフェースである。スタブI/F12、22は、異なるコア間では直接実装I/F11、12をコールすることができないことから、仮想的に他のコアのシステムをコールすることができるように論理的に設定したスタブとして機能する。第1コア10のスタブI/F12は、第2コア20の実装I/F21と論理的に同じインタフェースとして設定されており、一方、第2コア20のスタブI/F22は、第1コア10の実装I/F11と論理的に同じインタフェースとして設定されている。

【0012】

50

タスク送信部13、23は、コアが受信したタスクをメインメモリ30へと書き込むとともに、書き込みの際に他のコアのタスク受信部14、24へと書き込みを通知し、書き込み部に相当する。なお、タスク送信部13、23は、メインメモリ30への書き込みの際には、メインメモリ30の記憶領域への他の処理による書き込みを禁止する排他制御を行う。タスク受信部14、24は、タスク送信部13、23から受理した書き込み通知を契機に、メインメモリ30の指定の場所からデータの読み込み処理を行い、受付部に相当する。このように、メインメモリ30と介して、第1コア10、及び第2コア20の間でのデータの交換が行われる。

【0013】

また、メインメモリ30（共有メモリ）は、3つのセクション（記憶領域）を備えてい

50

る。第1セクション31は、データ容量が32バイト以下のデータの書き込み、及び読み込みに利用されるセクションである。第2セクション32は、32バイトより大きく、1キロバイト以下のデータの書き込み、及び読み込みに利用されるセクションである。第3セクション33は、1キロバイトより大きく、65キロバイト以下のデータの書き込み、及び読み込みに利用されるセクションである。なお、本実施形態にあっては、全てのセクションのサイズが異なる場合を表示したが、同じサイズに対応したセクションが複数も受けられるようにしてもよい。また、メインメモリ30は、図示しないセクション31～33以外の領域に、各セクションの位置の範囲を示すアドレス情報や、各セクション31～33が使用中であるか否かを示すフラグ情報などを記憶する領域を有している。

【0014】

10

図2は、どのセクション31～33に対してタスクが書き込まれるかについての概要を示した図である。図2に示されるように、スタブI/F12、22には、それぞれプロトコル(プログラムやライブラリ)毎に対応するインタフェースがそれぞれ設けられている。例えば、プロトコル1から要求されたタスクについては、スタブI/F Aがコールされて、タスク送信部13へと引き渡される。コールされるインタフェースは、タスクのデータ容量に応じて設定されている。したがって、プロトコルが要求するタスクごとに、そのデータ容量に応じて予め書き込みが行われるセクションが設定されている。

【0015】

次に、図1を参照して、コア間のデータの交換の処理の流れについて説明する。説明においては、第1コア10から第2コア20へとデータが渡される場合を想定している。図1に示されるように、(1)第1コア10に対して要求されたタスクは、第1コア10において設けられた第2コア20と論理的に接続されたスタブI/F12をコールする。この際、コールするスタブI/F12は、タスクに必要なメモリのサイズによって決定される。(2)続いて、スタブI/F12は、要求タスクをタスク送信部13へと転送して、処理要求を行う。(3)処理要求を受け付けたタスク送信部13は、メインメモリ30へとタスクの書き込みを行う。この際、タスクに必要なデータ容量に応じて予め設定されたセクション31～33に対して、書き込みが行われる。本実施形態では、コールされたスタブI/F12によって、書き込まれるメインメモリ30のセクションが予め対応付けられて決定されている。したがって、タスク1がスタブI/F12を選択した時点で書き込むセクションは決定されている。書き込むセクションの判断基準は、上述した各セクション31～33毎に定められた対応するデータ容量に基づく。なお、タスク送信部13は、データを書き込んだセクション31～33に対し、後述するタスク受信部14によるデータの読み込みが行われるまでの間、他のタスクの書き込みを禁止する排他制御を行う。また、この段階でタスク送信部13は、解放されて他のタスクを受け付けて処理可能となる。

20

30

【0016】

(4)続いて、タスク送信部13は、第2コア20のタスク受信部24に対して、メインメモリ30の書き込みを行ったセクション31～33を指定して、通知を行う。(5)通知を受け取ったタスク受信部24は、メインメモリ30の指定のセクション31～33からタスクが書き込んだ情報の読み込みを行う。(6)タスク受信部24は、実装I/F21をコールして、読み込んだタスクが書き込んだ情報を引き渡す。(7)タスクが書き込んだ情報を受け取った実装I/F21は、タスクが書き込んだ情報に基づいた処理を第2コア20にて実行し、その実行結果をタスク受信部24へと返信する。(8)タスク受信部24は、受け取った実行結果をメインメモリ30の該当するセクション31～33へと書き込む。(9)タスク受信部24は、実行結果を書き込んだことをコア10のタスク受信部14へと通知する。(10)第1コア10のタスク受信部14は、メインメモリ30の指定のセクション31～33から第2コア20におけるタスクの実行結果を読み込む。なお、この時点でタスク1によるメインメモリ30の排他制御は終了する。具体的には、上述する該当するセクションの使用状況が否かを示すフラグ情報が更新される。(11)タスク受信部14は、読み込んだタスクの実行結果をコールされたスタブI/F12へと

40

50

送信する。(12)そして、最後にスタブI/F12は、元のタスク1へと実行結果を送信し、処理を終了する。なお、第2コア20においてタスクを要求された場合においても同様の処理の流れとなる。

【0017】

続いて、以上で示したタスクの処理の流れを図3、図4のシーケンス図を用いて説明する。図3は、メインメモリ30に書き込みが成功した場合、図4はメインメモリ30への書き込みが出来なかった場合を示している。図3に示されるように、まずタスク1はスタブI/F12に対して、ファンクションコールを実行する(ステップS101)。次いで、スタブI/F12は、タスク送信部13に対して、ファンクションIDや引数情報、メインメモリ30で書き込みに必要なセクションサイズなどの情報を含むファンクションコール要求を行う(ステップS102)。タスク送信部13は、メインメモリ30に対して、スタブI/F12から要求されたセクションサイズに相当するメインメモリ30のセクション31~33を確保する(ステップS103)。また、タスク送信部13は、確保したメインメモリ30のセクション31~33に相当する使用中を示すフラグ情報を「使用中」に相当する値に更新する(ステップS104)。

10

【0018】

次いで、タスク送信部13は、メインメモリ30の該当するセクション31~33に、ファンクションIDと、引数を書き込む(ステップS105)。これらの処理をして、コア20からの返信があるまでスタブI/F12は待機中となる(ステップS106)。次いで、タスク送信部13は、コア20のタスク受信部24に対して、書き込んだメインメモリ30のセクション31~33の書き込み位置を通知する(ステップS107)。タスク受信部24は、実装I/F21に対してファンクションコールを送信し、実装I/F21を通じて処理が実行され、処理の結果を受け取る(ステップS108)。

20

【0019】

タスク受信部24は、タスクの処理結果として受け取ったファンクションIDと、戻り値をメインメモリ30に書き込む(ステップS109)。この際、書き込むセクション31~33も、ステップS103で確保したセクションと同様である。次いで、タスク受信部24は、コア10のタスク受信部14に対して戻り値を書き込んだメインメモリ30の位置情報を通知する(ステップS110)。タスク受信部14は、指定された位置情報のメインメモリ30からファンクションIDと戻り値を読み出す(ステップS111)。この際、タスク受信部14は、メインメモリ30の該当するセクション31~33のフラグ情報を「未使用」へと更新する(ステップS112)。タスク受信部14は、戻り値をスタブI/F12へと通知し(ステップS112)、戻り値はタスク1へと返却される(ステップS113)。

30

【0020】

次に、メインメモリ30への書き込みが出来なかった場合について図4を参照して説明する。ステップS103までは図3の場合と同様であり説明は省略する。図4に示されるように、ステップS103のメモリ領域の確保にかかる処理の結果として、タスク送信部13は、エラーを受け取る。タスク送信部13は、スタブI/F12に対してエラー戻り値を通知し(ステップS201)、タスク1にエラー戻り値が返却される(ステップS202)。

40

【0021】

以上で示した本実施の形態のマルチコアプロセッサ1においては、複数のコア間におけるデータの交換がメインメモリ30を介して行われる場合に、タスクのデータサイズに応じて利用するメインメモリ30のセクション31~33を変更する。このため、複数のタスクが並列して処理を要求される場合に、排他処理によってメインメモリ30への書き込みを行うことができずに待ち時間が発生してしまう頻度を極力抑えることができる。したがって、複数のタスクを処理するマルチコアプロセッサシステムの処理速度を向上させることができるようになる。

【0022】

50

このような並列するタスクの例としては、例えばＨＴＴＰのプロトコルで、プリンタなどの管理画面を取得する処理と、ＳＮＭＰのプロトコルでプリンタの状態を制御する処理などがある。このように同時に要求されることが多いタスクがあっても、メモリへのアクセスの待機待ちを回避して、処理の高速化を図ることができるようになる。

【００２３】

また、タスク送信部１３は、メインメモリ３０への書き込みが終わった時点で、新たなタスクを受理して処理することが可能となる。このため、処理の高速化を図ることができるようになる。

【００２４】

なお、本実施形態においては、メインメモリ３０のセクション数を３つとしたが、これは適宜変更することができる。また、同じデータサイズに対応するセクションを複数も受けてもよい。また、メインメモリ３０の各セクションごとに記憶可能なデータ量は本実施形態で示した例に限定されず、他の組み合わせに変更してもよい。

10

【符号の説明】

【００２５】

１ マルチコアプロセッサ

１０ 第１コア

１１ 実装Ｉ／Ｆ

１２ スタブＩ／Ｆ

１３ タスク送信部

１４ タスク受信部

20

２０ 第２コア

２１ 実装Ｉ／Ｆ

２２ スタブＩ／Ｆ

２３ タスク送信部

２４ タスク受信部

３０ メインメモリ

３１ 第１セクション

３２ 第２セクション

３３ 第３セクション

30

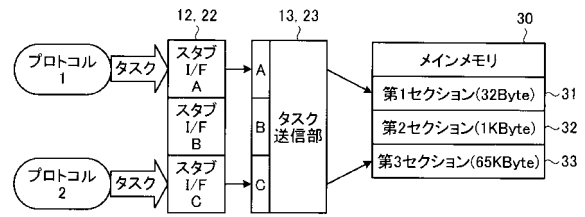
【先行技術文献】

【特許文献】

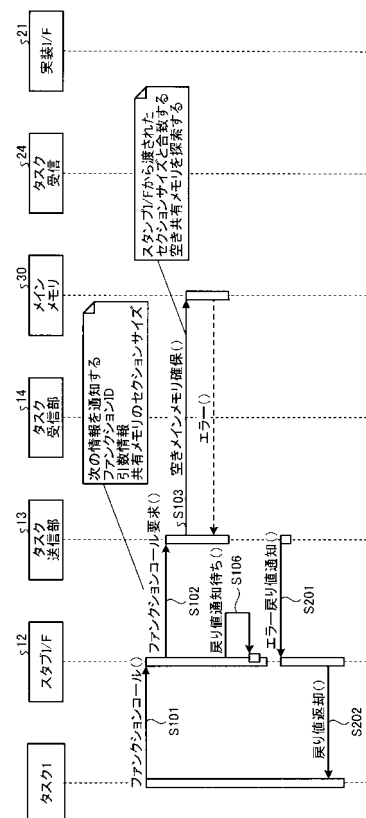
【００２６】

【特許文献１】特開昭５７－１６１９６２号公報

【 図 2 】



【 図 4 】



フロントページの続き

(72)発明者 山崎 真吾

東京都中央区晴海 1 - 8 - 10 晴海アイランドトリトンスクエアオフィスタワーX リコーＩＴソ
リューションズ株式会社内

Fターム(参考) 5B045 DD02 EE03 EE18 EE25 KK08