

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4756599号
(P4756599)

(45) 発行日 平成23年8月24日(2011.8.24)

(24) 登録日 平成23年6月10日(2011.6.10)

(51) Int.Cl.

G06F 9/48 (2006.01)

F 1

G06F 9/46 455B

請求項の数 9 (全 17 頁)

(21) 出願番号 特願2006-183973 (P2006-183973)
 (22) 出願日 平成18年7月4日 (2006.7.4)
 (65) 公開番号 特開2008-15638 (P2008-15638A)
 (43) 公開日 平成20年1月24日 (2008.1.24)
 審査請求日 平成21年1月21日 (2009.1.21)

(73) 特許権者 302062931
 ルネサスエレクトロニクス株式会社
 神奈川県川崎市中原区下沼部 1753番地
 (74) 代理人 100089071
 弁理士 玉村 静世
 (72) 発明者 森島 憲太
 東京都国分寺市東恋ヶ窪一丁目280番地
 株式会社日立製作所 中央研究所内

審査官 稲垣 良一

最終頁に続く

(54) 【発明の名称】データ処理装置

(57) 【特許請求の範囲】

【請求項 1】

第1CPUコアと、第2CPUコアとを具備して、
 前記第1CPUコアは第1レジスタファイルと第1命令デコーダ・演算器とを含み、前記第2CPUコアは第2レジスタファイルと第2命令デコーダ・演算器とを含み、

前記第1CPUコアの前記第1レジスタファイルの内容と前記第2CPUコアは前記第2レジスタファイルの内容とを格納する共有メモリを更に具備して、

前記第1CPUコアは前記第2CPUコアの前記第2レジスタファイル内部のスタックポインタの値を前記第1CPUコアのスタックポインタの値として処理するための第1記憶回路を含み、前記第2CPUコアは前記第1CPUコアの前記第1レジスタファイル内部のスタックポインタの値を前記第2CPUコアのスタックポインタの値として処理するための第2記憶回路を含み、

前記第1CPUコアと前記第2CPUコアとの一方のCPUコアが処理を実行している間に、前記第1CPUコアと前記第2CPUコアとの他方のCPUコアに前記一方のCPUコアが実行している前記処理よりも優先度の低い処理の要求が発生することによって、優先度の低い前記処理は前記他方のCPUコアによって実行されることなく中断され、前記他方のCPUコアの前記レジスタファイルのプログラムカウンタの内容とステータスレジスタの内容とがスタック領域に退避される一方、前記他方のCPUコアのレジスタファイルのその他の内容が前記共有メモリに退避され、

前記一方のCPUコアが前記処理を完了すると、前記一方のCPUコアは前記共有メモリ

10

20

リに退避された前記他方の C P U コアの前記レジスタファイルの前記その他の内容を前記一方の C P U コアのレジスタファイルに格納して、前記一方の C P U コアは、前記他方の C P U コアの前記レジスタファイルのスタックポインタの内容を前記一方の C P U コアの記憶回路に転送して、前記他方の C P U コアによって中断された前記処理を前記一方の C P U コアが再開して実行するデータ処理装置。

【請求項 2】

前記第 1 C P U コアと前記第 2 C P U コアとに接続されたシステムコントローラを更に具備して、

前記システムコントローラは、前記第 1 の C P U コアの中断した処理の復帰を抑制する第 1 フラグ情報と、前記第 2 の C P U コアの中断した処理の復帰を抑制する第 2 フラグ情報とを格納して、

10

前記一方の C P U コアが前記処理を完了すると、前記一方の C P U コアは前記他方の C P U コアの中断した処理の復帰を抑制するようにフラグ情報を所定の状態にセットして、

前記一方のフラグ情報が前記所定の状態にセットされることに応答して、前記他方の C P U コアによって中断された前記処理を前記他方の C P U コアが再開して実行することが禁止され、前記他方の C P U コアによって中断された前記処理を前記一方の C P U コアが再開して実行することが許可される請求項 1 に記載のデータ処理装置。

【請求項 3】

前記システムコントローラは前記第 1 C P U コアの前記第 1 レジスタファイルの内容を前記共有メモリの第 1 領域に格納する一方、前記第 2 C P U コアの前記第 2 レジスタファイルの内容を前記共有メモリの前記第 1 領域と異なる第 2 領域に格納するアドレス管理ユニットを含む請求項 1 と請求項 2 のいずれかに記載のデータ処理装置。

20

【請求項 4】

前記データ処理装置は、前記第 1 C P U コアと前記第 2 C P U コアとに接続された内部メモリを更に具備する一方、メインメモリと接続可能であり、

前記内部メモリと前記メインメモリとの少なくともいずれか一方に、前記他方の C P U コアの前記レジスタファイルの前記プログラムカウンタの前記内容と前記ステータスレジスタの前記内容とが退避される前記スタック領域が形成される請求項 1 から請求項 3 のいずれかに記載のデータ処理装置。

【請求項 5】

30

第 1 C P U コアと、第 2 C P U コアとを具備して、

前記第 1 C P U コアは第 1 レジスタファイルと第 1 命令デコーダ・演算器とを含み、前記第 2 C P U コアは第 2 レジスタファイルと第 2 命令デコーダ・演算器とを含み、

前記第 1 C P U コアの前記第 1 レジスタファイルの内容と前記第 2 C P U コアは前記第 2 レジスタファイルの内容とを格納する共有メモリを更に具備して、

前記第 1 C P U コアは前記第 2 C P U コアの前記第 2 レジスタファイル内部のスタックポインタの値を格納するための第 1 記憶回路を含み、前記第 2 C P U コアは前記第 1 C P U コアの前記第 1 レジスタファイル内部のスタックポインタの値を格納するための第 2 記憶回路を含み、

前記第 1 C P U コアと前記第 2 C P U コアとの一方の C P U コアが処理を実行している間に、前記第 1 C P U コアと前記第 2 C P U コアとの他方の C P U コアに前記一方の C P U コアが実行している前記処理よりも優先度の低い処理の要求が発生することによって、優先度の低い前記処理は前記他方の C P U コアによって実行されることなく中断され、前記他方の C P U コアの前記レジスタファイルのプログラムカウンタの内容とステータスレジスタの内容とがスタック領域に退避される一方、前記他方の C P U コアのレジスタファイルのその他の内容が前記共有メモリに退避され、

40

前記一方の C P U コアが前記処理を完了すると、前記一方の C P U コアは前記共有メモリに退避された前記他方の C P U コアの前記レジスタファイルの前記その他の内容を読み出して、読み出された前記他方の C P U コアの前記レジスタファイルの前記その他の内容を前記一方の C P U コアの前記レジスタファイルに格納する処理と、前記一方の C P U コ

50

アは、前記他方のC P Uコアの前記レジスタファイルのスタックポインタの内容を前記一方のC P Uコアの記憶回路に転送する処理と、前記一方のC P Uコアは、前記一方の記憶回路に転送された前記スタックポインタの前記内容で指示される前記スタック領域から前記他方のC P Uコアの前記レジスタファイルの前記プログラムカウンタの前記内容と前記ステータスレジスタの前記内容とを読み出して、読み出された前記プログラムカウンタの前記内容と前記ステータスレジスタの前記内容とを前記一方のC P Uコアの前記レジスタファイルのプログラムカウンタとステータスレジスタとにそれぞれ格納する処理とを実行して、

その後、前記一方のC P Uコアは、前記一方のC P Uコアの前記レジスタファイルに格納された前記他方のC P Uコアの前記レジスタファイルの前記その他の内容と前記一方のC P Uコアの前記レジスタファイルの前記プログラムカウンタと前記ステータスレジスタとにそれぞれ格納された前記他方のC P Uコアの前記レジスタファイルの前記プログラムカウンタの前記内容と前記ステータスレジスタの前記内容とを使用することによって、前記他方のC P Uコアによって中断された前記処理を前記一方のC P Uコアが再開して実行するデータ処理装置。10

【請求項 6】

前記第1C P Uコアと前記第2C P Uコアとに接続されたシステムコントローラを更に具備して、

前記システムコントローラは、前記第1のC P Uコアの中断した処理の復帰を抑制する第1フラグ情報と、前記第2のC P Uコアの中断した処理の復帰を抑制する第2フラグ情報とを格納して、20

前記一方のC P Uコアが前記処理を完了すると、前記一方のC P Uコアは前記他方のC P Uコアの中断した処理の復帰を抑制するようにフラグ情報を所定の状態にセットして、

前記一方のフラグ情報が前記所定の状態にセットされることに応答して、前記他方のC P Uコアによる前記共有メモリに退避された前記他方のC P Uコアの前記レジスタファイルの前記その他の内容の読み出しが禁止される一方、前記一方のC P Uコアによる前記共有メモリに退避された前記他方のC P Uコアの前記レジスタファイルの前記その他の内容の読み出しが許可され、前記他方のC P Uコアによる前記スタック領域から前記他方のC P Uコアの前記レジスタファイルの前記プログラムカウンタの前記内容と前記ステータスレジスタの前記内容とを読み出しが禁止されて、前記一方のC P Uコアによる前記スタック領域から前記他方のC P Uコアの前記レジスタファイルの前記プログラムカウンタの前記内容と前記ステータスレジスタの前記内容との読み出しが許可され、読み出された前記プログラムカウンタの前記内容と前記ステータスレジスタの前記内容とを前記一方のC P Uコアの前記レジスタファイルのプログラムカウンタとステータスレジスタとにそれぞれ格納する請求項5に記載のデータ処理装置。30

【請求項 7】

前記一方のフラグ情報が前記所定の状態にセットされることに応答して、前記他方のC P Uコアによって中断された前記処理を前記他方のC P Uコアが再開して実行することが禁止され、前記他方のC P Uコアによって中断された前記処理を前記一方のC P Uコアが再開して実行することが許可される請求項6に記載のデータ処理装置。40

【請求項 8】

前記システムコントローラは前記第1C P Uコアの前記第1レジスタファイルの内容を前記共有メモリの第1領域に格納する一方、前記第2C P Uコアの前記第2レジスタファイルの内容を前記共有メモリの前記第1領域と異なる第2領域に格納するアドレス管理ユニットを含む請求項5から請求項7のいずれかに記載のデータ処理装置。

【請求項 9】

前記データ処理装置は、前記第1C P Uコアと前記第2C P Uコアとに接続された内部メモリを更に具備する一方、メインメモリと接続可能であり、

前記内部メモリと前記メインメモリとの少なくともいづれか一方に、前記他方のC P Uコの前記レジスタファイルの前記プログラムカウンタの前記内容と前記ステータスレジス50

タの前記内容とが退避される前記スタック領域が形成される請求項 5 から請求項 7 のいずれかに記載のデータ処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、複数の C P U コアを有するデータ処理装置に関し、特にマルチ C P U コアのタスクのスケジューリングを容易とするのに有益な技術に関する。

【背景技術】

【0002】

マイクロコンピュータのようなデータ処理装置においては、割込み例外処理や、マルチタスクでのタスク切換え時には、元の状態に復帰するために汎用レジスタもしくはレジスタファイルのステータスレジスタ等のレジスタファイルのデータ退避が行なわれる。退避先としては、外部のメインメモリ等のスタック領域を利用することも可能であるが、レジスタバンクと呼ばれる専用レジスタバンクを用いることでレジスタファイルの退避及び復帰を高速化することができ、リアルタイムな応答を実現できるようになる。

【0003】

下記特許文献 1 にはレジスタバンク方式のシングルチップでオーバーフローフラグを設けることにより、レジスタバンクがすべて使用状態であってもスタック領域に退避することが出来るようになりますことで、効率的に割込み応答が可能なレジスタバンク方式を実現している。

20

【0004】

一方、下記非特許文献 1 には、デュアル C P U コアをチップ上に搭載したプロセッサが記載されている。

【0005】

【特許文献 1】特開 2004 - 157636 号公報

【非特許文献 1】Samuel Naffziger et al., "The Implementation of a 2-Core, Multi-Threaded Itanium Family Processor", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 41, N. 1 JANUARY 2006, PP. 197 - 209.

30

【発明の開示】

【発明が解決しようとする課題】

【0006】

マルチタスク環境での近年の処理要求の増大に対応するために、C P U (中央処理ユニット)を複数備えたマルチ C P U コアのデータ処理装置が検討されている。

【0007】

本発明に先立って本発明者等は、メモリを共有するマルチコア C P U において、割込み等の高速切り換え処理するデータ退避のための前記レジスタバンクを複数の C P U で共有する共有レジスタバンクを採用することを検討した。しかしながら、マルチコア C P U で各 C P U コアに任意にタスクを実行させていくと、条件によってタスクのスケジューリングの問題が発生することが判明した。

40

【0008】

具体的には例えば、共有レジスタバンクを有するマルチコア C P U において、2つの C P U コアの一方の C P U コアが優先度の高い処理を行なっている間に、他方の C P U コアに一方の C P U コアよりも優先度の低い複数の処理要求が連續して発生したとする。スーパースカラー・アーキテクチャー等によるマルチコア C P U においては、一方の C P U コアでの優先度の高い処理と他方の C P U コアでの優先度の低い処理とがそれぞれの C P U コアのレジスタファイルを使用することで実行可能であり、2つの処理の実行のために内蔵 R A M、外部のメインメモリ、入出力装置、周辺装置のハードウェアリソースの使用の競合が発生せずに、優先度の高い処理の結果と優先度の低い処理の結果とが独立であれば

50

、2つの処理は並列に実行可能である。

【0009】

しかし、2つの処理並列の実行条件が満足されなければ、他方のCPUコアでの優先度の低い処理は実行されることができず、中断されることとなる。この中断に際して、最初は他方のCPUコアのレジスタファイルの内容は、共有レジスタバンクに退避される。しかし、処理要求の連続発生によって、退避先の共有レジスタバンクがオーバーフローする。すると、他方のCPUコアのレジスタバンクに中断された処理が、内蔵RAMや外部のメインメモリの複数のスタック領域に退避されることになる。しかる後に一方のCPUコアが優先度の高い処理が終了した時点で、新たな処理要求が無い場合には、処理のストールが生じてタスクのスケジューリングが不可能となる。これは新たな処理要求が与えられていない一方のCPUコアが、他方のCPUコアに関して処理待ちの優先度の低い複数の処理要求が外部のメインメモリの複数のスタック領域に退避されていることを認識できることに起因している。10

【0010】

その結果、他方のCPUコアに関しては処理待ちの優先度の低い複数の処理要求が外部のメインメモリの複数のスタック領域に退避されているにもかかわらず、新たな処理要求が与えられていない一方のCPUコアはアイドル状態（ストール状態）となってしまうことが判明した。

【0011】

本発明は本発明に先立って本発明者等によって行われた上記のような検討の結果を基にしてなされたものであり、その目的とするところは、マルチCPUコアのデータ処理装置においてタスクのスケジューリングを容易とすることにある。20

【0012】

また本発明のその他の目的とするところは、マルチCPUコアのデータ処理装置において優先度の高い処理を行なっていた一方のCPUコアが優先度の高い処理が終了した時点で、一方のCPUコアに新たな処理要求が与えられていない場合の一方のCPUコアでのアイドル状態（ストール状態）を回避することにある。

【0013】

本発明の前記並びにその他の目的と新規な特徴とは、本明細書の記述及び添付図面から明らかになるであろう。30

【課題を解決するための手段】

【0014】

本願において開示される発明のうち代表的なものの概要を簡単に説明すれば、下記の通りである。

【0015】

すなわち、本発明のひとつの形態によるデータ処理装置（1）は、第1CPUコア（2）と、第2CPUコア（3）とを具備する。前記第1CPUコア（2）は第1レジスタファイル（21）と第1命令デコーダ・演算器（22）とを含み、前記第2CPUコア（3）は第2レジスタファイル（31）と第2命令デコーダ・演算器（32）とを含んでいる。前記データ処理装置（1）は、前記第1CPUコア（2）の前記第1レジスタファイル（21）の内容と前記第2CPUコア（3）は前記第2レジスタファイル（31）の内容とを格納する共有メモリ（5）を更に具備する。40

【0016】

前記第1CPUコア（2）は前記第2CPUコア（3）の前記第2レジスタファイル（31）内部のスタックポインタ（SP、R15）の値をあたかも前記第1CPUコア（2）のスタックポインタの値として処理するための第1記憶回路（（Vt1_SP）23）を含み、前記第2CPUコア（3）は前記第1CPUコア（2）の前記第1レジスタファイル（21）内部のスタックポインタ（SP、R15）の値をあたかも前記第2CPUコア（3）のスタックポインタの値として処理するための第2記憶回路（（Vt1_SP）33）を含んでいる。50

【0017】

前記第1CPUコア(2)と前記第2CPUコア(3)との一方のCPUコア(第2CPUコア(3))が優先度の高い処理を実行している間に、前記第1CPUコア(2)と前記第2CPUコア(3)との他方のCPUコア(第1CPUコア(2))に前記一方のCPUコア(第2CPUコア(3))が実行している前記処理よりも優先度の低い要求が発生することによって、優先度の低い前記要求は前記他方のCPUコア(第1CPUコア(2))によって実行されることなく中断され、前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))のプログラムカウンタ(PC)の内容とステータスレジスタ(SR)の内容とがスタック領域に退避される一方、前記他方のCPUコア(第1CPUコア(2))のレジスタファイル(第1レジスタファイル(21))のその他の内容が前記共有メモリ(5)に退避される。10

【0018】

前記一方のCPUコア(第2CPUコア(3))が優先度の高い前記処理を完了すると、前記一方のCPUコア(第2CPUコア(3))は前記共有メモリ(5)に退避された前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))の前記その他の内容を前記一方のCPUコア(第2CPUコア(3))のレジスタファイル(第2レジスタファイル(31))に格納して、前記一方のCPUコア(第2CPUコア(3))は、前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))のスタックポインタ(SP)の内容を前記第1記憶回路((Vt1_SP)23)と前記第2記憶回路((Vt1_SP)33)の一方の記憶回路(第2記憶回路(33))に転送する。次に、前記他方のCPUコア(第1CPUコア(2))によって中断された前記処理を前記一方のCPUコア(第2CPUコア(3))が再開して実行する(図1参照)。20

【0019】

本発明の前記ひとつの形態による手段によれば、マルチCPUコアのデータ処理装置において優先度の高い処理を行なっていた一方のCPUコア(第2CPUコア(3))が優先度の高い処理が終了した時点で、一方のCPUコア(第2CPUコア(3))に新たな処理要求が与えられていない場合に、他方のCPUコア(第1CPUコア(2))によって中断された処理を一方のCPUコア(第2CPUコア(3))が再開して実行するものである。その結果、一方のCPUコア(第2CPUコア(3))でのアイドル状態(ストール状態)を回避することが可能となる。30

【0020】

本発明の1つの好適な形態によるデータ処理装置(1)は、前記第1CPUコア(2)と前記第2CPUコア(3)とに接続されたシステムコントローラ(4)を更に具備する。前記システムコントローラ(4)は、第1フラグ情報(BL0)と第2フラグ情報(BL1)とを格納する。

【0021】

前記一方のCPUコア(第2CPUコア(3))が優先度の高い前記処理を完了すると、前記一方のCPUコア(第2CPUコア(3))は前記第1フラグ情報(BL0)と前記第2フラグ情報(BL1)との一方のフラグ情報(BL1)を所定の状態("1")にセットする。前記一方のフラグ情報(BL1)が前記所定の状態("1")にセットされることに応答して、前記他方のCPUコア(第1CPUコア(2))によって中断された前記処理を前記他方のCPUコア(第1CPUコア(2))が再開して実行することが禁止され、前記他方のCPUコア(第1CPUコア(2))によって中断された前記処理を前記一方のCPUコア(第2CPUコア(3))が再開して実行することが許可される。40

【0022】

本発明の他の1つの更に好適な形態によるデータ処理装置(1)では、前記システムコントローラ(4)は前記第1CPUコア(2)の前記第1レジスタファイル(21)の内容を前記共有メモリ(5)の第1領域(Reg bank 0, 2, 4...)に格納する一方、前記第2CPUコア(3)の前記第2レジスタファイル(31)の内容を前記共有メモリ50

(5) の前記第1領域(Regbank0、2、4...)と異なる第2領域(Regbank1、3、5...)に格納するアドレス管理ユニット(41、42...48、49)を含む。格納する共有バンクメモリ(5)を更に具備する(図1参照)。

【0023】

本発明の前記他の1つの好適な形態による手段によれば、相手側のCPUコアからの前記共有メモリ(5)退避データの上書きによるデータ破壊を回避することが可能である。

【0024】

本発明の最も具体的な形態によるデータ処理装置(1)は、前記第1CPUコア(2)と前記第2CPUコア(3)とに接続された内部メモリ(RAM8)を更に具備する一方、メインメモリと接続可能である。

10

【0025】

前記内部メモリ(RAM8)と前記メインメモリとの少なくともいずれか一方に、前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))の前記プログラムカウンタ(PC)の前記内容と前記ステータスレジスタ(SR)の前記内容とが退避される前記スタック領域が形成される。

【0026】

また、本発明の他のひとつの形態によるデータ処理装置(1)は、第1CPUコア(2)と、第2CPUコア(3)とを具備する。前記第1CPUコア(2)は第1レジスタファイル(21)と第1命令デコーダ・演算器(22)とを含み、前記第2CPUコア(3)は第2レジスタファイル(31)と第2命令デコーダ・演算器(32)とを含んでいる。前記データ処理装置(1)は、前記第1CPUコア(2)の前記第1レジスタファイル(21)の内容と前記第2CPUコア(3)は前記第2レジスタファイル(31)の内容とを格納する共有メモリ(5)を更に具備する。

20

【0027】

前記第1CPUコア(2)は前記第2CPUコア(3)の前記第2レジスタファイル(31)内部のスタックポインタ(SP、R15)の値を格納するための第1記憶回路((Vt1_SP)23)を含み、前記第2CPUコア(3)は前記第1CPUコア(2)の前記第1レジスタファイル(21)内部のスタックポインタ(SP、R15)の値を格納するための第2記憶回路((Vt1_SP)33)を含んでいる。

【0028】

30

前記第1CPUコア(2)と前記第2CPUコア(3)との一方のCPUコア(第2CPUコア(3))が優先度の高い処理を実行している間に、前記第1CPUコア(2)と前記第2CPUコア(3)との他方のCPUコア(第1CPUコア(2))に前記一方のCPUコア(第2CPUコア(3))が実行している前記処理よりも優先度の低い要求が発生することによって、優先度の低い前記要求は前記他方のCPUコア(第1CPUコア(2))によって実行されることなく中断され、前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))のプログラムカウンタ(PC)の内容とステータスレジスタ(SR)の内容とがスタック領域に退避される一方、前記他方のCPUコア(第1CPUコア(2))のレジスタファイル(第1レジスタファイル(21))のその他の内容が前記共有メモリ(5)に退避される。

40

【0029】

前記一方のCPUコア(第2CPUコア(3))が優先度の高い前記処理を完了すると、前記一方のCPUコア(第2CPUコア(3))は前記共有メモリ(5)に退避された前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))の前記その他の内容を読み出して、読み出された前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))の前記その他の内容を前記一方のCPUコア(第2CPUコア(3))のレジスタファイル(第2レジスタファイル(31))に格納する処理と、前記一方のCPUコア(第2CPUコア(3))は、前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))のスタックポインタ(SP)の内容を前記第1

50

記憶回路(((Vt1_S_P)23)と前記第2記憶回路((Vt1_S_P)33)の一方の記憶回路(第2記憶回路(33))に転送する処理と、前記一方のCPUコア(第2CPUコア(3))は、前記一方の記憶回路(第2記憶回路(33))に転送された前記スタックポインタ(S_P)の前記内容で指示される前記スタック領域から前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))の前記プログラムカウンタ(PC)の前記内容と前記ステータスレジスタ(SR)の前記内容とを読み出して、読み出された前記プログラムカウンタ(PC)の前記内容と前記ステータスレジスタ(SR)の前記内容とを前記一方のCPUコア(第2CPUコア(3))の前記レジスタファイル(第2レジスタファイル(31))のプログラムカウンタ(PC)とステータスレジスタ(SR)とにそれぞれ格納する処理とを実行する。 10

【0030】

その後、前記一方のCPUコア(第2CPUコア(3))は、前記一方のCPUコア(第2CPUコア(3))の前記レジスタファイル(第2レジスタファイル(31))に格納された前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))の前記その他の内容と前記一方のCPUコア(第2CPUコア(3))の前記レジスタファイル(第2レジスタファイル(31))の前記プログラムカウンタ(PC)と前記ステータスレジスタ(SR)とにそれぞれ格納された前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))の前記プログラムカウンタ(PC)の前記内容と前記ステータスレジスタ(SR)の前記内容とを使用することによって、前記他方のCPUコア(第1CPUコア(2))によって中断された前記処理を前記一方のCPUコア(第2CPUコア(3))が再開して実行する(図1参照)。 20

【0031】

本発明の前記他のひとつの形態による手段によれば、マルチCPUコアのデータ処理装置において優先度の高い処理を行なっていた一方のCPUコア(第2CPUコア(3))が優先度の高い処理が終了した時点で、一方のCPUコア(第2CPUコア(3))に新たな処理要求が与えられていない場合に、他方のCPUコア(第1CPUコア(2))によって中断された処理を一方のCPUコア(第2CPUコア(3))が再開して実行するものである。その結果、一方のCPUコア(第2CPUコア(3))でのアイドル状態(ストール状態)を回避することが可能となる。 30

【0032】

本発明の他の1つの好適な形態によるデータ処理装置(1)は、前記第1CPUコア(2)と前記第2CPUコア(3)とに接続されたシステムコントローラ(4)を更に具備する。前記システムコントローラ(4)は、第1フラグ情報(BL0)と第2フラグ情報(BL1)とを格納する。

【0033】

前記一方のCPUコア(第2CPUコア(3))が優先度の高い前記処理を完了すると、前記一方のCPUコア(第2CPUコア(3))は前記第1フラグ情報(BL0)と前記第2フラグ情報(BL1)との一方のフラグ情報(BL1)を所定の状態("1")にセットする。 40

【0034】

前記一方のフラグ情報(BL1)が前記所定の状態("1")にセットされることに応答して、前記他方のCPUコア(第1CPUコア(2))による前記共有メモリ(5)に退避された前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))の前記その他の内容の読み出しが禁止される一方、前記一方のCPUコア(第2CPUコア(3))による前記共有メモリ(5)に退避された前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))の前記その他の内容の読み出しが許可され、前記他方のCPUコア(第1CPUコア(2))による前記スタック領域から前記他方のCPUコア(第1CPUコア(2))の前記レジスタファイル(第1レジスタファイル(21))の前記プログラムカ 50

ウンタ(P C)の前記内容と前記ステータスレジスタ(S R)の前記内容とを読み出しが禁止されて、前記一方の C P U コア(第 2 C P U コア(3))による前記スタッカ領域から前記他方の C P U コア(第 1 C P U コア(2))の前記レジスタファイル(第 1 レジスタファイル(2 1))の前記プログラムカウンタ(P C)の前記内容と前記ステータスレジスタ(S R)の前記内容との読み出しが許可され、読み出された前記プログラムカウンタ(P C)の前記内容と前記ステータスレジスタ(S R)の前記内容とを前記一方の C P U コア(第 2 C P U コア(3))の前記レジスタファイル(第 2 レジスタファイル(3 1))のプログラムカウンタ(P C)とステータスレジスタ(S R)とにそれぞれ格納する(図 1 参照)。

【 0 0 3 5 】

10

本発明の前記他の 1 つの好適な形態による手段によれば、マルチタスクのスケジューリングが容易となる。

【 0 0 3 6 】

本発明の他の 1 つの更に好適な形態によるデータ処理装置(1)では、前記一方のフラグ情報(B L 1)が前記所定の状態(" 1 ")にセットされることに応答して、前記他方の C P U コア(第 1 C P U コア(2))によって中断された前記処理を前記他方の C P U コア(第 1 C P U コア(2))が再開して実行することが禁止され、前記他方の C P U コア(第 1 C P U コア(2))によって中断された前記処理を前記一方の C P U コア(第 2 C P U コア(3))が再開して実行することが許可される。

【 発明の効果 】

20

【 0 0 3 7 】

本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば、下記の通りである。

【 0 0 3 8 】

すなわち、本発明によれば、マルチ C P U コアのデータ処理装置においてタスクのスケジューリングを容易とすることができます。

【 0 0 3 9 】

また、本発明によれば、マルチ C P U コアのデータ処理装置において優先度の高い処理を行なっていた一方の C P U コアが優先度の高い処理が終了した時点で、一方の C P U コアに新たな処理要求が与えられていない場合の一方の C P U コアでのアイドル状態(ストール状態)を回避することも可能である。

30

【 発明を実施するための最良の形態 】

【 0 0 4 0 】

マルチ C P U コアのデータ処理装置の構成

図 1 は、本発明の 1 つの実施形態によるマルチ C P U コアのデータ処理装置を示す図である。

【 0 0 4 1 】

同図に示すように、マイクロコンピュータ 1 は、第 1 C P U コア 2 と第 2 C P U コア 3 との 2 つの C P U コアからなるデュアルコアプロセッサである。第 1 C P U コア 2 と第 2 C P U コア 3 には、割り込みやリセットによる C P U の動作制御を行うシステムコントローラ 4 が接続されている。

40

【 0 0 4 2 】

第 1 C P U コア 2 は第 1 レジスタファイル 2 1 と第 1 命令デコーダ・演算器 2 2 を含み、第 2 C P U コア 3 は第 2 レジスタファイル 3 1 と第 2 命令デコーダ・演算器 3 2 を含んでいる。

【 0 0 4 3 】

第 1 C P U コア 2 は専用の第 1 C P U バス 1 2 に接続されて、第 2 C P U コア 3 は専用の第 2 C P U バス 1 3 に接続され、第 1 C P U コア 2 と第 2 C P U コア 3 とは共有のレジスタバンクバス 1 1 を介してバンクメモリ 5 に接続されている。このバンクメモリ 5 は、第 1 C P U コア 2 の第 1 レジスタファイル 2 1 と第 2 C P U コア 3 の第 2 レジスタファイ

50

ル31の退避及び復帰を高速化するレジスタバンクReg_Bkを含んでいる。

【0044】

また、特に本発明の1つの実施形態では、第1CPUコア2は第2CPUコア3の第2レジスタファイル31内部のスタックポインタSPの値をあたかも自己のスタックポインタの値として処理するための第1仮想スタックポインタ(Vt1_SP)23を含み、同様に第2CPUコア3は第1CPUコア2の第1レジスタファイル21内部のスタックポインタSPの値をあたかも自己のスタックポインタの値として処理するための第2仮想スタックポインタ(Vt1_SP)33を含んでいる。良く知られているように、スタックポインタは、サブルーチン実行時や割り込み処理時にレジスタファイルや汎用レジスタの内容を退避するメモリのスタック領域の最上位のアドレスを示すものである。10

【0045】

第1CPUバス12と第2CPUバス13には、デュアルポートの不揮発性フラッシュメモリ等のリードオンリーメモリ(ROM)7と、デュアルポートのステティックランダムアクセスメモリ等のランダムアクセスメモリ(RAM)8と、バスステートコントローラ9とが接続されている。バスステートコントローラ9はシステムバス14にも接続され、システムバス14には割り込みコントローラ6と入出力ポート10とが接続されている。従って、第1CPUコア2は第1CPUバス12を介してROM7、RAM8をアクセスして、更にバスステートコントローラ9を経由してシステムバス14に接続された割り込みコントローラ6と入出力ポート10とをアクセスすることができる。同様に、第2CPUコア3は第2CPUバス13を介してROM7、RAM8をアクセスして、更にバスステートコントローラ9を経由してシステムバス14に接続された割り込みコントローラ6と入出力ポート10とをアクセスすることができる。20

【0046】

割り込みコントローラ6は、外部からの割り込み要因を整理して、システムコントローラ4に通知する。RAM8は、第1CPUコア2及び第2CPUコア3の一時的記憶領域もしくはワーク領域となる。ROM7には、第1CPUコア2及び第2CPUコア3の動作プログラムが格納される。

【0047】

図1の左上と右上とに第1CPUコア2の第1レジスタファイル21と第2CPUコア3の第2レジスタファイル31がそれぞれ示されている。2つの第1レジスタファイル21、31は、同一のレジスタの構成となっており、現在実行中のプログラムのメモリアドレスを指定するプログラムカウンタPCと、サブルーチンプロシージャからの戻りアドレスを格納するプロシージャレジスタPRと、グローバルベースレジスタ間接アドレッシングモードのベースアドレスを示すグローバルベースレジスタGCRと、CPUコアの状態を示すステータスレジスタSRと、乗算と積和演算の結果を格納する2つの積和レジスタMACH、MACLと、データ処理やアドレス計算に使用される16本の汎用レジスタR0...R15とを含んでいる。30

【0048】

16本目の汎用レジスタR15は、ハードウェアスタックポインタSPとして使用される。割り込み処理時や例外処理時のステータスレジスタSRとプログラムカウンタPCの退避、復帰は16本目の汎用レジスタR15(SP)を用いてRAM8や入出力ポート10に接続される外部のメインメモリのスタック領域を参照して実行される。40

【0049】

一方、第1レジスタファイル21、31内部で、破線Sv_Reg_Bkで囲まれたレジスタの内容は、オーバーフローしない限りはバンクメモリ5のレジスタバンクReg_Bkに退避され、バンクメモリ5のレジスタバンクReg_BkがオーバーフローするとRAM8や入出力ポート10に接続される外部のメインメモリのスタック領域に退避される。

【0050】

図1の左下にバンクメモリ5のレジスタバンクReg_Bkの詳細が、示されている。

50

【 0 0 5 1 】

レジスタ退避用のレジスタバンク `Reg_Bk` として、メモリバンク 5 のメモリセルアレイに割り当てられたレジスタバンク `Reg_bank0 ~ Reg_banki` が使用される。バンクメモリ 5 のレジスタバンク `Reg_Bk` に退避されるのは、上述したように 2 つの第 1 レジスタファイル 21、31 の汎用レジスタ `R0 ~ R14` と、グローバルベースレジスタ `GBR` と、2 つの乗算レジスタ `MACH`、`MACL` と、プロシージャレジスタ `PR` 以外に、デバッグ情報としての割り込みに対するベクタ番号 (`IVN`) とされている。

【 0 0 5 2 】

また、本実施形態では、システムコントローラ 4 による論理制御によって、バンクメモリ 5 のレジスタバンク `Reg_Bk` の複数のレジスタバンク `Reg_bank0 ~ Reg_banki` の内部で第 1 CPU コア 2 で割り込み発生時にレジスタ退避するためのレジスタバンクと第 2 CPU コア 3 で割り込み発生時にレジスタ退避するためのレジスタバンクとが区別される。10

【 0 0 5 3 】

図 1 の上中央には、システムコントローラ 4 の詳細な構成が示されている。システムコントローラ 4 は、第 1 CPU コア 2 のための加減算器 41、レジスタバンクレジスタ (`BN0`) 43、演算器 45、バンクロックビットレジスタ (`BL0`) 47 と、第 2 CPU コア 3 のための加減算器 42、レジスタバンクレジスタ (`BN1`) 44、演算器 46、バンクロックビットレジスタ (`BL1`) 48 と、セレクタ 49 とから構成されている。第 1 CPU コア 2 の割り込み退避時のバンクメモリ 5 のレジスタバンク `Reg_Bk` のレジスタバンクを指定するのがレジスタバンクレジスタ `BN0` であり、第 2 CPU コア 3 の割り込み退避時のバンクメモリ 5 のレジスタバンク `Reg_Bk` のレジスタバンクを指定するのがレジスタバンクレジスタ `BN1` である。マイクロコンピュータ 1 のパワーオン時やパワーオンリセット時のシステム初期化時に、レジスタバンクレジスタ `BN0`、`BN1` は共に “0” にリセットされる。20

【 0 0 5 4 】**マルチ CPU コアのデータ処理装置の動作**

割り込み例外処理が発生して、第 1 CPU コア 2 からバンクメモリ 5 のレジスタバンク `Reg_Bk` への第 1 レジスタファイル 21 の所定のレジスタセット `PR`、`GBR`、`MACH`、`MACL`、`R0 ... R14` の内容のレジスタ退避要求が発生したと想定する。すると、レジスタバンクレジスタ `BN0` の値が、演算器 45 によって 2 倍される。その演算結果がセレクタ 49 で選択され、レジスタバンクレジスタ `BN0` の 2 倍の値が指示するバンクメモリ 5 のレジスタバンク `Reg_bank0` に所定のレジスタセットの内容が退避せられる。その後、レジスタバンクレジスタ `BN0` の値は加減算器 41 で +1 インクリメントされてレジスタバンクレジスタ `BN0` に格納される。30

【 0 0 5 5 】

第 1 CPU コア 2 からレジスタ復帰命令 (`RESBANK` 命令) が発行された場合には、レジスタバンクレジスタ `BN0` の値はまず加減算器 41 で -1 ディクリメントされてレジスタバンクレジスタ `BN0` に格納される。続いてレジスタバンクレジスタ `BN0` の値を 2 倍した値がセレクタ 49 で選択されて、バンクメモリ 5 のレジスタバンク `Reg_bank0` から第 1 レジスタファイル 21 の所定のレジスタセット `PR`、`GBR`、`MACH`、`MACL`、`R0 ... R14` へのレジスタ復帰データの転送が行なわれる。40

【 0 0 5 6 】

同様に、第 2 CPU コア 3 で割り込み例外処理が発生して、バンクメモリ 5 のレジスタバンク `Reg_Bk` への第 2 レジスタファイル 31 の所定のレジスタセット `PR`、`GBR`、`MACH`、`MACL`、`R0 ... R14` の内容の退避要求が発生したと想定する。すると、レジスタバンクレジスタ `BN1` の値が、演算器 44 によって 2 倍された後 +1 インクリメントされる。その演算結果がセレクタ 49 で選択され、レジスタバンクレジスタ `BN1` の 2 倍に 1 を足した値が指示するバンクメモリ 5 のレジスタバンク `Reg_bank1` に所定のレジスタセットの内容が退避せられる。その後、レジスタバンクレジスタ `BN1` の値50

は加減算器 4 2 で + 1 インクリメントされてレジスタバンクレジスタ BN 1 に格納される。
。

【 0 0 5 7 】

第 2 C P U コア 3 からレジスタ復帰命令 (R E S B A N K 命令) が発行された場合に、レジスタバンクレジスタ BN 1 の値はまず加減算器 4 3 で - 1 ディクリメントされてレジスタバンクレジスタ BN 1 に格納される。続いてレジスタバンクレジスタ BN 1 の値を 2 倍して 1 足した値の指示するバンクメモリ 5 のレジスタバンク R e g b a n k 1 から第 2 レジスタファイル 3 1 の所定のレジスタセット PR 、 G B R 、 M A C H 、 M A C L 、 R 0 ... R 1 4 へのレジスタ復帰データの転送が行なわれる。このように、バンクメモリ 5 では、偶数番号のレジスタバンク R e g b a n k 0 、 2 、 4 . . . で第 1 C P U コア 2 の第 1 レジスタファイル 2 1 のレジスタセットの退避・復帰処理が行なわれ、奇数番号のレジスタバンク R e g b a n k 1 、 3 、 5 . . . で第 2 C P U コア 3 の第 2 レジスタファイル 3 1 レジスタセットの退避・復帰処理が行なわれることになる。尚、レジスタバンクレジスタ BN 0 、 BN 1 は、次にレジスタセットの退避・復帰処理に使用されるバンクメモリ 5 のレジスタバンクの値を指示することになる。このように本実施形態では、第 1 C P U コア 2 の使用するレジスタバンクと第 2 C P U コア 3 の使用するレジスタバンクとをレジスタバンクレジスタ BN 0 、 BN 1 により指示している。このように、第 1 C P U コア 2 に偶数番号のレジスタバンク、第 2 C P U コア 3 の奇数番号のレジスタバンクを割り当てることで、相手側のコアからのレジスタバンクの上書きによる破壊を回避している。

【 0 0 5 8 】

更に、本実施形態では、第 2 C P U コア 3 が多重割り込み状態にある一方、第 1 C P U コア 2 に実行すべき処理が無い場合に、第 2 C P U コア 3 からバンクメモリ 5 のレジスタバンク R e g _ B k に退避されている処理を第 1 C P U コア 2 が引き継ぐための第 1 仮想スタックポインタ (V t 1 _ S P) 2 3 を第 1 C P U コア 2 が含んでいる。同様に、第 1 C P U コア 2 が多重割り込み状態にある一方、第 2 C P U コア 3 に実行すべき処理が無い場合に、第 1 C P U コア 2 からバンクメモリ 5 のレジスタバンク R e g _ B k に退避されている処理を第 2 C P U コア 3 が引き継ぐための第 2 仮想スタックポインタ (V t 1 _ S P) 3 3 を第 2 C P U コア 3 が含んでいる。

【 0 0 5 9 】

また更に、本実施形態では、この第 1 仮想スタックポインタ (V t 1 _ S P) 2 3 と第 2 仮想スタックポインタ (V t 1 _ S P) 3 3 とに関係してシステムコンローラ 4 は、バンクロックビットレジスタ (B L 0) 4 7 、バンクロックビットレジスタ (B L 1) 4 8 を含んでいる。

【 0 0 6 0 】

次に図 2 の処理フローを参照して、バンクロックビットレジスタ (B L 0) 4 7 、バンクロックビットレジスタ (B L 1) 4 8 により制御される第 1 仮想スタックポインタ (V t 1 _ S P) 2 3 と第 2 仮想スタックポインタ (V t 1 _ S P) 3 3 の機能を説明する。尚、システムコンローラ 4 のバンクロックビットレジスタ (B L 0) 4 7 のバンクロックビット B L 0 とバンクロックビットレジスタ (B L 1) 4 8 のバンクロックビット B L 1 とは、最初はともに " 0 " にリセットされている。

【 0 0 6 1 】

まず、第 2 C P U コア 3 が優先度の高い処理を行なっている間に、第 1 C P U コア 2 に第 2 C P U コア 3 よりも優先度の低い複数の処理要求が連続して発生したとする。優先度の高い処理の実行と優先度の低い処理の実行とが並列に実行できない条件では、優先度の低い処理は第 1 C P U コア 2 により処理されることができず中断される。従って、最初は第 1 C P U コア 2 の第 1 レジスタファイル 2 1 の内容は、バンクメモリ 5 のレジスタバンク R e g _ B k に退避される。しかし、処理要求の連続発生によって、退避先のバンクメモリ 5 のレジスタバンク R e g _ B k がオーバーフローする。すると、第 1 C P U コア 2 の第 1 レジスタファイル 2 1 の内容は、内蔵 R A M 8 や外部のメインメモリの複数のスタック領域に退避されることになる。

10

20

30

40

50

【0062】

その後、ステップ201で第2CPUコア3が優先度の高い処理が終了した時点で、第2CPUコア3に新たな処理要求が与えられていないと、第2CPUコア3はシステムコンローラ4のバンクロックビットレジスタ(BL1)48のバンクロックビットBL1を“0”から“1”にセットしようとする。この時に第1CPUコア2の第1レジスタファイル21のレジスタセットのデータがバンクメモリ5のレジスタバンクReg_Bkに退避されていなければ、レジスタバンクレジスタBN0の値は“0”なので、バンクロックビットBL1はセットされず“0”的ままである。しかし、第1CPUコア2の第1レジスタファイル21のレジスタセットのデータが退避されていればレジスタバンクレジスタBN0の値は“1”以上なので、バンクロックビットレジスタ(BL1)48のバンクロックビットBL1は“1”にセットされる。それにより、第1CPUコア2の新たな割り込みは禁止されて、また第1CPUコア2によるレジスタ復帰命令(RESBANK命令)の実行もストールされる。10

【0063】

続くステップ202で第2CPUコア3はシステムコンローラ4のバンクロックビットレジスタ(BL1)48のバンクロックビットBL1が“1”であるか否か確認する。“0”的場合はステップ203のアイドルルーチンの処理へ分岐して、“1”である場合にはステップ204へ分岐する。

【0064】

ステップ203のアイドルルーチンの処理では、第2CPU3には第1CPUコア2から引き継ぐ処理は無いのでこのフローを終了し、アイドル時に実行すべき処理へ移行する。20

【0065】

ステップ204では、第2CPU3は退避先のバンクメモリ5のレジスタバンクReg_Bkや内蔵RAM8や外部のメインメモリに退避されていたデータを復帰するレジスタ復帰命令(RESBANK命令)を実行する。このレジスタ復帰命令(RESBANK命令)の実行は、バンクロックビットレジスタ(BL1)48のバンクロックビットBL1の“1”へのセットと、第1CPUコア2の第1レジスタファイル21内部のスタックポインタSPの値をあたかも第2CPUコア3の第2レジスタファイル31内部のスタックポインタSPの値であるかのように処理する第2仮想スタックポインタ(Vt1_SP)33の引き継ぎ機能により実現することができる。30

【0066】

すなわち、システムコンローラ4のバンクロックビットレジスタ(BL1)48のバンクロックビットBL1は、“1”にセットされている。従って、第2CPU3によるレジスタ復帰命令(RESBANK命令)の実行においては第2CPU3のためのレジスタバンクレジスタBN1の指示するレジスタバンクからではなく、第1CPUコア2のためのレジスタバンクレジスタBN0の指示するバンクメモリ5のレジスタバンクReg_Bkから、第1CPU1の第1レジスタファイル21のレジスタセットではなく、第2CPU3の第2レジスタファイル21のレジスタセットPR、GBR、MACH、MACL、R0...R14へ、復帰データが格納される。また、1つのレジスタ復帰命令(RESBANK命令)が実行されると、第2CPU3はシステムコントローラ4の第1CPUコア2のためのレジスタバンクレジスタBN0の値を-1デクリメントする。40

【0067】

次のステップ205では、バンクメモリ5のレジスタバンクReg_BkではなくRAM8や入出力ポート10に接続される外部のメインメモリのスタック領域に退避されていた第1CPUコア2の第1レジスタファイル21のステータスレジスタSRの値とプログラムカウンタPCの値とを回復する割り込み復帰命令(RTE命令)が、第2CPUコア3によって実行される。既に説明したように、ステータスレジスタSRとプログラムカウンタPCの退避、復帰は16本目の汎用レジスタR15(SP)を用いてRAM8や入出力ポート10に接続される外部のメインメモリのスタック領域を参照して実行される。50

た、第2CPUコア3による割り込み復帰命令(RET命令)の実行も、第1CPUコア2の第1レジスタファイル21内部のスタックポインタSPの値をあたかも第2CPUコア3の第2レジスタファイル31内部のスタックポインタSPの値であるかのように処理する第2仮想スタックポインタ(Vt1_SP)33の引き継ぎ機能により実現されることができる。まず、第2CPUコア3は、第2仮想スタックポインタ(Vt1_SP)33によって指示される内蔵RAM8や入出力ポート10に接続される外部のメインメモリのスタック領域から第1CPUコア2の第1レジスタファイル21のステータスレジスタSRの値とプログラムカウンタPCの値とを読み出す。次に、第2CPUコア3は、読み出された第1CPUコア2の第1レジスタファイル21のステータスレジスタSRの値とプログラムカウンタPCの値とを第1CPU1の第1レジスタファイル21のステータスレジスタSRとプログラムカウンタPCとではなく第2CPU3の第2レジスタファイル31のステータスレジスタSRとプログラムカウンタPCとに格納する。次に、第2仮想スタックポインタ(Vt1_SP)33の値はステータスレジスタSRの値とプログラムカウンタPCの値のサイズにより減算された後、演算結果は第1CPUコア2の第1レジスタファイル21の16本目の汎用レジスタR15(SP)に格納される。その後、第2仮想スタックポインタ(Vt1_SP)33はクリアされ、システムコンローラ4のバンクロックビットレジスタ(BL1)48のバンクロックビットBL1は"0"にリセットされる。
10

【0068】

従って、第2CPU3によるレジスタ復帰命令(RESBANK命令)と割り込み復帰命令(RET命令)の実行の後には、第1CPUコア2のためのレジスタバンクレジスタBN0の指示するバンクメモリ5のレジスタバンクReg_BKから第2CPU3の第2レジスタファイル21のレジスタセットPR、GBR、MACH、MACL、R0...R14へ復帰データが格納され、第2仮想スタックポインタ(Vt1_SP)33によって指示される内蔵RAM8や入出力ポート10に接続される外部のメインメモリのスタック領域から読み出された第1CPUコア2の第1レジスタファイル21のステータスレジスタSRの値とプログラムカウンタPCの値とが第2CPU3の第2レジスタファイル31のステータスレジスタSRとプログラムカウンタPCとに格納されている。

【0069】

次のステップ206では、第2CPU3の第2レジスタファイル21のレジスタセットPR、GBR、MACH、MACL、R0...R14とステータスレジスタSRとプログラムカウンタPCとへそれぞれ格納された復帰データとステータス情報とプログラムカウンタ値とを使用して、第2CPUコア3は第1CPUコア2によって中断された処理を再開して実行することが可能となる。
30

【0070】

以上の説明では、第2CPUコア3が優先度の高い処理を行なっている間に、第1CPUコア2に第2CPUコア3よりも優先度の低い複数の処理要求が連続して発生して、優先度の低い処理が第1CPUコア2によって実行されずに中断され、その後、第2CPUコア3が優先度の高い処理が終了した時点で、第2CPUコア3は第1CPUコア2によって中断された処理を再開して実行するものである。
40

【0071】

一方、本実施形態のマイクロプロセッサは対照型アーキテクチャーを採用しているので、上記と反対に、第1CPUコア2が優先度の高い処理を行なっている間に、第2CPUコア3に第1CPUコア2よりも優先度の低い複数の処理要求が連続して発生して、優先度の低い処理が第2CPUコア3によって実行されずに中断され、その後、第1CPUコア2が優先度の高い処理が終了した時点で、第1CPUコア2は第2CPUコア3によって中断された処理を再開して実行することは言うまでもないであろう。

【0072】

以上本発明者によってなされた発明を実施形態に基づいて具体的に説明したが、本発明はそれに限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であ
50

ることは言うまでもない。

【0073】

例えば、図1において、第1CPUコア2にレベル1の第1のキャッシュメモリを接続して、第2CPUコア3にレベル1の第1のキャッシュメモリを接続して、バンクメモリ5のいくつかのレジスタバンクをレベル2の共有キャッシュメモリとして使用することも可能である。

【0074】

更に、マイクロプロセッサ1内部のCPUコアは2個以外の例えは4個としたマルチCPUコアとすることも可能であり、更に極めて多数のCPUコアを接続した超並列アーキテクチャーにも適用することも可能である。

10

【0075】

また本発明は、マイクロコントローラやマイクロプロセッサ以外に各種の用途に使用されるシステムLSIやディジタル・アナログ・ミックスド・シグナルLSIのように、マルチメディア画像データの高速処理のために複数のCPUコアを含むLSI全般に適用可能であることは言うまでもないであろう。

【産業上の利用可能性】

【0076】

本発明は、複数のCPUコアを含むマルチCPUコアアーキテクチャーに広く実施することができる。

20

【図面の簡単な説明】

【0077】

【図1】本発明の1つの実施形態によるマルチCPUコアのデータ処理装置を示す図である。

【図2】図2は、図1に示したマルチCPUコアのデータ処理装置の動作を説明するための処理フローである。

【符号の説明】

【0078】

- 1 データプロセッサ
- 2、3 CPUコア
- 4 システムコントローラ
- 5 バンクメモリ
- 21、31 レジスタファイル
- 22、32 命令デコーダ&演算回路部
- 23 第1仮想スタックポインタ(Vt1_SP)
- 33 第2仮想スタックポインタ(Vt2_SP)
- 6 割込みコントローラ
- 7 ROM
- 8 RAM
- 9 バスステートコントローラ(BSC)

30

- 10 I/Oポート
- 11 レジスタバンクバス
- 12、13 CPUバス
- 14 システムバス

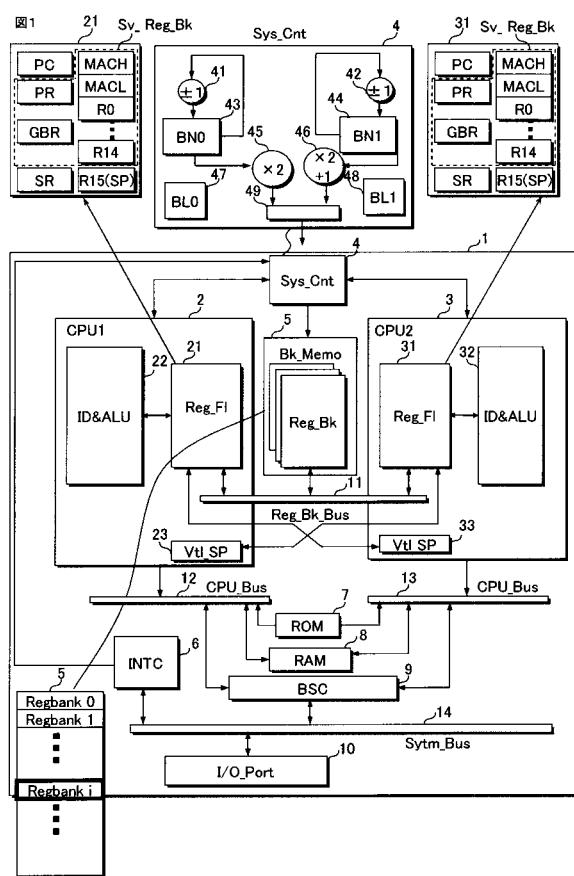
40

- 21、31 レジスタセット
- R0～R15 汎用レジスタ
- SP スタックポインタ
- MACH、MACL 積和レジスタ
- PR プロシージャレジスタ
- GBR グローバルベースレジスタ
- SR ステータスレジスタ

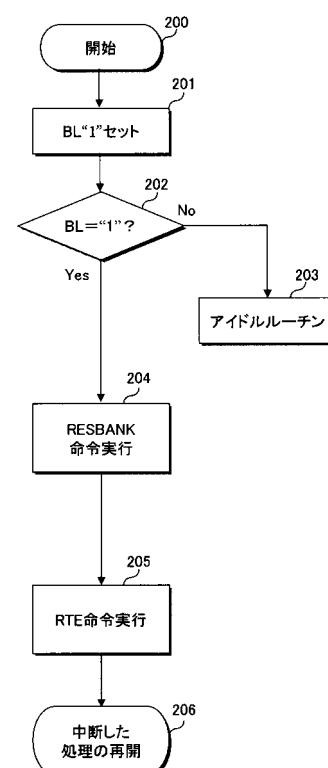
50

- 4 1、4 2 加減算器
 4 3、4 4 レジスタバンクレジスタ
 4 5、4 6 演算器
 4 7、4 8 バンクロックビットレジスタ
 4 9 セレクタ

【図1】



【図2】



フロントページの続き

(56)参考文献 特開2003-323309(JP,A)

特開平1-288954(JP,A)

特開2004-157636(JP,A)

(58)調査した分野(Int.Cl., DB名)

G 06 F	9 / 46	-	9 / 54
G 06 F	15 / 16	-	15 / 177