(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2010-524140
(P2010-524140A)
(43)公表日　平成22年7月15日(2010.7.15)

| (51)Int.Cl. | | F I | | | テーマコード（参考） |
|---|---|---|---|---|---|
| G06F 11/28 | (2006.01) | G06F | 11/28 | 310B | 5B033 |
| G06F 9/52 | (2006.01) | G06F | 9/46 | 475A | 5B042 |
| G06F 9/30 | (2006.01) | G06F | 9/30 | 350F | |

審査請求 有　予備審査請求 未請求　（全 32 頁）

| | |
|---|---|
| (21)出願番号 | 特願2010-503252 (P2010-503252) |
| (86)(22)出願日 | 平成20年4月11日 (2008.4.11) |
| (85)翻訳文提出日 | 平成21年12月14日 (2009.12.14) |
| (86)国際出願番号 | PCT/US2008/060117 |
| (87)国際公開番号 | WO2008/128107 |
| (87)国際公開日 | 平成20年10月23日 (2008.10.23) |
| (31)優先権主張番号 | 11/734,199 |
| (32)優先日 | 平成19年4月11日 (2007.4.11) |
| (33)優先権主張国 | 米国 (US) |

(71)出願人　595020643
クゥアルコム・インコーポレイテッド
ＱＵＡＬＣＯＭＭ　ＩＮＣＯＲＰＯＲＡＴＥＤ
アメリカ合衆国、カリフォルニア州　９２１２１－１７１４、サン・ディエゴ、モアハウス・ドライブ　５７７５
(74)代理人　100058479
弁理士　鈴江　武彦
(74)代理人　100108855
弁理士　蔵田　昌俊
(74)代理人　100091351
弁理士　河野　哲
(74)代理人　100088683
弁理士　中村　誠

最終頁に続く

(54)【発明の名称】マルチスレッドプロセッサのためのインタースレッドトレースアライメント方法およびシステム

(57)【要約】
通信（例えばＣＤＭＡ）システムにおける送信を処理することを含む（しかし制限されない）デジタル信号プロセッサの設計および使用のための技術。実行トレース処理によりインタースレッドトレースアライメントは共通所定イベントに関するタイミングデータを記録することを含んでいる。上記のイベントは、最後のスレッドが実行トレーシングを開始してからのサイクル数であってもよいし、またはスレッドがすべて実行トレーシングを終了してからのサイクル数であってもよい。スレッドが実行トレーシングを開始するサイクル数は、実行トレーシングのタイミングを維持するための共通所定イベントに参照される。その後、共通所定イベントに関するデータは、スレッドが実行トレーシングを開始した時に関連づけるために更新される。その結果はすべてのスレッドに関連したタイミングデータを整列させることを可能にすることである。相互関係のある記録は、すべての動作するスレッドに関するタイミングデータを同期させることと同様に、マルチスレッドプロセッサにおいて動作するスレッドに関する相互依存の実行トレーシング情報を再

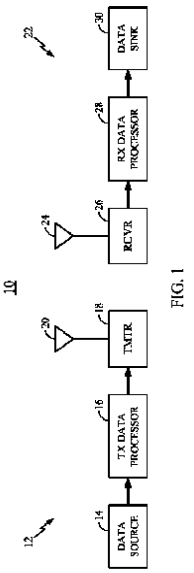

FIG. 1

【特許請求の範囲】
【請求項１】
実行トレーシングプロセスの間にマルチスレッドプロセッサのスレッド間のインタースレッドトレースタイミングアライメントのための方法であって、
共通所定イベントに関するタイミングデータを記録し、
前記共通所定イベントはコアプロセッサ実行トレーシング中に前記マルチスレッドプロセッサの全ての動作スレッドによって参照可能であり、
前記共通所定イベントに関連するスレッドに関する実行トレーシングのタイミングを維持するための前記共通所定イベントへの実行トレーシングを前記スレッドが開始する時間を参照し、
前記スレッドが実行トレーシングを開始した前記時間に関連づけるために、前記共通所定イベントを更新し、その結果、実行トレーシングが発生している前記マルチスレッドプロセッサの全ての他のスレッドに関連したタイミングデータに実行トレーシングを前記スレッドが開始した前記時間を整列させる方法。
【請求項２】
前記共通所定イベントを、最後のスレッドが実行トレーシングをオンした時間に関連づけることをさらに具備する請求項１の方法。
【請求項３】
前記共通所定イベントを、全てのスレッドが実行トレーシングをオフしてからのマルチスレッドプロセッササイクル数に関連づけることをさらに具備する請求項１の方法。
【請求項４】
複数のデータパケットに前記共通所定イベントを記録することをさらに具備する請求項１の方法。
【請求項５】
前記共通所定イベントに関連した複数のデータパケットを用いて、前記マルチスレッドプロセッサにおいて動作するスレッドに関して相互関係のある実行トレーシング情報を再構成することをさらに具備する請求項１の方法。
【請求項６】
前記共通所定イベントに関する前記タイミングデータを同期させることをさらに具備する請求項１の方法。
【請求項７】
前記共通所定イベントの発生からのサイクルのグローバルカウントを生成することをさらに具備する請求項１の方法。
【請求項８】
前記共通所定イベントからのサイクルのグローバルカウントを用いて、実行トレーシングを行うすべてのスレッド間でのインタースレッドタイミングデータを再確立することをさらに具備する請求項１の方法。
【請求項９】
データ損失の前記イベントにおいて、データ損失を決定し、前記データ損失に応じて前記共通所定イベントからのサイクルのグローバルカウントを用いて、実行トレーシングを行うすべてのスレッド間でインタースレッドタイミングデータを再確立することをさらに具備する請求項１の方法。
【請求項１０】
ゼロ値を介して繰り返されるグローバルカウンタのイベントにおいてすべてのスレッドに関する同期パケットを生成することをさらに具備する請求項１の方法。
【請求項１１】
デジタル信号プロセッサと関連する動作に関するデジタル信号プロセッサデバッギングシステムであって、マルチスレッドソフトウェア実行フローの間にマルチスレッドプロセッサのスレッドに関して相互に関係するタイミングデータへの能力を含むシステムであって、

共通所定イベントに関するタイミングデータを記録するための複数のオフセットフィールドと、
前記共通所定イベントはコアプロセッサ実行トレーシング中に前記マルチスレッドプロセッサの全ての動作スレッドによって参照可能であり、
前記共通所定イベントに関連するスレッドに関する実行トレーシングのタイミングを維持するための前記共通所定イベントへの実行トレーシングを前記スレッドが開始する時間を参照するための複数の時間参照命令パケットと、
前記スレッドが実行トレーシングを開始した前記時間に関連づけるために、前記共通所定イベントを更新し、その結果、実行トレーシングが発生している前記マルチスレッドプロセッサの全ての他のスレッドに関連したタイミングデータに実行トレーシングを前記スレッドが開始した前記時間を整列させるための複数の時間参照更新パケットと、を具備するデジタル信号プロセッサデバッギングシステム。

【請求項１２】
前記共通所定イベントを、最後のスレッドが実行トレーシングをオンした時間に関連づけるための複数の命令および回路網をさらに具備する請求項１１のデジタル信号プロセッサデバッギングシステム。

【請求項１３】
前記共通所定イベントを、全てのスレッドが実行トレーシングをオフしてからのマルチスレッドプロセッササイクル数に関連づけるための複数の命令および回路網をさらに具備するデジタル信号プロセッサデバッギングシステム。

【請求項１４】
複数のデータパケットに前記共通所定イベントを記録するための複数の命令および回路網をさらに具備する請求項１１のデジタル信号プロセッサデバッギングシステム。

【請求項１５】
前記共通所定イベントに関連した複数のデータパケットを用いて、前記マルチスレッドプロセッサにおいて動作するスレッドに関して相互関係のある実行トレーシング情報を再構成するための複数の命令および回路網をさらに具備する請求項１１のデジタル信号プロセッサデバッギングシステム。

【請求項１６】
前記共通所定イベントに関する前記タイミングデータを同期させるための複数の命令および回路網をさらに具備する請求項１１のデジタル信号プロセッサデバッギングシステム。

【請求項１７】
前記共通所定イベントの発生からのサイクルのグローバルカウントを生成するための複数の命令および回路網をさらに具備する請求項１１のデジタル信号プロセッサデバッギングシステム。

【請求項１８】
前記共通所定イベントからのサイクルのグローバルカウントを用いて、実行トレーシングを行うすべてのスレッド間でのインタースレッドタイミングデータを再確立するための複数の命令および回路網をさらに具備する請求項１１のデジタル信号プロセッサデバッギングシステム。

【請求項１９】
データ損失を決定し、前記データ損失に応じて前記共通所定イベントからのサイクルのグローバルカウントを用いて、実行トレーシングを行うすべてのスレッド間でインタースレッドタイミングデータを再確立するための実行トレーシング命令をさらに具備する請求項１１のデジタル信号プロセッサデバッギングシステム。

【請求項２０】
ゼロ値を介して繰り返されるグローバルカウンタのイベントにおいてすべてのスレッドに関する同期パケットを生成するための複数の同期命令をさらに具備する請求項１１のデジタル信号プロセッサデバッギングシステム。

【請求項２１】
パーソナル電子デバイスを支援する動作に関するマルチスレッドデジタル信号プロセッサであって、実行トレーシングプロセスを行い、これに関連して前記マルチスレッドプロセッサのスレッド中でインタースレッドトレースタイミングを整列させるデバッギング手段を具備しているプロセッサにおいて、
共通所定イベントに関するタイミングデータを記録する手段と、
前記共通所定イベントはコアプロセッサ実行トレーシング中に前記マルチスレッドプロセッサの全ての動作スレッドによって参照可能であり、
前記共通所定イベントに関連するスレッドに関する実行トレーシングのタイミングを維持するための前記共通所定イベントへの実行トレーシングを前記スレッドが開始する時間を参照する手段と、
前記スレッドが実行トレーシングを開始した前記時間に関連づけるために、前記共通所定イベントを更新し、その結果、実行トレーシングが発生している前記マルチスレッドプロセッサの全ての他のスレッドに関連したタイミングデータに実行トレーシングを前記スレッドが開始した前記時間を整列させる手段と、を具備するデジタル信号プロセッサシステム。
【請求項２２】
前記共通所定イベントを、最後のスレッドが実行トレーシングをオンした時間に関連づける手段をさらに具備する請求項２１のデジタル信号プロセッサシステム。
【請求項２３】
前記共通所定イベントを、全てのスレッドが実行トレーシングをオフしてからのマルチスレッドプロセッササイクル数に関連づける手段をさらに具備する請求項２１のデジタル信号プロセッサシステム。
【請求項２４】
複数のデータパケットに前記共通所定イベントを記録する手段をさらに具備する請求項２１のデジタル信号プロセッサシステム。
【請求項２５】
前記共通所定イベントに関連した複数のデータパケットを用いて、前記マルチスレッドプロセッサにおいて動作するスレッドに関して相互関係のある実行トレーシング情報を再構成する手段をさらに具備する請求項２１のデジタル信号プロセッサシステム。
【請求項２６】
前記共通所定イベントに関する前記タイミングデータを同期させるための複数の命令および回路網をさらに具備する請求項２１のデジタル信号プロセッサシステム。
【請求項２７】
前記共通所定イベントの発生からのサイクルのグローバルカウントを生成する手段をさらに具備する請求項２１のデジタル信号プロセッサシステム。
【請求項２８】
前記共通所定イベントからのサイクルのグローバルカウントを用いて、実行トレーシングを行うすべてのスレッド間でのインタースレッドタイミングデータを再確立する手段をさらに具備する請求項２１のデジタル信号プロセッサシステム。
【請求項２９】
データ損失を決定し、前記データ損失に応じて前記共通所定イベントからのサイクルのグローバルカウントを用いて、実行トレーシングを行うすべてのスレッド間でインタースレッドタイミングデータを再確立する手段をさらに具備する請求項２１のデジタル信号プロセッサシステム。
【請求項３０】
ゼロ値を介して繰り返されるグローバルカウンタのイベントにおいてすべてのスレッドに関する同期パケットを生成する手段をさらに具備する請求項２１のデジタル信号プロセッサシステム。
【請求項３１】

実行トレーシングプロセスを行うことと、これに関連してマルチスレッドプロセッサのスレッド中でインタースレッドトレースタイミングを整列させることを含むマルチスレッドデジタル信号プロセッサをデバッギングするコンピュータ読取可能な具現化されるプログラムコード手段を有するコンピュータ使用可能な媒体であって、

共通所定イベントに関するタイミングデータを記録するコンピュータ読取可能なプログラムコード手段と、

前記共通所定イベントはコアプロセッサ実行トレーシング中に前記マルチスレッドプロセッサの全ての動作スレッドによって参照可能であり、

前記共通所定イベントに関連するスレッドに関する実行トレーシングのタイミングを維持するための前記共通所定イベントへの実行トレーシングを前記スレッドが開始する時間を参照するコンピュータ読取可能なプログラムコード手段と、

前記スレッドが実行トレーシングを開始した前記時間に関連づけるために、前記共通所定イベントを更新し、その結果、実行トレーシングが発生している前記マルチスレッドプロセッサの全ての他のスレッドに関連したタイミングデータに実行トレーシングを前記スレッドが開始した前記時間を整列させるコンピュータ読取可能なプログラムコード手段と、を具備するコンピュータ使用可能な媒体。

【請求項３２】

前記共通所定イベントを、最後のスレッドが実行トレーシングをオンした時間に関連づけるコンピュータ読取可能なプログラムコード手段をさらに具備する請求項３１のコンピュータ使用可能な媒体。

【請求項３３】

前記共通所定イベントを、全てのスレッドが実行トレーシングをオフしてからのマルチスレッドプロセッササイクル数に関連づけるコンピュータ読取可能なプログラムコード手段をさらに具備する請求項３１のコンピュータ使用可能な媒体。

【請求項３４】

複数のデータパケットに前記共通所定イベントを記録するコンピュータ読取可能なプログラムコード手段をさらに具備する請求項３１のコンピュータ使用可能な媒体。

【請求項３５】

前記共通所定イベントに関連した複数のデータパケットを用いて、前記マルチスレッドプロセッサにおいて動作するスレッドに関して相互関係のある実行トレーシング情報を再構成するコンピュータ読取可能なプログラムコード手段をさらに具備する請求項３１のコンピュータ使用可能な媒体。

【発明の詳細な説明】

【技術分野】

【０００１】

開示された主題は、データ通信および同じようなアプリケーションでの使用を見つけてもよいような、データ処理システムとプロセスに関する。より詳しくは、この開示は、マルチスレッドプロセッサのためのインタースレッドトレースアライメント方法およびシステムを提供することを含むデジタル信号処理デバッギング動作のための新しく改善された方法およびシステムに関する。

【０００２】

本出願は次の同時係属の米国特許出願番号と関係する。NON-INTRUSIVE, THREAD-SELECTIVE, DEBUGGING METHOD AND SYSTEM FOR A MULTI-THREAD DIGITAL SIGNAL PROCESSORと題され、２００６年１１月１５日に出願された出願番号１１／５６０，２１７と、METHOD AND SYSTEM FOR A DIGITAL SIGNAL PROCESSOR DEBUGGING DURING POWER TRANSITIONSと題され、２００６年１１月１５日に出願された出願番号１１／５６０，３２３と、METHOD AND SYSTEM FOR TRUSTED/UNTRUSTED DIGITAL SIGNAL PROCESSOR DEBUGGING OPERATIONSと題され、２００６年１１月１５日に出願された出願番号１１／５６０，３３２と、EMBEDDED TRACE MACROCELL FOR ENHANCED DIGITAL SIGNAL PROCESSOR DEBUGGING OPERATIONSと題され、２００６年１１月１５日に出願された出願番号１１／５６０，３３９と、METHOD

AND SYSTEM FOR INSTRUCTION STUFFING OPERATIONS DURING NON-INTRUSIVE DIGITAL SIGNAL PROCESSOR DEBUGGINGと題され、２００６年１１月１５日に出願された出願番号１１／５６０，３４４。

【背景技術】

【０００３】

テレコミュニケーション、他の型の電子機器およびサポートビデオ、合成音声、テレビ会議、および他の豊富なソフトウェアアプリケーションは、ますます信号処理を含んでいる。信号処理は、複雑だが反復のアルゴリズムにおいて、高速の数学的な計算およびデータ生成を必要とする。多くのアプリケーションがリアルタイムな計算（すなわち、信号が時間の連続関数であり、それはサンプリングされ、数値処理のためのデジタル信号に変換されなければならない）を必要とする。プロセッサは、計算が達するように、サンプルについて個別の計算を行うアルゴリズムを実行しなければならない。

【０００４】

デジタル信号プロセッサ（ＤＳＰ）のアーキテクチャは上記のアルゴリズムを処理するために最適化される。よい信号処理エンジンの特性は、高速の融通性のある演算計算ユニット、計算ユニットへのまたは計算ユニットからの制約のないデータフロー、計算ユニットでの拡張精度およびダイナミックレンジ、デュアルアドレス生成器、効率的なプログラムシーケンス、およびプログラミングの容易さを含んでいる。

【０００５】

ＤＳＰ技術の１つの有望なアプリケーションは、テキストメッセージングおよび他のアプリケーションと同様に、衛星または地球上のリンク上のユーザー間で、音声およびデータ通信をサポートする符号分割多元接続（ＣＤＭＡ）システムのような通信システムを含んでいる。多重アクセス通信システムにおけるＣＤＭＡ技術の使用は、“SPREAD SPECTRUM MULTIPLE ACCESS COMMUNICATION SYSTEM USING SATELLITEあるいはTERRESTRIAL REPEATERS”と題された米国特許４，９０１，３０７番と、“SYSTEM AND METHOD FOR GENERATING WAVEFORMS IN A CDMA CELLULAR TELEHANDSET SYSTEM”と題された米国特許５，１０３，４５９番と、に開示されていて、両方はクレームされた主題の譲受人に譲渡されている。

【０００６】

ＣＤＭＡシステムは、１以上の標準規格に準拠することを典型的には設計されている。１つの上記の第１世代標準規格は、“TIA/EIA/IS-95 Terminal-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System”であり、以下ＩＳ－９５標準規格と称する。ＩＳ－９５ＣＤＭＡシステムは音声データとパケットのデータを送信することができる。より効率的にパケットデータを送信する可能性があるより新規の世代標準規格は、“3rd Generation Partnership Project”（３ＧＰＰ）と名付けたコンソーシアムによって提示され、文献番号３Ｇ ＴＳ ２５．２１１、３Ｇ ＴＳ ２５．２１２、３Ｇ ＴＳ ２５．２１３、および３Ｇ ＴＳ ２５．２１４（これらは公に即座に入手可能である）を含む１セットの文献に具体化されている。３ＧＰＰ標準規格は以下ではＷ－ＣＤＭＡ標準規格と称す。

【０００７】

Ｗ－ＤＣＭＡ標準規格を用いる複雑なＤＳＰオペレーショナルソフトウェアは例えば、ロバストな開発ツールを必要とする。上記の開発ツールは、コード生成、統合、試験、デバッギング、およびアプリケーションのパフォーマンス評価することに関するツールを含んでもよい。高度なテレコミュニケーションアプリケーションのような、開発中かつ動作するソフトまたは複雑なＤＳＰアプリケーションでは、精巧であるにもかかわらず、邪魔しないデバッギングソフトウェアの必要がある。すなわち、ソフトウェアアプリケーションをデバッグすることは、ソフトウェアの欠陥および運用上の問題の修正を監視し、試験し、かつサポートするためには単に十分にロバストではなくてはならない。同時に、ソフトウェアをデバッグすることは、同時デバッギング動作中にコアプロセッサソフトウエア動作に干渉しないように動作する必要がある可能性がある。そうでなければ、コア処理ソ

フトウェアにおける任意の問題は、デバッグ動作の間に検出されないか、適切に検出される可能性がある。

【０００８】

デバッグ動作中に、ＤＳＰの内の動作スレッドの処理の動作をトレースするための邪魔しないソフトウェア動作トレーシング機能に関連づける必要がある。上記のシステムは、特定のイベントが発生する前および後の両方に情報を捕獲するためのＤＳＰの状態パラメーターについてのそのような情報を提供してもよい。同時に望ましいトレーシング機能は、ＤＳＰが全力で動作している間にさえ、プロセッサの性能に重大な負担を加えることができない。邪魔しないデバッグ動作と一緒に、上記のトレーシングプロセスは特定の型の情報を捕えてもよい。従ってトレーシング機能は、マルチスレッドプロセッサにおいて邪魔しないデバッグ動作と共に、モニタリングし記録することを提供する。

【０００９】

トレーシング機能が提供してもよい情報の特に有用な１つのセットは、インタースレッド実行動作を含んでいる。すなわち、マルチスレッドＤＳＰの異なるスレッド間で相互に関係するトレーシングデータの能力がある１セットのトレースする機能について要求がある。既知のシステムは上記の情報を提供しない。ソフトウェア動作をデバッグする際に、ユーザーは、マルチスレッドプロセッサが特定の時点で実行している可能性がある命令がどれかを知るために任意の時点を選ぶことを望むかもしれない。この情報は、異なるスレッドが異なる時刻で異なるスレッドごとのデバッギング動作を起動するインスタンスにおいて特に価値がある可能性がある。

【００１０】

インタリーブされたマルチスレッドＤＳＰにおいて、多重命令シーケンスの実行が同時に発生してもよい。そのようなものとして、プロセッサはいくつかの単一スレッドされたプロセッサが独立に動作するとして見なされてもよい。一旦、上記のプロセッサは、動作スレッドの各々のランタイム実行シーケンスを記録する実行トレーシングユニットを含んでいてもよい。これらのトレースは、プログラムフローをパケットのシーケンスに分解することにより、プログラムデバッグ動作を容易にする。このようなシステムでは、スレッド数フィールドは、どのパケットがどのスレッドに属するかを識別するために、あるパケットに加えられてもよい。このようなアプローチで、特定のスレッドに関するパケットシーケンスはそれぞれ、すべてのプログラムフロー変更およびすべての命令タイミングを含む全実行シーケンスを再作成してもよい。

【００１１】

どのパケットがどのスレッドに属することが有利であるかを識別する一方、トレーシング機能は、実行トレーシング中にインタースレッドタイミング関係を識別する能力を提供しない。例えば、トレース動作間の時間差は、あるスレッドから別のスレッドまで非常に大きいかもしれない。スレッドトレーシングが異なる時間にある異なるスレッドに関して開始する場合、トレースされる他のスレッドで１つのスレッドのタイミングを整列させることは可能ではないかもしれない。

【００１２】

従って、異なるＤＳＰスレッド間でタイミング関係を確立し維持する邪魔しないデバッグプロセス内で動作することができる１セットのトレース機能のための要求がある。

【００１３】

しかし、マルチスレッドプロセッサの埋め込まれたトレースマクロセルプロセスでの使用に関してマルチスレッドプロセッサの異なるスレッドのアライメントを許す方法およびシステムについてさらなる要求がある。

【００１４】

またさらに、コアプロセッサソフトウエア動作中に発生する邪魔しないインシリコンデバッグプロセスの広いアレイと共同して動作することができるインタリーブされたマルチスレッドプロセッサのためのインタースレッドトレースアライメント方法およびシステムについての要求がある。

【発明の概要】
【００１５】
マルチスレッドプロセッサにおいてインタースレッドトレースアライメントを提供するための技術は開示される。その技術は、異なるスレッド間の種々のタイミング関係を確立し維持するために、インタースレッドタイミング関係を識別するための埋め込まれたトレースマクロセルに協力する。ここで開示された方法およびシステムは、関連したデジタルプロセッサ速度およびサービス品質を増大させるのと同様に、パーソナルコンピュータ、携帯情報端末、無線ハンドセットおよび同じような電子機器において動作するアプリケーションを含むますます強力なソフトウェアアプリケーションのために、デジタル信号プロセッサの動作とデジタル信号プロセッサ命令の効率的な使用との両方を改善する。開示された主題の一の態様によれば、方法とシステムは、共通所定イベントに関する記録タイミングデータを含んでいる実行トレース処理でインタースレッドトレースアライメントのために提供される。このような共通所定イベントは、最後のスレッドが実行トレーシングを開始してからのサイクル数であってもよく、または全てのスレッドが実行トレーシングを終了してからのサイクル数でもよい。スレッドが実行トレーシングを開始するサイクル数は、実行トレーシングのタイミングを維持するための共通所定イベントに参照される。その後、共通所定イベントに関するデータは、スレッドが実行トレーシングを開始した時と関連づけるために更新される。結果はすべてのスレッドに関連したタイミングデータを整列させることを可能にすることである。相互関係があったレコードは、すべての動作するスレッドについてタイミングデータを同期させると同様に、マルチスレッドプロセッサにおいて動作するスレッドに関する相互依存の実行トレーシング情報を再構成することも可能にする。
【００１６】
開示された主題のこれらおよび他の利点は、追加の新規な特徴と同様に、ここで提供される詳細な説明から明白になる。この要約の意図は、クレームされた主題の包括的な詳細な説明であるのではなく、むしろ主題の機能性のうちのいくらかの短い概要を提供することである。ここで提供される他のシステム、方法、特徴および利点は、以下の図および詳細な説明の試験上では当業者にとって明白になる。この詳細な説明内に含まれる特徴および利点、このような追加のシステム、および方法すべてが伴うクレームの範囲内であることが意図される。
【図面の簡単な説明】
【００１７】
開示された主題の特徴、性質および利点は、参照文字のようなものが全体にわたって対応して同一になる図面と共に得られたときに以下で述べられる詳細な説明からより明白になってもよい。
【図１】図１はここで開示された種々の実施形態のうちの１つを実装してもよい通信システムの単純化されたブロック図である。
【図２】図２は本開示の教えを先へ進めるためのＤＳＰアーキテクチャを示す。
【図３】図３はこの開示された主題の技術的な利点を組込むマルチスレッドデジタル信号プロセッサの一実施形態のアーキテクチャブロック図を提供する。
【図４】図４は本開示のＩＳＤＢ／ＪＴＡＧインターフェースの特徴を適用するデジタル信号プロセッサコアのある態様を開示する。
【図５】図５は本開示が関係する動作のデバッギングモードを含む、デジタル信号プロセッサのオペレーティングモードに適用可能なプロセスフロー図を与える。
【図６】図６は本開示の埋め込まれたトレースマクロセルの全体の機能概観を表すブロック図を提供する。
【図７】図７は、開示された埋め込まれたトレースマクロセルプロセスおよびシステムのトリガーブロック回路を示す。
【図８】図８はこの開示されたインタースレッドトレースアライメントプロセスの重要な概念を示す。

【図９】図９は開示された主題において適用可能なものとしてパケット生成ユニットの機能概要を示す。
【図１０】図１０は現在開示された主題の命令に適用可能な種々のアトムに関する定義の表を示す。
【図１１】図１１は本開示に役立つものとして３２ビットのＴＩＤフィールドをサポートする本開示に関するブランチアドレスパケットを提供する。
【図１２】図１２は現在開示されたプロセスで使用されるｉｓｙｎｃ－ｒｅｓｔａｒｔパケットの典型的な内容を与える。
【図１３】図１３は現在開示された主題をサポートするためのｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットの実施形態を表す。
【図１４】図１４はこの開示されたプロセスで使用されるサイクルカウントパケットを与える。
【図１５】図１５は本開示に適切なものとしてｐｈｅａｄｅｒフォーマット４パケットの一実施形態を示す。
【図１６】図１６は開示された主題の実施形態により使用のためのａｓｙｎｃパケットを表す。
【図１７】
【発明を実施するための形態】
【００１８】
マルチスレッドデジタル信号プロセッサの実行トレースプロセスに関連した使用のためのインタースレッドトレースアライメント方法およびシステムは、ここで与えられた利点が有利になる可能性があるあらゆる型のマルチスレッド処理に関するアプリケーションを持っている。上記の１つのアプリケーションが、テレコミュニケーション、および特に１以上のデジタル信号処理回路を用いる無線ハンドセットに現れる。上記の無線ハンドセットがどのように用いられてもよいかを説明することについて、図１は、開示された割込み処理方法およびシステムの本実施形態を実装してもよい通信システム１０の単純化されたブロック図を提供する。送信機ユニット１２では、データは、データソース１４から、１以上のアナログ信号を生成するためにデータをフォーマットし、符号化し、処理する送信（ＴＸ）データプロセッサ１６まで、典型的にはブロックで送られる。その後、アナログ信号は、変調された信号を生成するために、ベースバンド信号を変調し、フィルターし、増幅し、アップコンバートする送信機（ＴＭＴＲ）１８に提供される。その後、変調された信号は、１以上の受信機ユニットにアンテナ２０を介して送信される。
【００１９】
受信機ユニット２２では、送信された信号は、アンテナ２４によって受信され、受信機（ＲＣＶＲ）２６に提供される。受信機２６内では、受信信号は、フェーズ（Ｉ）および（Ｑ）サンプルにおいて生成するために、増幅され、フィルターされ、ダウンコンバートされ、復調され、デジタル化される。その後サンプルは、送信データを回復するために、受信（ＲＸ）データプロセッサ２８によってデコードされ処理される。受信機ユニット２２で復号することおよび処理することは、送信機ユニット１２で行われた処理およびコーディングと相互補完的に行われる。その後、回復されたデータは、データシンク３０に提供される。
【００２０】
以上記述された信号処理は、音声、ビデオ、パケットデータ、メッセージング、および一方向での他の通信タイプの送信をサポートする。双方向通信システムは双方向データ伝送をサポートする。しかし、別の方向に関する信号処理は単純化のため図１に示されない。通信システム１０は、符号分割多元接続（ＣＤＭＡ）システム、時分割多重アクセス（ＴＤＭＡ）通信システム（例えばＧＳＭシステム）、周波数分割多元接続（ＦＤＭＡ）通信システム、または地球上のリンク上でのユーザー間の音声およびデータ通信をサポートする他の多重アクセス通信システムかもしれない。特定の実施形態では、通信システム１０はＷ－ＣＤＭＡ標準規格に準拠するＣＤＭＡシステムである。

【００２１】
図２は、図１の受信データプロセッサ２８および送信データプロセッサ１６として役立つ可能性があるＤＳＰ ４０アーキテクチャを示す。我々は、ＤＳＰ ４０が、ここで与えられた教えと概念を有効に使用する非常に多数の可能なデジタル信号プロセッサの実施形態のなかで単に一実施形態を表しているに過ぎないことを強調する。従ってＤＳＰ ４０では、スレッドＴ０からＴ５（参照数字４２から５２）は、異なるスレッドからの命令のセットを含んでいる。回路５４は命令アクセス機構を表し、スレッドＴ０からＴ５について命令をフェッチすることに用いられる。回路５４に関する命令は命令キュー５６にキューされる。命令キュー５６での命令は、プロセッサパイプライン６６（以下を参照）へ発行される準備ができている。命令キュー５６から、単一のスレッド（例えばスレッドＴ０）は問題論理回路５８によって選択される可能性がある。選択されたスレッドのレジスタファイル６０は読まれ読まれたデータはＳＬＯＴ０からＳＬＯＴ３について実行データパス６２に送られる。この例においてＳＬＯＴ０からＳＬＯＴ３は、本実施形態において用いられた組合せをグループ化するパケットを提供する。
【００２２】
実行データパス６２からの出力は、ＤＳＰ ４０の動作から結果を返すために、個別のスレッドＴ０からＴ５を適応するように構成されて、レジスタファイル書き込み回路６４に行く。このように、回路５４および前からレジスタファイル書き込み回路６４までのデータ経路は、処理パイプライン６６を形成する。本実施形態は、６つのスレッドＴ０からＴ５まで有するシングルプロセッサを用いて、異種機種環境にある要素プロセッサ（ＨＥＰ）システムのハイブリッドを用いてもよい。プロセッサパイプライン６６は、６つの段階を持っていて、回路５４からレジスタ６０および６４までのデータ項目をフェッチするのに必要な、プロセッササイクルの最小数と一致する。ＤＳＰ ４０は、同時にプロセッサパイプライン６６内の異なるスレッドＴ０からＴ５の命令を実行する。すなわち、ＤＳＰ ４０は、６つの独立したプログラムカウンタ、プロセッサパイプライン６６内のスレッドＴ０からＴ５の命令を区別する内部タギング機構、およびスレッドスイッチをトリガーする機構を提供する。スレッドスイッチオーバーヘッドはゼロからほんの少しのサイクルまで変る。
【００２３】
ＤＳＰ ４０は、従って、種々様々の信号、画像およびビデオの処理アプリケーションに関して、高性能および低電力なように設計された汎用のデジタル信号プロセッサを提供する。図３は、開示された主題の１つの明示に関する関連した命令セットアーキテクチャのいくつかの態様を含むＤＳＰ４０アーキテクチャの概要に提供する。ＤＳＰ ４０アーキテクチャサポートの実装はマルチスレッド（ＩＭＴ）処理をインタリーブした。この実行モデルでは、パイプラインでの異なるスレッドからの命令をインタリーブすることによって、多重ハードウェアスレッドＴ０からＴ５の同時実行をサポートする。この特徴は、常に高いコアおよびメモリ使用を維持している一方、ＤＳＰ ４０が強引なクロック周波数を含むことを許容する。ＩＭＴ処理は、障害実行、広範囲な転送ネットワークなどのような高価な補償機構に関する要求なしで高スループットを提供する。さらに、ＤＳＰ ４０は、Ｍ．Ａｈｍｅｄらによる「変数インタリーブされマルチスレッドされたプロセッサの方法およびシステム」、「マルチスレッドプロセッサにおける可変スレッド割り当ておよびスイッチングのための方法およびシステム」と題された、一般に譲渡された米国特許出願において開示された変形と新規のアプローチのようなＩＭＴ処理の変形を含んでも良い。
【００２４】
図３は、特に、開示された主題の教えを用いてもよい単一のスレッドに適用されるようなＤＳＰ ４０についてコアプロセシングアーキテクチャブロック図７０を提供する。ブロック図７０は、ＡＸＩバス７４からバスインタフェース（Ｉ／Ｆ）７３を介して命令を受信する、共有される命令キャッシュ７２を表す（命令は混合された１６ビットおよび３２ビットの命令を含んでいる）。これらの命令は、スレッドＴ０からＴ５の、監視制御レ

ジスタ８０、シーケンサー７６、およびユーザー制御レジスタ７８に達する。開示された主題のコアレベルシステムアーキテクチャーはまた、インシリコンデバッギングシステム（ＩＳＤＢ）８２を含んでいる。それはＪＴＡＧインターフェース８４を通じてコアプロセッサ７０を接続し、それらの両方はより詳細に以下で説明される。

【００２５】

シーケンサー７６は、Ｓパイプユニット８６、Ｍ－パイプユニット８８、ＬＤ［Ｌｏａｄ］－パイプ９０、およびＬＤ／ＳＴ［Ｓｔｏｒｅ］－パイプユニット９２、（これらのすべては汎用レジスタ９４と通信する）に、ハイブリッドの双方向のスーパースカラー命令および４方向のＶＬＩＭ命令を提供する。ＡＸＩバス７４はまた、スレッドＴ０からＴ５へのＬＤ／ＳＴ命令を、バスＩ／Ｆ ７３を介して共用データキャッシュ９６と通信する。オプションのＬ２キャッシュ／ＴＣＭ ９８信号は、共用データＴＣＭ １００を有するＬＤ／ＳＴ命令を含んでいる。ＬＤ／ＳＴ命令はさらにスレッド汎用レジスタ９４に流れる。ＡＨＢ周辺バス１０２から、ＭＳＭ特定コントローラ１０４はＴ０からＴ５で、割込みコントローラ命令、デバッギング命令、およびタイミング命令を含む割込みを通信する。グローバル制御レジスタ１０６はスレッドＴ０からＴ５で制御レジスタ命令を通信する。

【００２６】

ＤＳＰ ４０は従って、それぞれグローバル制御レジスタ１０６およびプライベート監視制御レジスタ８０を含む６つの仮想ＤＳＰコアを含んでいる。グローバル制御レジスタ１０６はすべてのスレッド間で共有される。スレッドはそれぞれ共通データキャッシュおよび共通の命令キャッシュを共有する。ロード、蓄積およびフェッチ動作は共通バスインタフェースによって提供される。高性能ＡＸＩバス７４およびより低い性能のＡＨＢバス１０２は、オフコアメモリおよび周辺装置にデータおよび命令トラフィックを接続するために用いられる。統合レベル２のメモリ（キャッシュおよび（または）ＴＣＭ）入力９８はオプションである。周辺のアクセスはメモリマップされたロードおよびストアを介してもよい。ＡＨＢとＡＸＩとの間の物理アドレスパーティションはＭＳＭレベルでは構成されてもよい。

【００２７】

明確に、ＤＳＰ ４０に関する与えられたアーキテクチャは時間で発展し変ってもよい。例えば、ＤＳＰ ４０が用いてもよい命令キャッシュの数は、６から１に変化する、または他のキャッシュ数に変化する。スーパースカラーディスパッチ、ＴＣＭ ９８でのＬ１データ、および他のアーキテクチャの態様は変ってもよい。しかし、現在の主題は、種々様々の構成においてかつＤＳＰ ４０の変更の大きなファミリーに関して、引き続き有効であってもよい。

【００２８】

ＩＳＤＢ ８２は、ＪＴＡＧインターフェース８４を介して、ＤＳＰ ４０にハードウェアデバッガを提供する。ＩＳＤＢ ８２は共有方式または監視のみのレジスタによってＪＴＡＧインターフェース８４を介してソフトウェアデバッグ特徴を提供する。これらのレジスタは、すべてのスレッド間のグローバル制御レジスタ１０６と同様にスレッド基底ごとで監視制御レジスタ８０に分割される。システム制御レジスタは、スレッドごとの割込み、および例外処理制御およびスレッドごとの記憶管理動作に使用される。グローバルレジスタは動作をデバッグするためのＩＳＤＢ ８２と相互作用することを許容する。

【００２９】

ＩＳＤＢ ８２によって、ソフトウェア開発者はＤＳＰ ４０が動作している間にソフトウェアをデバッグすることができる。ＩＳＤＢ ８２ハードウェアは、ＩＳＤＢ ８２で動作するソフトウェアデバッガプログラムと結合して、システムソフトウェアを操作するＤＳＰ ４０をデバッグするために用いられてもよい。ＩＳＤＢ ８２はハードウェアスレッドを個々にデバッグすることをサポートする。ユーザーは、スレッド実行を延期してもよく、スレッドレジスタを観察し変更してもよく、命令およびデータメモリを観察し変更してもよく、複数のステップスレッドを選出してもよく、スレッドに命令を詰め込んでもよ

く、スレッド実行を再開してもよい。
【００３０】
信頼されているユーザーはＩＳＤＢ ８２特徴のすべてへのアクセス権を有するが、信頼されていないユーザーは特徴の一部にアクセス権を有している。
【００３１】
ＩＳＤＢ ８２は、プログラムカウンタ（ＰＣ）に存在するソフトウェアをデバッグするＩＳＤＢ ８２と通信するために、ＪＴＡＧインターフェース８４を全く介してだが、デバッガインターフェースカードと接続してもよい。ホストデバッガソフトウェアは、ＩＳＤＢ制御レジスタを読み出しおよび書き込みすることにより、ＩＳＤＢ ８２と相互に作用してもよい。通信は、例えば、３２ビットのデータペイロードと同様に、読み出しまたは書き込みが発生するＩＳＤＢレジスタを識別する４０ビットのパケットを介してであってもよい。この動作をサポートするパケットフォーマットは、各々３２ビット幅である６４までの制御レジスタであってもよい。
【００３２】
図４は、デバッギング機構と開示された主題のコアプロセッサとの間のＩＳＤＢ／ＪＴＡＧインターフェース１１０の重要な態様を示す。ＤＳＰ ４０コアアーキテクチャ７０と共同して、ＩＳＤＢ ８２は、ＩＳＤＢ ＪＴＡＧ回路１１４からＪＴＡＧインターフェース経路１１２を介してＪＴＡＧ ８４と通信する。ＩＳＤＢ ＪＴＡＧ回路１１４は、ＪＴＡＧ ８４とＩＳＤＢ ８２との間のデータの流れを処理する。ＩＳＤＢ ＪＴＡＧ回路１１４はさらにＩＳＤＢ ＪＴＡＧＳｙｎｃ回路１１６を接続する。ＩＳＤＢ ＪＴＡＧＳｙｎｃ回路１１６は、ＩＳＤＢコントローラ１１８、命令ユニット（ＩＵ）１２０および制御ユニット（ＣＵ）１２２とさらに通信する。特に、ＩＳＤＢ ＪＴＡＧＳｙｎｃ回路１１６は、ＩＵ １２０のＩＵ ＩＳＤＢ論理回路とＣＵ１２２のＣＵ ＩＳＤＢコントローラ１２６をインターフェースで接続する。ＣＵ ＩＳＤＢコントローラ１２６は、ＩＳＤＢコントローラ１１８と同様に、ＣＵ ＩＳＤＢ論理回路１２８と通信する。ＩＳＤＢコントローラ１１８からの制御出力は、ＩＳＤＢデータ出力１３０、ＩＳＤＢリセット信号１３２およびＩＳＤＢ割込み１３４を含んでいる。さらにＩＳＤＢコントローラ１１８へのインターフェースは、ＭＣＤインターフェース１３６およびＥＴＭブレークポイントトリガー１３８を含んでいる。
【００３３】
図５は、デバッグプロセスの間にＩＳＤＢ ８２の動作を含むＤＳＰ ４０の、種々のモード制御態様についての処理モード図１４０を示す。図５では、ＤＳＰ ４０はすべてのスレッドに対してグローバルで、個体スレッドに対してローカルな処理モードをサポートする。ＤＳＰ ４０ハードウェアスレッドはそれぞれ、すべて図５に現れているように、個々に２つの実行モード（ユーザモード１４２およびスーパーバイザーモード１４４）、３つの非処理モード（待機モード１４６、オフモード１４８およびデバッグモード１５０）をサポートする。スレッドのモードは他のスレッドに依存せず、例えば、１つのスレッドは待機モード１４６であって別のものがユーザモード１４２であってもよい。
【００３４】
図５のスレッドごとのモード状態図は種々の命令またはイベントによってサポートされる。これらは、「除外」あるいは内部例外イベント、「ｉｎｔ」または外部割込みイベント、「ＲＴＥ」または例外モードからのソフトウェアリターン命令、「ＳＳＲ」またはＳＳＲレジスタ命令へのアップデートを含み、任意のモードから入ってもよい「停止」またはソフトウェア停止命令、同様に任意のモードから入ってもよい「開始」またはソフトウェア開始命令、「ｔｒａｐ」またはソフトウェアトラップ命令、「待機」またはソフトウェア待機命令、「レジューム」またはソフトウェアレジューム命令、「ＤＥ」またはデバッグイベント、および、「ＤＲ」またはデバッグ命令を含む。クレームされた主題の異なる実行における機能はここで与えられたものとわずかに異なっていてもよいが、「開始」「待機」、「再開」、「ＤＥ」および（または）「ＤＲ」の意味は、クレームされた主題の範囲と一致するそれらの最も広い解釈を与えられてもよい。

【００３５】
レジスタは、ユーザモード１４２およびスーパーバイザーモード１４４の両方におけるＤＳＰ ４０において利用可能である。ユーザーモードレジスタは、１セットの汎用レジスタおよび１セットの制御レジスタに分割される。汎用レジスタは、アドレス発生、スカラーおよびベクトル算術を含むすべての多目的計算のために用いられる。制御レジスタは、ハードウェアループ、述語などのような専用機能性をサポートする。多目的レジスタは、３２ビット幅で、単一のレジスタ、または整列したペアの２つのレジスタとしてアクセスされてもよい。汎用レジスタファイルは、ロード／蓄積のためのアドレス、数値命令のためのデータオペランド、およびベクトル命令のためのベクトルオペランドを含む命令に関するオペランドをすべて提供する。

【００３６】
デバッグモード１５０は、スレッドがＩＳＤＢ ８２からのコマンドを待機している特別の状態を提供する。ＩＳＤＢデバッグイベントが発生する場合は常に、ソフトウェアブレークポイント命令、ＩＳＤＢ ８２からのブレークポイントコマンド、またはハードウェアブレークポイントの発生の実行によってのように、指示されたスレッドはデバッグモード１５０に入ってもよい。デバッグモード１５０の場合、コアはＪＴＡＧインターフェース８４からのコマンドを介してＩＳＤＢ ８２によって制御される。ＩＳＤＢ ８２がレジュームコマンドの実行によりスレッドを解放する場合、スレッドはそれらの現在モード設定に従って動作を再開してもよい。スレッドがデバッグモード１５０にある場合、それはＩＳＤＢ ８２によって制御され、他のスレッドによって制御され得ない。デバッグモード１５０においてスレッドをターゲットとする、実行中のスレッドからの待機、再開、開始、あるいは停止命令は無視されてもよい。同様に、マスク不可能割り込み（ＮＭＩ）はデバッグモード１５０におけるスレッドによって無視されてもよい。

【００３７】
ＨＡＲＤＷＡＲＥ ＲＥＳＥＴモード（図５に図示せず）およびデバッグモード１５０は、すべてのスレッドに対してグローバルである。任意のスレッドの処理状態にかかわらず、ハードウェアリセットピンがアサートされる場合は常に、ＤＳＰ ４０はＨＡＲＤＷＡＲＥ ＲＥＳＥＴモードに入ってもよい。ＨＡＲＤＷＡＲＥ ＲＥＳＥＴモードにおいて、レジスタはすべてそれらのリセット値に設定される。ハードウェアリセットピンがデアサートされるまで、処理は発生しなくてもよい。リセットピンがアサートされる場合、プロセッサはリセットモードへ移行し、レジスタはすべてそれらのＨＡＲＤＷＡＲＥ ＲＥＳＥＴ値にリセットされてもよい。リセットピンがデアサートされた後、スレッドＴ０はソフトリセット割込みを与えられてもよい。これはスレッドＴ０にスーパーバイザーモード１４４に入らせ、リセットベクトル位置で実行することを始めさせてもよい。他のすべてのスレッドはオフのままであってもよい。この点では、ソフトウェアは、各スレッドそれぞれに関する制御モード遷移に個々に自由である。

【００３８】
図６から１７は、ＤＳＰ ４０の埋め込まれたトレースマクロセル（ＥＴＭ）ユニットの現在開示された新規かつ有利な特徴に関する。それは、ソフトウェア実行フローに関する詳細情報をリアルタイムで捕獲することによって符号のユーザーデバッギングを向上させる。ＥＴＭは、選択されたＤＳＰ ４０実行を非侵入に監視し記録し、実行情報をパケットへ形成し、パケットストリームをＥＴＢとして知られるオンチップメモリまたはオフチップへ送る。ＥＴＭはまた、関心領域への追跡情報の生成を制限するか集中させるために多くの機構を含んでいる。パケットストリームを用いると、実行の再構成は、ユーザーにコードのランタイム動作の直接の視界を与えて生成されうる。

【００３９】
ＥＴＭは従って、ＤＳＰ ４０および他の同じようなデジタル信号プロセッサに関する包括的なデバッグおよびトレース機能を提供する。これらの機能は、ＤＳＰ ４０がフルスピードで実行するにもかかわらず、プロセッサの性能に負担を加えていない間、特定のイベントの前後でプロセッサの状態についての情報が捕獲されることを許容する。ＥＴＭ

は、精選された追跡情報だけを条件の特殊シーケンスを経た後にだけ単に捕獲するソフトウェアで構成されてもよい。専用で設定可能なトレースポートおよびＦＩＦＯは、圧縮トレースデータが、プロセッサに割り込まず、影響を及ぼすことなく、外部のトレースポート解析器によってチップから読み込まれることを許容する。

【００４０】

トレースポートはコアクロックに独立したトレースクロックと共に１から３２ビットデータバスで構成されうる。例えば、ＥＴＭからのデータレートは、コアクロックの半分でありえ、ピンの数はデータ帯域幅を維持するために増加されうる。同様に、ピンの数は半分にすることができ、データレートは増加されうる。ＥＴＭは、スタンドアロンおよびマルチコア環境内の両方において用いられてもよい。開発者が多重かつ非同期コアからの同時かつ相関したトレースを眺めることを許容する。

【００４１】

図６は、本開示をサポートする様々な全体のＥＴＭ機能を表すブロック図１６０を提供する。ＤＳＰコアプロセッサ７０は、ＥＴＭ１６２をインターフェース接続する。トリガリングフィルタリング回路１６４および圧縮パケット化回路１６６を含んでいる。トリガリングフィルタリング回路１６４および圧縮パケット化回路１６６による処理に続いて、ＥＴＭ出力１６８は、例えば埋め込まれたトレースバッファ（ＥＴＢ）回路またはオフチップ回路であってもよいトレース保存部に流れる。トレース保存部１７０から、ソフトウェア実行記録はデバッグホスト１７３（図６にない）へ出力１７２（図６にない）として流れる。デバッグホスト１７３は、トレース保存部出力１７２を受信し再構成された実行フロー１７６からそこに生成するためのデコンプレッサ成分１７４を含んでいる。ＥＴＭ１６２はＪＴＡＧ ８４から、入力したＪＴＡＧ ８４がデバッグホスト１７３からのデータおよび命令に応じて生成する制御量１７８を受信する。

【００４２】

図６に示すように、ＥＴＭ１６２はＤＳＰ ４０パイプラインを監視する。この情報を用いて、ＥＴＭ１６２は２つの主要な機能（フィルタリング／トリガリング、および圧縮／パケット化）を行う。フィルタリングおよびトリガリングする動作は、ＪＴＡＧインターフェース８４を介してユーザーによってプログラムされ、トレースをオンオフする時を定義するために用いられる。圧縮／パケット化ユニットはＤＳＰ ４０に実行情報を持ち込み、効率的にそれを、トレースポートを介してＥＴＭ１６２から送り出されるパケットにする。ＥＴＭ１６２を出るトレースストリームは、トレース保存部に供給される。トレース保存部は、トレース記録を記録するために大きなメモリキャパシティを提供し、オフチップまたはオンチップのいずれかであってもよい。オンチップ保存部は埋め込まれたトレースバッファ（ＥＴＢ）として知られている。デコンプレッサコンポーネント１７４は、トレース保存部１７０からパケットストリームを取り、プログラムイメージに加えて、ユーザーにＤＳＰパイプライン６６への詳細な視界を与えて、ＤＳＰ ４０の実行フローを再構成するデバッグホスト１７３上で動作するソフトウエアコンポーネントである。

【００４３】

ＥＴＭ１６２は、プロファイリングカウント（キャッシュミス、バンクコンフリクトおよびミクロｔｌｂのミス）を記録し送り出す能力と同様に、６つのスレッドすべてに関するトレース命令順序付けおよびタイミングを提供する。

【００４４】

ＥＴＭ１６２は、ＬＤＳＴデータと同様に、ＰＣとＬＤＳＴのアドレスについてもトリガーしてもよい。ＥＴＭ１６２はシリアルかつ外部のイベント検出をサポートする。さらにＥＴＭ１６２はまた、ＩＳＤＢブレークポイントトリガーイベント、外部トリガーイベントおよびＤＳＰ ４０割込みを生成してもよい。ＥＴＭ１６２はＪＴＡＧ ８４を介してプログラム可能なであり、ある実施形態では５１２ｘ３２ビットの専用のＥＴＢトレース保存部をサポートしてもよい。ＥＴＭ１６２は４つのトリガーブロック（各々２つのアドレスおよび１つのデータコンパレーターを有する）を含み、３状態のシーケンサーを含んでいてもよい。ＥＴＭ１６２トレーシングは、安全なＤＳＰ ４０の制御がレジスタを可能

にする下で、動作してもよく、ＤＳＰ ４０パワーが下落している間の動作に関してプログラムされてもよい。崩壊に動力を供給する。

【００４５】

ＥＴＭ１６２は、ある時のあるウィンドウ上のスレッドに関するプログラムカウンタの十分な進行の記録として命令トレースを生成する。オプションで、プログラムカウンタ進行のタイミング（つまりストールサイクルの識別）も、命令トレースに含まれうる。イベントリソース機構はいつこれらの命令トレースを生成するべきであるかを定義するために用いられる。トリガーとフィルタリング機能はイベントリソースのプログラミングを通じて制御される。より詳細には、イベントリソース制御フィルタリング、トリガリング、およびＩＳＤＢブレークポイント生成においてである。フィルタリングは、いつ命令トレースを可能にし無効にするべきであるかを決定する機能を含んでいる。トリガリングはいつトリガーマーカーをパケットストリームに挿入するかを決定することを含んでいる。ＩＳＤＢブレークポイント判定は、ＩＳＤＢ ８２が動作をデバッグするためのブレークポイントを生成しブレークポイントに応答する条件を指定することを含んでいる。

【００４６】

ＥＴＭ１６２は、ＤＳＰ ４０内の特定の条件がいつ発生するかを検知するために、多くのプライマリイベントリソース（例えばアドレスおよびデータコンパレーター）を含んでいる（例えば、あるＰＣが実行されるかどうか、またはある記憶位置が読み込まれるかどうか）。更に、イベントのより複雑な配置の検出を可能にする第２のイベントリソース（トリガーブロックおよびシーケンサー）がある。

【００４７】

ＥＴＢトレース保存部は、追跡情報がデバイスのピンでトレースポートを介して直ちにエキスポートされることではなくキャプチャーの間に記憶される、オンチップメモリエリアを提供する。その後保存された情報は、一旦キャプチャーが完了したならば、ＥＴＢトレース保存部から縮小されたクロックレートで読取ることができる。これはＪＴＡＧインターフェース８４を介してなされる。この２ステップのプロセスは、多数の高速デバイスピンを用いるワイドのトレースポートに関する必要性を除去する。事実上、「ゼロピン」トレースポートは、デバイスがピンで既にＪＴＡＧポートを持つところで生成される。ＥＴＢトレース保存部１７０は、トレースポート帯域幅限定を超えて、より高い周波数かつ十分な３２ビットのデータポートでデータを読み込み、システムインテグレーターによって供給されたＲＡＭブロックで統合されてもよい。

【００４８】

一実施形態において、ＥＴＢトレース保存部１７０は、５１２のエントリとして２ＫＢのサイズを有していて、それぞれ３２ビット幅である。しかし、ＥＴＢトレース保存部に関する他のサイズは、明らかに開示された主題の範囲内である。ＥＴＢトレース保存部１７０は、１セットのＪＴＡＧアクセス可能なレジスタを介してユーザーをインターフェース接続する。レジスタはそれぞれ、ＪＴＡＧインターフェース８４を介して読み出されまたは書き込まれうる。これらのレジスタは、一旦トレースキャプチャが完了すると、トレースキャプチャーセッションに関するＥＴＢトレース保存部１７０をセットアップし、かつＥＴＢトレース保存部１７０の内容を読出すために用いられる。ＥＴＢトレース保存部１７０は、ＥＴＢトレース保存部１７０の記憶配列の中へインデックスとして読取りポインタを提供する。ＪＴＡＧインターフェース８４を介してＥＴＢトレース保存部１７０の内容を読出す場合、読取りポインタは読む位置を指示する。ＥＴＢトレース保存部１７０はまた、ＥＴＢトレース保存部１７０メモリアレイの中へインデックスとして書き込みポインタを提供する。トレースデータがＥＴＢトレース保存部１７０へ書き込まれる場合、それは書き込みポインタによって指示されたエントリに書かれている。書込み操作の各々は、書込みが発生する後に、次の位置への書き込みポインタを自動インクリメントする。ＥＴＢトレース保存部１７０は、単にＥＴＭトレースストリームの小さなウィンドウを捕えてもよい。ＥＴＢは、データおよびトリガーカウンタを捕獲することがＥＴＢトレース保存部２４０によって捕獲されたトリガー前のデータとトリガー後のデータとの間での分

割を特定するためにいつ使用されるかを判定するために、ＥＴＭからトリガーパケットを捜す。

【００４９】

ＥＴＭ１６２は、従って、ＤＳＰ ４０に関する符号をデバッグする場合にプログラマを援助する。ＥＴＭ１６２は、ある時のあるウィンドウ上のスレッドに関する実行フローの記録である命令トレースを生成する。記録された命令トレースを用いると、プログラマは、それらの符号のランタイム動作についての詳細な眺望を見ることができる。例えば、利用者のプログラムが例外を生成し説明できない場合、ＥＴＭ１６２は例外までの命令のフローを決定する際に援助し、その結果ユーザーが厳密に起こったことにアクセスすることを許容する。ＥＴＭ１６２は、効率的にプログラムフローを表し、トレースデータの生成を最小化するためにパケットに基いた特定のプロトコルを用いる。

【００５０】

ＥＴＭ１６２の一態様は、イベントおよびより複雑なイベント検出シナリオをつなぐためのシーケンサープロセスを含んでいる。シーケンサープロセスおよび関連するトリガーブロック回路１８０の動作を例証するために、図７はシーケンサーフロー図１９０を与える。トリガーブロック回路入力１８２から１８８に応じて、シーケンサープロセス１８０は３状態（Ｓ０からＳ２）での示された例において作動する。シーケンサープロセス３１０の動作については、状態Ｓ０から、プロセスフローはＳ１またはＳ２に行ってもよい。Ｓ１から、順序付けは、Ｓ２に先んじてまたはＳ０に戻るかのどちらかまで進む。Ｓ２から、順序付けは、Ｓ１またはＳ０のいずれかまで進む。

【００５１】

シーケンサープロセス１８０は従って、プログラム可能でありトリガーブロック回路（１８２から１８８）からの一致に基づく、状態間での遷移を有し３つの状態（Ｓ０からＳ２）を含んでいる。シーケンサープロセス１８０は、状態可能が条件として設けられることを各トリガーブロック回路（１８２から１８８）に可能にするためのトレースフィルタリングで使用する。これは、トレーシングがある状態に制限されることを許容する。新規の状態に入る際、各トリガーブロック回路（１８２から１８８）でのカウンタは初期値にリロードされてもよい。ある状態に入る際、ＩＳＤＢブレークポイントはアサートされうる。ある状態に入る際、トリガーマーカーはトレースストリームに挿入されうる。ある状態に入る際、外部トリガー制御もアサートされてもよい。外部トリガーは、シーケンサーがある状態であるといつでもアサートされるままであってもよい。ある状態に入る際、ＤＳＰ ４０への割込みはアサートされてもよい。リセットの後、カウンタはＳ０に初期化される。多重の遷移が同時に発火する場合、シーケンサーは現在の状態のままである。

【００５２】

一実施形態において、ＥＴＭ１６２は、ＤＳＰ ４０性能に関係する種々のイベントを記録することができる６つのカウンタを含んでいる。基本操作はプログラム可能なソースとして各カウンタを利用する。ユーザプログラム可能な領域カウンタは実行をサイクルの固定数のウィンドウに分割する。ウィンドウの期間に、イベントはカウンタへ蓄積される。ウィンドウの端では、カウンタ値はパケットに形成され、トレースポートを介して送られる。その後、カウンタはリセットされ、プロセスはまた始まる。プロファイリングユニットがプログラムフロートレーシングとして同時に動作される場合、これは結果として、性能イベントに関する詳細情報に覆われているプログラムフロートレースをもたらす。更に、プロファイリングユニットは、ユニットがアクティブな場合を制限するために状態可能にするマスクを含んでいる。

【００５３】

領域カウンタは実行をサイクルの固定数のウィンドウに分割するために用いられる。領域のサイズはユーザプログラム可能なレジスタによって決定される。領域カウンタはユーザーに指定された値に初期化され、全てのプロファイリングイベントカウンタがリセットされる。その後、領域カウンタはカウントダウンを始める。領域カウンタがゼロに達する時、プロファイリングイベントカウントの各々に関する値はトレースストリームにおいて

放出される。その後、プロセスはまた始まる。領域カウンタは状態可能が一致するときを単に数える。プロファイリングが不活発な場合、領域カウンタは、その値を維持し、イネーブル状態が再び入れられる場合再び始まる。

【００５４】

プロファイリングカウンタが情報を蓄積してもよい異なるイベントは（ａ）ｄキャッシュミスと、
（ｂ）ｄキャッシュストールサイクルと、
（ｃ）ｉキャッシュミスと、
（ｄ）ｉキャッシュストールサイクルと、
（ｅ）ＩＴＬＢとＤＴＬＢのミスと、
（ｆ）合計のストールサイクルと、を含む。
更に、プロファイリングカウンタはそれぞれ、あるハードウェアスレッドに発生するイベントにカウンタを制限するために、６つのスレッドマスクを含んでいる。

【００５５】

領域カウンタでのように、状態可能なマスクが現状に一致する場合、プロファイリングカウンタは単にアクティブである。すべての他の回の間にカウントはそれらの値を維持し、イネーブル状態が再び入れられる場合カウントすることは再び始まる。

【００５６】

本開示は、インタースレッドトレースアライメントをサポートするために新しいパケットプロトコル機能セットを提供する。すなわち、ＥＴＭ１６２を用いてトレースされているあらゆるスレッドについては、任意の時点では、本開示はトレースされている他のすべてのスレッドについて、その時点でのそれらのスレッドのプログラムカウンタ値および指示状態（例えば、ストール、実行など）を識別することを許容する。このように、ＥＴＭ１６２がサイクルの正確なモードで動作する場合、トレースアライメントはインタースレッドトレースアライメントをサポートするトレースポートプロトコルを利用する。従って、本開示は、インタースレッドアライメントに関する完全に正確なサイクルカウントフィールドと同様に、ｉｓｙｎｃ－ｒｅｓｔａｒｔパケットスレッドに関するサイクルカウントフィールドが類似スレッドになるようにする。この開示配置は、すべてのスレッドに関する同じスレッドサイクルでｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットを生成する。スレッドアライメントが任意の理由で失われる場合、これはさらに再アライメントを可能にさせる。

【００５７】

本開示に一実施形態において、同時に発生する多重の命令シーケンスの実行を提供し、ＤＳＰ ４０は独立して動作する、いくつかの単一にスレッドされたプロセッサとして見られてもよい。ＥＴＭ１６２では、プログラムフローはパケットのシーケンスに分割され、それはどのパケットがどのスレッドに属するかを識別するためにあるパケットに対してスレッド数（ｔｎｕｍ）フィールドを含んでいる。インタースレッドタイミング関係を識別するために、本開示は、異なるスレッド間のタイミング関係を確立し維持する。スレッドがそれぞれ命令追跡を独立して可能にしたり無効にしてもよいので、スレッドが命令追跡をオンする場合、他のスレッドは暫くの間可能にしたそれらのトレーシングを可能にしてもよい。現在開示された方法およびシステムは、スレッドがトレーシングをオンする場合と、他のスレッドがトレーシングをオンし続いてオンする場合との間のオフセットをマーキングすることによって、スレッド実行順序の相対的なタイミングを確立する。

【００５８】

従って本開示は、最後のスレッドがトレーシングをオンしてからのサイクル数を指示するためのスレッドサイクルオフセットフィールドを含んでいる。また、他のスレッドがアクティブでない場合、サイクルオフセットフィールドは、最近のトレースがすべてのスレッド間でオフしてからのサイクル数を含んでいる。トレースセッションの後、パケットはスレッド実行を再構成することを可能にさせる。その後、オフセットフィールドを用いて、実行順序は適切にスレッド中に整列してもよい。更に、命令アライメント機構はデータ

損失の場合には実行トレーシングを再確立することを許容する。

【００５９】

開示された主題はインタースレッドタイミング関係を再確立することを更に許容する。同期パケットを周期的に生成することを許容するグローバルなカウンタを維持することによって、カウンタがゼロに達する場合、パケットは各スレッドについて生成されてもよい。上記のパケットは、そのスレッドに関する現在のプログラムカウンタ値を含んでいる。さて、すべてのスレッドに関する同時のパケット生成に関するＤＳＰ ４０に存在してもよい種々の限定のために、本開示は同期パケットへのサイクルオフセットフィールドを含んでいる。最後のスレッド（他のスレッド間の）が同期パケットを生成してから、サイクルオフセットフィールドはサイクル数を指示する。サイクルオフセットフィールドは制限のあるサイズであり、かつカウントが飽和すれば、同期は実現されることができず、カウンタを待ってもよい。以下により完全に記述されるこれらの機構を用いて、本開示は１つのスレッドに関する任意のある点でインタースレッド実行タイミング関係を維持することを許容する。結果は、デバッギングおよび他の重要な目的のためのすべてのスレッドの動作および状態を観測する能力である。

【００６０】

この理解では、図８は、現在開示されたインタースレッドトレースアライメントプロセスの重要な概念を示す。図８を参照すると、タイムライン２０２が示すように、インタースレッドアライメントプロセス２００は時間ｔ＝０に基づいている。ＤＳＰ４０上で動作し得る種々のスレッド２０４、２０６、２０８、２１０および２１２については、スレッド開始が異なる時点で発生してもよい。このように、時間ｔ＝ｔ０では、スレッド２０４は動作を開始してもよい。その後遅い時間ｔ＝ｔ１で、スレッド２０６は動作を開始してもよい。時間ｔ＝ｔ２で、スレッド２０８は始まり、ｔ＝ｔ３でスレッド２１０は動作を始め、ｔ＝ｔ４でスレッド２１２は動作を開始する。より完全に以下で説明されるように、継続期間２１４は、トレーシングがスレッド２１２に関してアクティブになるまで、サイクル数を表す。継続期間２１４は、ｉｓｙｎｃ－ｒｅｓｔａｒｔパケットからトレース開始プログラムカウンタ（ＰＣ）に関するサイクル数を表す。すべての相互関連するスレッド（２０４から２１２）については、ｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットはすべてのスレッドについて同時に生成される。更に、継続期間２２０は、例えばスレッド２１２で、発生している長いストールの状況を扱っている。

【００６１】

より完全に本開示のインタースレッド動作を記述するために、図９は、パケット生成ユニット２３０の一実施形態の機能概要を示す。パケット生成ユニット２３０は、パケット生成制御回路２３２を含んでいる。それは、ＦＩＦＯ入力ｍｕｘ２３４へ、ｇｅｎＩｓｙｎｃＰ、ｇｅｎＡｓｙｎｃ、ｇｅｎＢｒａｎｃｈ、ｇｅｎＰｈｅａｄｅｒ、ｇｅｎＩｓｙｎｃＲ、ｇｅｎＰｒｏｆおよびｇｅｎＣＣｏｕｎｔの出力を生成するためのアトムW、アトムNおよびアトムＥの入力を受信した。Ｐｈｅａｄｅｒ生成回路２３６は、パケット生成制御回路２３２およびアトムＥ入力（ＦＩＦＯ入力ｍｕｘ２３４へのｐｈｅａｄｅｒＰＫ出力を生成するためのすべて）からｗカウント、ｅカウントおよびｎカウント入力を受信する。ＰＣ／ｔｌｄ／ａｓｉｄ／ｔｎｕｍ入力は、ｉｓｙｎｃ－ｒｅｓｔａｒｔ／ｃｙｃｌｅ－ｃｏｕｎｔ生成機能２３８、ｉｓｙｎｃ－ｐｅｒｉｏｄｉｃ生成機能２４０およびｂｒａｎｃｈ－ａｄｄｒｅｓｓ生成機能２４２に流れる。Ｉｓｙｎｃ－ｒｅｓｔａｒｔ／ｃｙｃｌｅ－ｃｏｕｎｔ生成機能２３８は、ＦＩＦＯ入力ｍｕｘ２３４への、ｉｓｙｎｃＲＰＫ、ｉｓｙｎｃＲＬｅｎ、ＣＣＰＫおよびＣＣＬｅｎ出力を提供する。Ｉｓｙｎｃ－ｐｅｒｉｏｄｉｃ生成機能２４０は、ＦＩＦＯ入力ｍｕｘ２３４へｉｓｙｎｃＰＰＫとｉｓｙｎｃＰＬｅｎの入力を提供する。ｂｒａｎｃｈ－ａｄｄｒｅｓｓ生成機能２４２（また、それらは分岐型入力を受信する）は、ＦＩＦＯ入力ｍｕｘ２３４へｂｒａｎｃｈＰＫとｂｒａｎｃｈＬｅｎの出力を提供する。更に、プロファイル生成機能２４４は、ＦＩＦＯ入力ｍｕｘ２３４へのｐｒｏｆＰＫとｐｒｏｆＬｅｎの出力を生成するためのプロファイル識別子入力を受信する。

【００６２】
ＦＩＦＯ入力ｍｕｘ２３４は、所与のサイクルで生成される必要のあるすべてのパケットをとり、連続的なチャンクをそれらから作る。ＦＩＦＯ入力ｍｕｘ２３４は、その連結を行うときに可変サイズであるパケットを説明しなければならない。ＦＩＦＯ入力ｍｕｘ２３４の出力はＦＩＦＯに送り出される前に登録される。
【００６３】
本開示パケット生成およびＦＩＦＯ入力ｍｕｘ２３４に関しては、動作が３つの段階にわたって発生する。パケット生成の第１の段階では、パケット生成制御機能２３２および個別のパケット生成エンジン２３６から２４４は、アトムカウンタであるｗ－カウント、ｅ－カウントおよびｎ－カウントインクリメントと同様に、作動する。ＦＩＦＯ入力ｍｕｘ２３４の動作は、この第１の段階における連続ブロックへのサイクルで生成されたパケットをすべて合併することを含んでいる。第２の段階では、ＦＩＦＯ書込みは、ＦＩＦＯ書き込みポインタに整列させるべきデータを循環させること、書込み可能性を計算して循環させること、および、データをＦＩＦＯｍｕｘ２３４レジスタに書き込むことを含むために発生する。ＦＩＦＯ読取り段階である第３の段階では、データはＦＩＦＯレジスタから読まれてもよい。この第３の段階では、またトリガーパケットを挿入し、ＥＴＢにデータを送る特別な場合が発生する。
【００６４】
図９において見られるように、個別のパケット生成ユニットの各々はそのそれぞれのデータをパケットにし、次に、連結について結果として生じるパケットおよび長さをＦＩＦＯ入力ｍｕｘ２３４へ送る。更に、パケット生成制御回路２３２はアトムカウンタを維持し、特定のサイクルでどのパケットを生成するかを決定する。パケット生成ユニット２３０は、今後のｐｈｅａｄｅｒまたはサイクルカウントパケットで送り出されてもよい未決のアトムの数を記録するために３つのカウンタを保持している。Ｅ－アトムに遭遇する場合は常に、Ｅ－アトムカウンタはインクリメントする。Ｎ－アトムに遭遇し、カウントが現在のアトムを含んでいる場合は常に、Ｎ－アトムカウンタはインクリメントする。すなわち、現在のアトムがＮならば、それはＮ－アトムカウントに含まれる。Ｗ－アトムに遭遇する場合は常に、Ｗ－アトムカウンタはインクリメントする。これらのカウンタはスレッドごとのカウンタであり、また、そういうものとして、本実施形態では各々の６つのコピーがある。カウンタがｐｈｅａｄｅｒまたはサイクルカウントパケットを介して送り出される場合は常に、カウンタはリセットされる。
【００６５】
ターゲットが前のループバックとは異なる場合、ｂｒａｎｃｈ－ａｄｄｒｅｓｓ生成機能２４２から、ｂｒａｎｃｈ－ａｄｄｒｅｓｓパケットは間接のブランチループバックのターゲット用に生成されてもよい。全ループバックモードが設定される場合、パケットはすべてのループバックについて送り出される。また、ｉｓｙｎｃ－ｒｅｓｔａｒｔおよびｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットが生成された後、新規のループバックパケットは追い出される。その後ＰＣ相対ブランチＩＦ直接のブランチモードのターゲットはセットされる。イベントのターゲット（割込み、例外、ＳＷＩ、ＴＲＡＰなど）はＲＴＥ命令を返す。トレース可能性がアサートされ、かつそれがトレーシングの第１のサイクルでない場合、ｂｒａｎｃｈ－ａｄｄｒｅｓｓパケットは生成されてもよい。パケット生成制御ユニットは、各々スレッドがループバックｂｒａｎｃｈ－ａｄｄｒｅｓｓパケットが必要とされるかどうかを判定するために、前のループバックのターゲットレジスタを維持する。
【００６６】
図１０は、現在開示された主題の命令に適用可能な種々のアトムに関する定義のテーブル２５０を示す。パケットプロトコルの主な目的はインタースレッドトレースアライメントをサポートすることである。すなわち、トレースされているあらゆるスレッドについては、任意の時点では、ユーザーは、その時点でのこれらのスレッドのプログラムカウンタ値および指示状態（ストール、実行）を、トレースされている他のすべてのスレッドについて識別することができるべきである。ＥＴＭ１６２がサイクルの正確なモードで動作す

る場合、トレースアライメントは役に立つ。トレースポートプロトコルはインタースレッドトレースアライメントをサポートするために拡張されている。「Ｓ」型命令アトムは、デュアルジャンプとして知られているブランチ機構を提供する。ここで、ＥＴＭ１６２はＳ型を含んでいる。それはアトムを保持するためにｐｈｅａｄｅｒ－ｆｏｒｍａｔ４パケット型を用いて定義される。

【００６７】

図１１は、３２ビットのＴＩＤフィールドをサポートする本開示に関するブランチアドレスパケット２８０を提供する。現在開示されたｂｒａｎｃｈ－ａｄｄｒｅｓｓパケットは、図１１に現れる内容である３２ビットのＴＩＤフィールドをサポートする。ｂｒａｎｃｈ－ａｄｄｒｅｓｓパケットは１と１１バイトとの間を含む可変長である。バイト０から４のＭＳＢでの連続ビットは、パケットがその点を越えて連続するかどうかを示す。別々に、Ｔ－ビットは、タイプフィールドがパケットの終わりに付加されるかどうかを示す。バイト４のＭＳＢにおける１は、５バイトが続くということを示す（４バイトのＴＩＤおよび１バイトのＡＳＩＤ）。

【００６８】

ｂｒａｎｃｈ－ａｄｄｒｅｓｓパケットはＰＣ、ＴＩＤ／ＡＳＩＤおよび型圧縮に可変である。ＴＩＤ／ＡＳＩＤは、どちらか一つがｔｎｕｍに関して最後のｂｒａｎｃｈ－ａｄｄｒｅｓｓパケットまたはｉｓｙｎｃ（ｒｅｓｔａｒｔ ｏｒ ｐｅｒｉｏｄｉｃ）以来変っている場合、送り出されてもよい。型は、最後の型が送り出されてからそれが変っている場合、送り出されてもよい。それはｉｓｙｎｃ－ｒｅｓｔａｒｔまたはｉｓｙｎｃ－ｐｅｒｉｏｄｉｃの後に最初のｂｒａｎｃｈ－ａｄｄｒｅｓｓパケットについて生成されてもよい。ＰＣは、ｂｒａｎｃｈ－ａｄｄｒｅｓｓ、ｉｓｙｎｃ－ｒｅｓｔａｒｔ、またはｉｓｙｎｃ－ｐｅｒｉｏｄｉｃによって送られた前のＰＣと比較して圧縮される。変化したＰＣのより低い部分が送り出されてもよい。各スレッドについては、前のＰＣ、前のＴＩＤ／ＡＳＩＤ、および前の型レジスタは適切な圧縮を決定するために用いられる。

【００６９】

図１２は、現在開示されたプロセスで使用されるためのｉｓｙｎｃ－ｒｅｓｔａｒｔパケット２６０の典型的な内容を提示する。ｉｓｙｎｃ－ｒｅｓｔａｒｔパケットは、トレーシングが開始される場合、生成されてもよい（トレース可能性は以前には低かった）。更に、オーバフロー条件から出てくるときで、トレース可能性がまだ高い場合には、ｉｓｙｎｃ－ｒｅｓｔａｒｔパケットはオーバフローがあったことを示す理由フィールドに生成される。本開示は、ｉｓｙｎｃ－ｒｅｓｔａｒｔパケットを用いて、アライメントを提供する。命令アライメントは、トレースされているすべてのスレッドのすべてのサイクルに、グローバルスレッドサイクルを割り当てることとしてみなせうる。デコンプレッサ手順は、ｉｓｙｎｃ－ｒｅｓｔａｒｔパケットがトレースストリームにおいて利用可能な場合についてこれを遂行する。デコンプレッサは、最後のｉｓｙｎｃ－ｒｅｓｔａｒｔカウントのカウンタ値を維持する。

【００７０】

ｉｓｙｎｃ－ｒｅｓｔａｒｔパケットが遭遇するときはいつでも、本開示はサイクルカウントフィールドおよびサイクルカウントタイプフィールドを検査してもよい。タイプフィールドが「グローバルな」場合、プロセスはサイクルカウントフィールド値に最後のｉｓｙｎｃ－ｒｅｓｔａｒｔカウントを割り当てて、最後のｉｓｙｎｃ－ｒｅｓｔａｒｔカウントを有するそのパケットに注釈を付ける。タイプフィールドが「オフセット」である場合、プロセスはサイクルカウントフィールドでの値によって最後のｉｓｙｎｃ－ｒｅｓｔａｒｔカウントをインクリメントし、さらに、最後のｉｓｙｎｃ－ｒｅｓｔａｒｔカウントによりパケットに注釈を付けてもよい。その点で、ｉｓｙｎｃ－ｒｅｓｔａｒｔパケットはそれぞれグローバルスレッドサイクル値に注釈を付けられる。

【００７１】

次のステップはアトムを注釈することである。そのプロセスは、トレースストリームを、パケットストリームにおけるｔｎｕｍ値に基いて個別のスレッドローカルのパケットス

トリームへ分割する。個々のスレッドローカルのパケットストリームについては、プロセスは可変なグローバルスレッドサイクルを維持する。この値は、いつでもｉｓｙｎｃ－ｒｅｓｔａｒｔパケットに遭遇され、グローバルスレッドサイクルは注釈を付けられた値に割当てられるように維持される。命令アトムに遭遇する（Ｅ、Ｎ、ＳまたはW）ごとに、そのプロセスは、１ずつグローバルスレッドサイクルおよびインクリメントグローバルスレッドサイクルを有するそのアトムに注釈を付けてもよい。通常のデコンプレッション手順を用いて、プロセスは、各アトムに関するプログラムカウンタ値を決定する。この手順が適用された後、トレースされているすべてのスレッドのすべてのサイクルはグローバルスレッドサイクルカウント値に注釈を付けられてもよい。結果として、例えばあるスレッドサイクルに関して、６つのスレッドすべてに関する現在のプログラムカウンタ値は、それらのスレッドに関するパイプラインの状態と同様に決定されてもよい（ストール、実行など）。

【００７２】

非サイクル正確なバージョンについては、ＴＩＤフィールドは３２ビットまで拡大される。サイクル正確なバージョンについては、ＴＩＤは３２ビットまで拡大され、新規のサイクルカウントフィールドはトレースアライメントをサポートするために定義される。グローバルなサイクルカウントは、最後のハードウェアリセットからスレッドサイクル数を指示する。最後のｉｓｙｎｃ－ｒｅｓｔａｒｔパケットが生成されてから、オフセットサイクルカウントはサイクル数を指示する。

【００７３】

図１３は、現在開示された主題をサポートするためのｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケット２９０の実施形態を表す。Ｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットは、（すべてのスレッドの中から）最後のｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットが生成されてから、スレッドサイクル数を指示する「オフセット」フィールドを含んでいる。開示されたプロセスはまたｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットを用いて、アライメントをサポートする。場合によっては、トレースデータの制限されたウィンドウだけが利用可能である。従って、ｉｓｙｎｃ－ｒｅｓｔａｒｔパケットは利用可能ではないかもしれない。これらの場合では、ｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットはインタースレッドアライメントを決定するために用いられてもよい。パケット境界が未知の場合、プロセスはパケットアライメントを回復するためにａｓｙｎｃパケットを探索してもよい。デコンプレッサはカウンタｌａｓｔ－ｉｓｙｎｃ－ｐｅｒｉｏｄｉｃ－ｃｏｕｎｔを維持してもよい。マルチスレッドパケットストリームを通り抜けると、プロセスはオフセットフィールドがない最も古いｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットを探索することを含んでもよい。その後、プロセスはｌａｓｔ－ｉｓｙｎｃ－ｐｅｒｉｏｄｉｃ－ｃｏｕｎｔを０に割り当て、値によりそのパケットを注釈する。他方のｔｎｕｍｓからの次のｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットの各々については、プロセスは、オフセットフィールド値、およびｉｓｙｎｃ－ｐｅｒｉｏｄｉｃ－ｃｏｕｎｔをもつｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットの各々によってｉｓｙｎｃ－ｐｅｒｉｏｄｉｃ－ｃｏｕｎｔをインクリメントしてもよい。その後、プロセスはスレッドを個別のパケットストリームへ分割し、それらを別々にデコンプレスする。

【００７４】

ＥＴＭｉｓｙｎｃ－ｐｅｒｉｏｄｉｃカウンタは、パケットバイトがＦＩＦＯに送られる場合は常に、ディクリメントされてもよい。カウンタがスレッドに関してゼロに達する場合、ｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットが未決としてマークされてもよい。次の機会では、ｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットは生成される。１つのｉｓｙｎｃ－ｐｅｒｉｏｄｉｃは、次のパケットのうちのどれかが生成されている場合（ｉｓｙｎｃ－ｒｅｓｔａｒｔ、トリガー、ｂｒａｎｃｈ－ａｄｄｒｅｓｓ、ａｓｙｎｃ、プロファイル）、阻止されてもよい。スレッドはそれぞれそれ自身のｉｓｙｎｃ－ｐｅｒｉｏｄｉｃカウンタを維持してもよい。ｉｓｙｎｃカウンタは、いずれかｉｓｙｎｃ－ｒｅｓｔａｒｔパケットがそのスレッドについて生成される場合は常に、リセットされてもよい。

【００７５】
図１４は、現在開示されたプロセスで使用される１セットのサイクルカウントパケットを与える。サイクルカウントパケットはＷ－アトムを蓄積するために用いられてもよい。また、サイクルカウントパケットは単にサイクル正確なモードにおいて生成されてもよい。サイクルカウントパケットは次の場合に生成されてもよい。ＷからＥまたはＷからＮへ切り替える場合、Ｗ－アトムカウントがフォーマット３ｐｈｅａｄｅｒキャパシティに入らない場合、未決のＷ－アトムをもつアトムブレークの場合、およびＷ－アトムが多すぎてフォーマット３ｐｈｅａｄｅｒに入らない場合。Ｗ－アトムはサイクルカウントパケットに入れられてもよく、また、およびＥまたはＮアトムはフォーマット１ｐｈｅａｄｅｒで出てきてもよいし、サイクルカウントパケットはその完全に正確な最大に達してもよい。サイクルカウントパケットが、それがその最大３２ビット値に達するまでインクリメントすることを許容する代りに、ＥＴＭは十分な正確さと制限のある正確さとの間の限界で多重のより小さなサイクルカウント値を送り出す。この場合、現在のアトムはＷ－アトムでもよい。このアトムがパケットに含まれていないので、カウントは０の代りに１にリセットされる。
【００７６】
ｉｓｙｎｃ－ｒｅｓｔａｒｔパケットでのサイクルカウントは２つの可能な型として再定義されている。最後のハードウェアリセットから参照される６４ビット「グローバル」カウント、および（すべてのスレッドの中から）最後のｉｓｙｎｃ－ｒｅｓｔａｒｔパケットからのスレッドサイクル数を指示する１６ビットオフセットカウント。
【００７７】
２バイト３００および３バイト３１０のサイクルカウントパケットに加えて、本開示は、１６ビットのペイロードに３２ビットの値を入れるために３バイトパケット３２０を提供する。現在の主題では、１バイトのヘッダーおよび４バイトのペイロードを含む５バイトのパケットは、図１４に示されるように定義される。各スレッドについては、開示されたプロセスはカウントグローバルスレッドサイクルを維持する。遭遇する最初のｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットについては、プロセスはグローバルスレッドサイクルをそのパケットの注釈を付けられた値に設定し、その値により前のアトムに注釈を付ける。その後、グローバルスレッドサイクルは１だけインクリメントされる。アトム（Ｅ、Ｎ、Ｓ、Ｗ）に遭遇するといつでも、プロセスはそのアトムにグローバルスレッドサイクルによって注釈を付け、グローバルスレッドカウントを１だけインクリメントする。プロセスは、各アトムにプログラムカウンタ値を割当てるために通常として、ストリームをデコンプレスしてもよい。この点では以前のように、トレースされているすべてのスレッドのすべてのサイクルはそれに関連したグローバルスレッドサイクルを有する。
【００７８】
図１５は、本開示に適切なものとしてｐｈｅａｄｅｒ　ｆｏｒｍａｔ４パケット３３０の一実施形態を示す。ｐｈｅａｄｅｒパケットは異なる場合に生成されてもよい。例えば、パケットの中にこれ以上のアトムを保持することができない場合、およびｂｒａｎｃｈ－ａｄｄｒｅｓｓ、プロファイルまたはｉｓｙｎｃ－ｐｅｒｉｏｄｉｃパケットが生成される場合、ｐｈｅａｄｅｒはアトムストリーム内のそれぞれのパケットの位置をマークするために生成される。フォーマット１ｐｈｅａｄｅｒパケットは３つのアトムペイロードフィールドがあってもよい。０から１のＥアトム（フィールド２）が後続する０から３のＮアトム（フィールド１）が後続する０から３１のＥのアトム（フィールド０）。フォーマット１ｐｈｅａｄｅｒパケットを生成するための規則は以下であってもよい。最高でのアトムＥカウントおよび現在のアトムがＥである。カウントされたＥは、フィールド０に入り、現在のＥアトムはフィールド２に入れられてもよい。最高でのアトムＮカウント（カウントは現在のＮアトムを含んでいる）。カウントされたＥは、フィールド０に入れられ、Ｎアトムカウントはフィールド１に入れられてもよい。現在のアトムはＥであり、Ｎアトムカウントは０でない。カウントされたＥおよびＮアトムカウントは、フィールド０およびフィールド１に配置されてもよく、現在のＥアトムはフィールド２に配置されても

よい。現在のアトムがＷであり、ＥまたはＮアトムが未決の場合、現在のＥおよびＮカウントはペイロードフィールド０およびフィールド１で送り出されてもよい。アトムブレークおよび未決のＷ－アトムがない場合、現在のＥおよびＮカウントはフィールド０およびフィールド１に配置されてもよく、また、現在のアトムがＥである場合、それはフィールド２に配置されてもよい。

【００７９】

図１６は、開示された主題の実施形態での使用のためのａｓｙｎｃパケット３４０を表す。ａｓｙｎｃパケット長はより長いサイクルカウントフィールドを調整し、例えば１０バイトを含んでいてもよい。ａｓｙｎｃ－ｐｅｒｉｏｄｉｃカウンタもａｓｙｎｃパケットがいつ生成されるかを定義するために用いられてもよい。ａｓｙｎｃパケットは、ＥＴＭの周期的なカウンタが達する場合は常に、未決のものとしてマークされる。ａｓｙｎｃパケットは単独で現れなければならないし、他のパケットが生成される場合は常に未決のままである。ａｓｙｎｃカウンタはグローバルでかつ、パケットバイトがＦＩＦＯに送られる場合は常に、ディクリメントされてもよい。

【００８０】

要約すると、開示された主題は、実行トレース処理での使用についてマルチスレッドプロセッサを有するインタースレッドトレースアライメントのための方法およびシステムを提供する。従って開示された主題は、共通所定イベントに関するタイミングデータを記録することを含んでいる。上記のイベントは、実行トレーシングを開始された最後のスレッドからのサイクル数かもしれないし、実行トレーシングを終了された全てのスレッドからのサイクル数かもしれない。スレッドが実行トレーシングを開始するサイクル数は、実行トレーシングのタイミングを維持するための共通所定イベントに参照される。共通所定イベントに関するデータはその後、スレッドが実行トレーシングを開始した時と関連づけるために更新されてもよい。結果はすべてのスレッドに関連したタイミングデータを整列させることを可能にする。相互関係があった記録は、すべての動作するスレッドに関してタイミングデータを同期させると同様に、マルチスレッドプロセッサにおいて動作するスレッドに関する相互依存の実行トレーシング情報を再構成することを可能にする。

【００８１】

上記でわかるように、インタリーブされたマルチスレッドプロセッサを含むマルチスレッドデジタル信号プロセッサでマルチスレッドプロセッサによるインタースレッドトレースアライメントについて本明細書で記述された処理特徴および機能は、種々の手法で実装されてもよい。例えば、ＤＳＰ ４０は上記動作を行うだけでなく、本実施形態は特定用途向け集積回路（ＡＳＩＣ）、マイクロコントローラ、デジタル信号プロセッサまたはここに記述された機能を行うように設計された他の電子回路において実装されてもよい。さらに、ここで記述されたプロセスと特徴は、上記の種々の信号および命令処理システムによる読み出しおよび実行に関して、磁気記録媒体、光学の記録媒体、または他の記録媒体に記憶されてもよい。従って、好ましい実施形態の先の詳細な説明は、いずれか当業者がクレームされた主題を作るか用いることを可能にするために提供される。これらの実施形態への種々の変更は当業者に即座に明白になり、本明細書で定義された総括的な原理は、革新的な能力の使用のない他の実施形態に適用されてもよい。このように、クレームされた主題は、本明細書で示された実施形態に限定的のようには意図されず、本明細書で開示された原理と新規な特徴に一致する最も広い範囲を与えられるべきである。

【図１】

図１

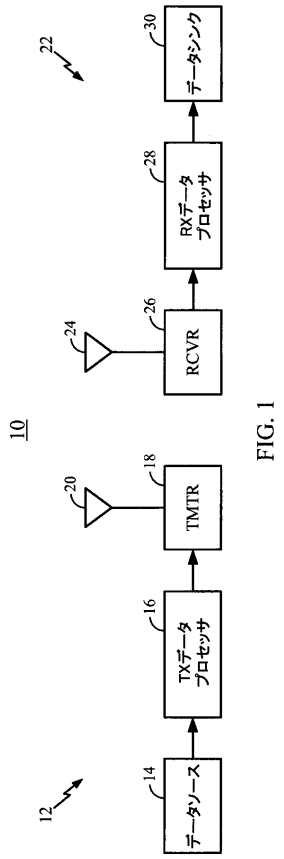


FIG. 1

【図２】

図２

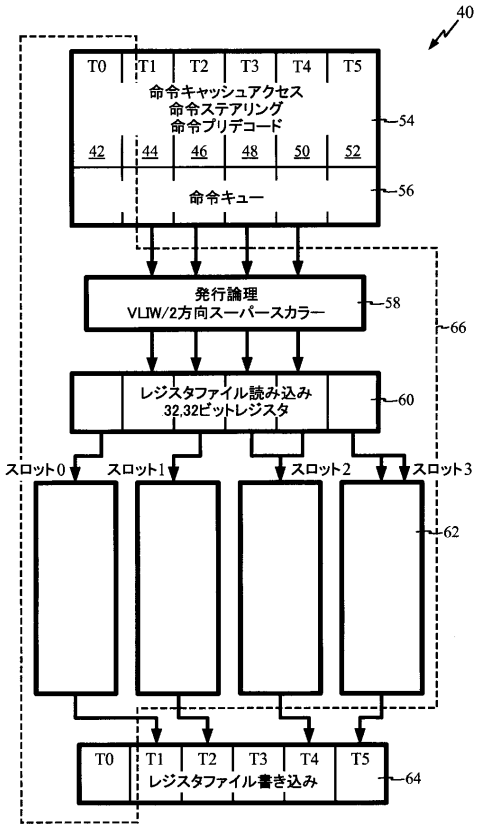


FIG. 2

【図３】

図３

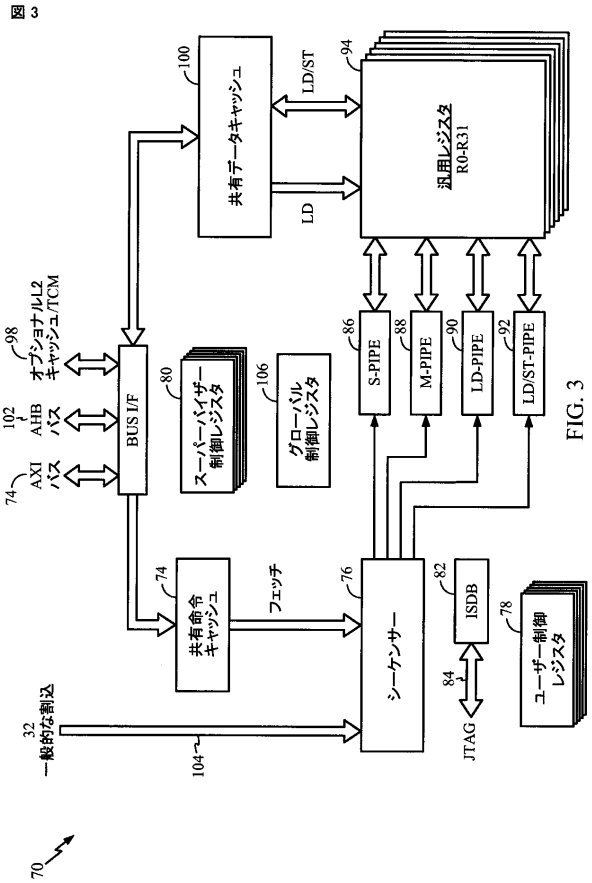


FIG. 3

【図４】

図４

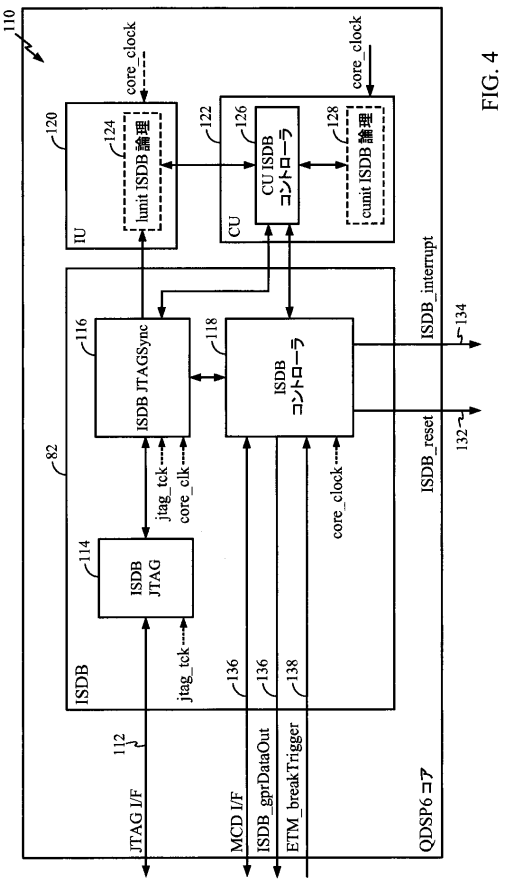


FIG. 4

【図５】

図５

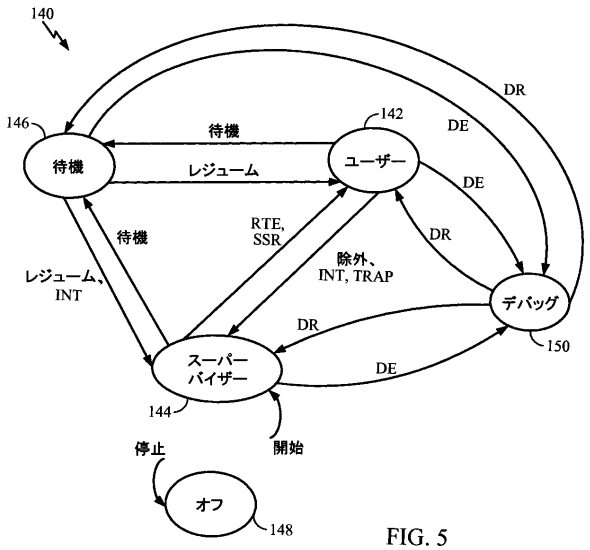


FIG. 5

【図６】

図６

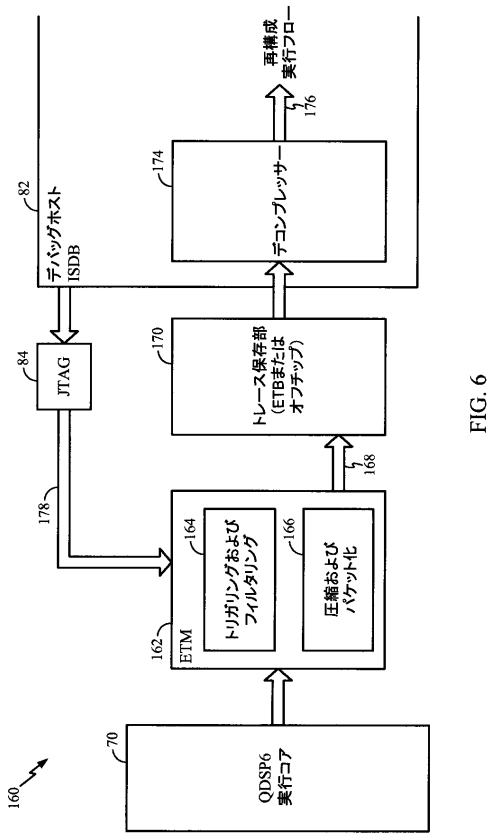


FIG. 6

【図７】

図７

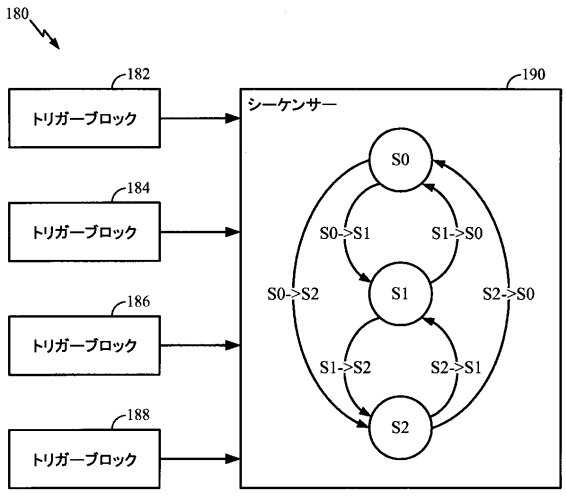


FIG. 7

【図８】

図８

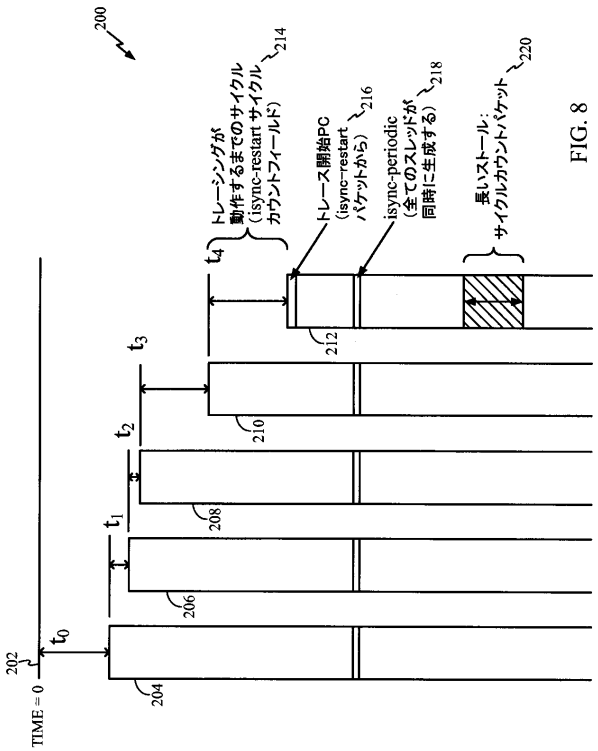


FIG. 8

【図９】

図 9

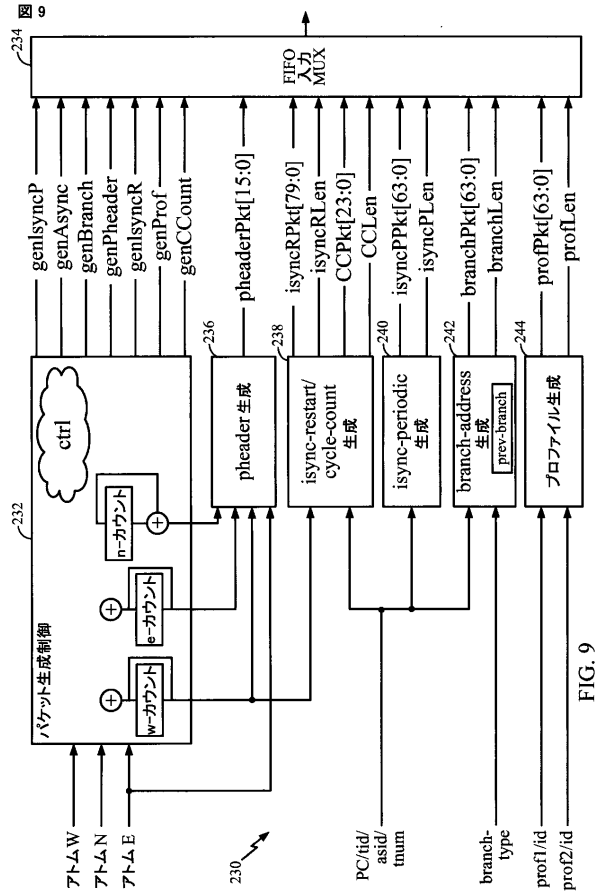


FIG. 9

【図１０】

図 10

| アトム | 定義（デュアルジャンプでない場合） | 定義（デュアルジャンプである場合） |
|---|---|---|
| W | ストールサイクル | ストールサイクル |
| E | 命令が実行。<br>条件ジャンプが採用 | 第1ジャンプが採用 |
| N | 条件ジャンプが誤り<br>ループエンド中止 | 両方のジャンプが不採用 |
| S | | 第2ジャンプが採用 |

250

FIG. 10

【図１１】

図 11

| | | | | |
|---|---|---|---|---|
| バイト0 | 1 | PC[5:1] | T | 1 |
| バイト1 | 1 | PC[12:6] | | |
| バイト2 | 1 | PC[19:13] | | |
| バイト3 | 1 | PC[26:20] | | |
| バイト4 | 1 | 0 0 | PC[31:27] | |
| バイト5 | TID[7:0] | | | |
| バイト6 | TID[15:8] | | | |
| バイト7 | TID[23:16] | | | |
| バイト8 | TID[31:24] | | | |
| バイト9 | 1 0 | ASID[5:0] | | |
| バイト10 | TYPE[7:0] | | | |

280

T：タイプフィールド存在フラグ

FIG. 11

【図１２】

図 12

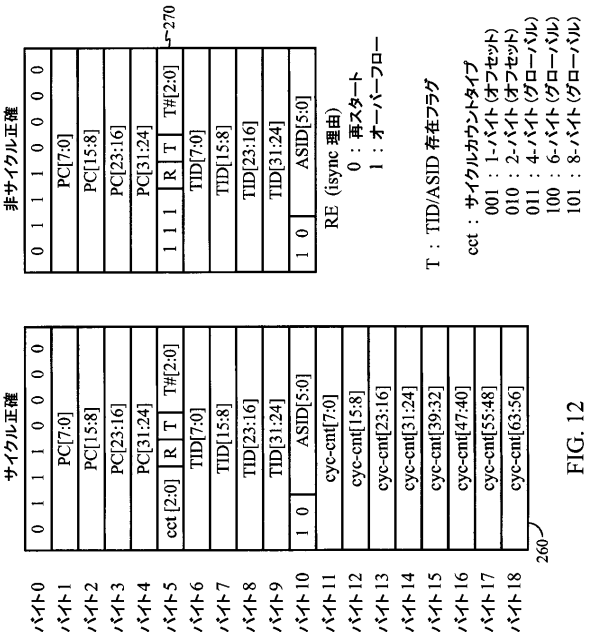


FIG. 12

【図１３】

図 13

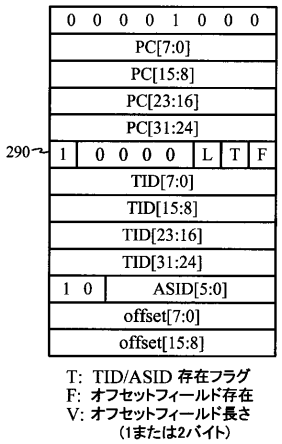


FIG. 13

【図１４】

図 14

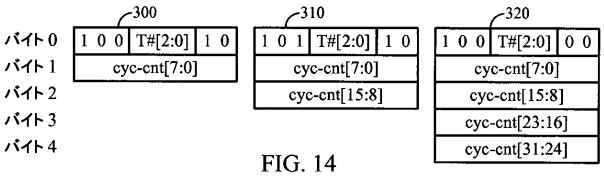


FIG. 14

【図１５】

図 15

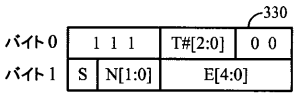


FIG. 15

【図１６】

図 16

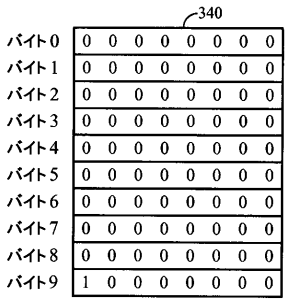


FIG. 16

【図１７】

図 17

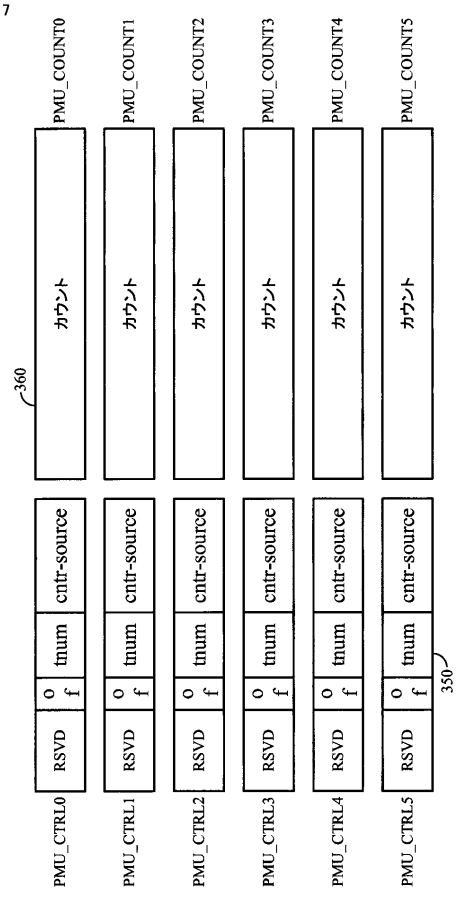


FIG. 17

【国際調査報告】

61000170004

# INTERNATIONAL SEARCH REPORT

International application No
PCT/US2008/060117

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F11/36

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | WO 2006/030195 A (IGNIOS LTD [GB]; LIPPETT MARK DAVID [GB]; OUNG AYEWIN [GB] COWARE INC) 23 March 2006 (2006-03-23)<br>abstract<br>page 24, line 10 - page 27, line 20<br>claims 1,12<br>-/— | 1,11,21, 31 |

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 21 January 2010 | 02/02/2010 |

| Name and mailing address of the ISA/ | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2<br>NL – 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040,<br>Fax: (+31-70) 340-3016 | Renault, Sophie |

Form PCT/ISA/210 (second sheet) (April 2005)

12.4.2010

2

## INTERNATIONAL SEARCH REPORT

International application No
PCT/US2008/060117

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | YANG QIAN ET AL: "Cycle accurate thread timer for linux environment" PERFORMANCE ANALYSIS OF SYSTEMS AND SOFTWARE, 2001. ISPASS. 2001 IEEE INTERNATIONAL SYMPOSIUM ON NOV. 4-6, 2001, PISCATAWAY, NJ, USA,IEEE, 4 November 2001 (2001-11-04), pages 38-44, XP010583886 ISBN: 978-0-7695-7230-7 abstract page 40, left-hand column, line 5 - line 13 page 42, left-hand column, line 10 - line 23 | 1-35 |
| A | WU C E ET AL: "Trace-based analysis and tuning for distributed parallel applications" PARALLEL AND DISTRIBUTED SYSTEMS, 1994. INTERNATIONAL CONFERENCE ON HSINCHU, TAIWAN 19-21 DEC. 1994, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, 19 December 1994 (1994-12-19), pages 716-723, XP010223604 ISBN: 978-0-8186-6555-4 the whole document | 1-35 |

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

3

## INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2008/060117

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| WO 2006030195 A | 23-03-2006 | CN 101084488 A | 05-12-2007 |
| | | EP 1805621 A2 | 11-07-2007 |
| | | JP 2008513853 T | 01-05-2008 |
| | | KR 20070083590 A | 24-08-2007 |

Form PCT/ISA/210 (patent family annex) (April 2005)

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LS,MW,MZ,NA,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MT,NL,NO,PL,PT,RO,SE,SI,SK,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BR,BW,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IS,JP,KE,KG,KM,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LT,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PG,PH,PL,PT,RO,RS,RU,SC,SD,SE,SG,SK,SL,SM,SV,SY,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,ZA,ZM,ZW

（特許庁注：以下のものは登録商標）
１．ＧＳＭ

(74)代理人 100109830
弁理士 福原 淑弘
(74)代理人 100075672
弁理士 峰 隆司
(74)代理人 100095441
弁理士 白根 俊郎
(74)代理人 100084618
弁理士 村松 貞男
(74)代理人 100103034
弁理士 野河 信久
(74)代理人 100119976
弁理士 幸長 保次郎
(74)代理人 100153051
弁理士 河野 直樹
(74)代理人 100140176
弁理士 砂川 克
(74)代理人 100101812
弁理士 勝村 紘
(74)代理人 100070437
弁理士 河井 将次
(74)代理人 100124394
弁理士 佐藤 立志
(74)代理人 100112807
弁理士 岡田 貴志
(74)代理人 100111073
弁理士 堀内 美保子
(74)代理人 100134290
弁理士 竹内 将訓
(74)代理人 100127144
弁理士 市原 卓三
(74)代理人 100141933
弁理士 山下 元
(72)発明者 ジアンニニ、ルイス・アチレ
アメリカ合衆国、カリフォルニア州 ９２１２１、サン・ディエゴ、モアハウス・ドライブ ５７７５
(72)発明者 アンダーソン、ウィリアム・シー.
アメリカ合衆国、カリフォルニア州 ９２１２１、サン・ディエゴ、モアハウス・ドライブ ５７７５

(72)発明者　チェン、スーフェン
アメリカ合衆国、カリフォルニア州　９２１２１、サン・ディエゴ、モアハウス・ドライブ　５７７５

Ｆターム(参考) 5B033 BE05
5B042 GA23　HH30　MA08　MA13

【要約の続き】
構成することを可能にする。