(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2013-58207
(P2013-58207A)
(43)公開日　平成25年3月28日(2013.3.28)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 11/28	(2006.01)	G06F	11/28	315A	5B042
G06F 9/48	(2006.01)	G06F	9/46	310K	
G06F 9/46	(2006.01)	G06F	9/46	410	

審査請求 有　請求項の数 1　ＯＬ　外国語出願　（全 25 頁）

(21)出願番号　特願2012-209762 (P2012-209762)
(22)出願日　平成24年9月24日 (2012.9.24)
(62)分割の表示　特願2009-537298 (P2009-537298)の分割
原出願日　平成19年11月12日 (2007.11.12)
(31)優先権主張番号　11/560,217
(32)優先日　平成18年11月15日 (2006.11.15)
(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　100108855
弁理士　蔵田　昌俊
(74)代理人　100109830
弁理士　福原　淑弘
(74)代理人　100088683
弁理士　中村　誠
(74)代理人　100103034
弁理士　野河　信久

最終頁に続く

(54)【発明の名称】マルチスレッド化デジタル信号プロセッサに関する非侵入型、スレッド選択式デバッギング方法及びシステム

(57)【要約】　（修正有）
【課題】マルチスレッド化ＤＳＰにおけるリアルタイムの挙動に対して非侵入的にデバッグを行う。
【解決手段】ＩＳＤＢがＤＳＰ動作に関してイネーブルにされ（１３２）、ハードウェアブレークポイント１３４、ソフトウェアブレークポイント１３６、ＥＴＭブレークポイント１４０、ＪＴＡＧブレークポイント１４２、または、外部ブレークポイント１４４が存在すると、デバッギング動作１３８に進む。ＩＳＴＥＰデバッギングが有効である場合は、ＩＳＴＥＰデバッギング１５０を行う。命令スタッフィング動作が有効である場合は、命令スタッフィング動作１５４を行う。コアＤＳＰリセット命令がデバッギング動作によって生成されている場合は、コアＤＳＰデジタル信号プロセッサをリセットする（１５６）。
【選択図】図６

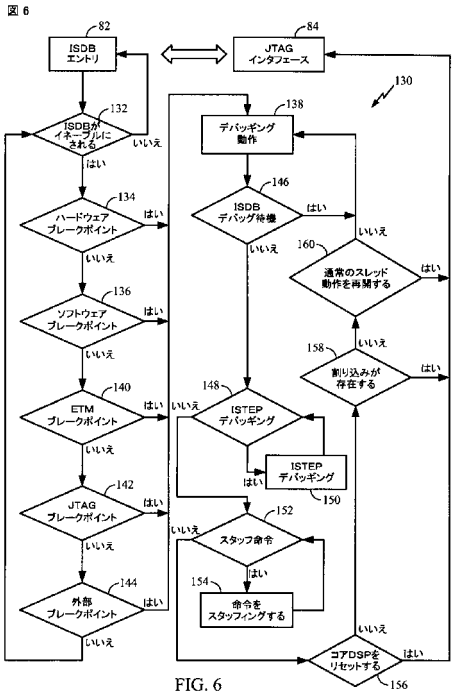

FIG. 6

【特許請求の範囲】
【請求項１】
マルチスレッド化デジタル信号プロセッサをデバッグするための非侵入型方法であって、
前記マルチスレッド化デジタル信号プロセッサの少なくとも１つ以上のスレッドを用いて複数の処理命令をマルチスレッド化プロセスにおいて実行することと、
少なくとも１つのデバッギングイベントを生成するための１つ以上のブレークポイント命令を識別することと、
前記ブレークポイント命令のうちの少なくとも１つを実行することに応じて前記少なくとも１つのデバッギングイベントを生成することと、
前記少なくとも１つのデバッギングイベントに応じて複数のデバッギング命令を実行することであって、前記デバッギング命令は、前記マルチスレッド化デジタル信号プロセッサの少なくとも１つ以上のスレッドをデバッギングモードに移行させることによって前記マルチスレッド化デジタル信号プロセッサにおいて前記複数の処理命令を前記実行することを非侵入方式でデバッグするためのデバッギング命令であることと、
前記複数のデバッギング命令を前記実行することを報告するために前記複数のデバッギング命令を前記実行することからの少なくとも１つのデバッギングリターンを生成すること、とを備える、非侵入型方法。
【請求項２】
前記少なくとも１つのデバッギングイベントは、前記マルチスレッド化デジタル信号プロセッサ内において内部生成されたインスタンスに応じて発生する請求項１に記載の方法。
【請求項３】
前記少なくとも１つのデバッギングイベントは、前記マルチスレッド化デジタル信号プロセッサ内において外部生成されたインスタンスに応じて発生する請求項１に記載の方法。
【請求項４】
前記少なくとも１つのデバッギングイベントは、前記マルチスレッド化デジタル信号プロセッサ内においてプログラムカウンタが予め決められたカウントに達することに応じて発生する請求項１に記載の方法。
【請求項５】
前記マルチスレッド化デジタル信号プロセッサの前記少なくとも１つ以上のスレッドは、前記デバッギングモードから出る請求項１に記載の方法。
【請求項６】
少なくとも１つ以上の命令スタッフィング命令は、前記マルチスレッド化デジタル信号プロセッサの動作に関連する予め決められたデバッギングアルゴリズムを実行する請求項１に記載の方法。
【請求項７】
前記マルチスレッド化デジタル信号プロセッサ内の前記少なくとも１つ以上のデバッギング命令は、前記複数のデバッギング命令を前記実行することと前記複数の処理命令を前記実行することとの間においてＪＴＡＧインタフェースインタフェースを用いる請求項１に記載の方法。
【請求項８】
マルチスレッド化デジタル信号プロセッサを非侵入方式でデバッグするための動作に関するシステムであって、
デジタル信号マルチスレッド化プロセスにおいて複数の処理命令を実行するための前記マルチスレッド化デジタル信号プロセッサの１つ以上のスレッドと、
少なくとも１つのデバッギングイベントを生成するための一組のブレークポイント命令と、
前記ブレークポイント命令のうちの少なくとも１つを実行することに応じて少なくとも

１つのデバッギングイベントを生成するためのデバッギングイベント生成命令と、
前記少なくとも１つのデバッギングイベントに応じて前記複数のデバッギング命令を実行するための前記スレッドのうちの１つ以上であって、前記複数のデバッギング命令は、前記マルチスレッド化デジタル信号プロセッサの少なくとも１つ以上のスレッドをデバッギングモードに移行させることによって前記マルチスレッド化デジタル信号プロセッサにおいて前記複数の処理命令を前記実行することを非侵入方式でデバッギングするためのデバッギング命令である前記スレッドのうちの１つ以上と、
前記マルチスレッド化デジタル信号プロセッサの前記少なくとも１つ以上のスレッドにおいて前記複数のデバッギング命令を前記実行することを報告するために前記複数のデバッギング命令を前記実行することからの少なくとも１つのデバッギングリターンを生成するためのデバッギングリターン命令と、を備える、システム。
【請求項９】
前記マルチスレッド化デジタル信号プロセッサ内においてプログラムカウンタが予め決められたカウントに達することに応じて前記デバッギングイベントが発生するための回路と命令とをさらに備える請求項８に記載のシステム。
【請求項１０】
前記マルチスレッド化デジタル信号プロセッサの前記少なくとも１つ以上のスレッドを前記デバッギングモードから出すための回路と命令とをさらに備える請求項８に記載のシステム。
【請求項１１】
前記マルチスレッド化デジタル信号プロセッサの動作に関連する予め決められたデバッギングアルゴリズムを実行するための一組の命令スタッフィング命令をさらに備える請求項８に記載のシステム。
【請求項１２】
ＪＴＡＧインタフェースを用いて前記マルチスレッド化デジタル信号プロセッサを非侵入方式でデバッグするための回路と命令とをさらに備える請求項８に記載のシステム。
【請求項１３】
パーソナル電子デバイスをサポートするための動作に関するマルチスレッド化デジタル信号プロセッサであって、マルチスレッド化処理中にデバッグするための非侵入型手段を備え、
前記マルチスレッド化デジタル信号プロセッサの少なくとも１つ以上のスレッドを用いてマルチスレッド化プロセスにおいて複数の処理命令を実行するための手段と、
少なくとも１つのデバッギングイベントを生成するための１つ以上のブレークポイント命令を識別するための手段と、
前記ブレークポイント命令のうちの少なくとも１つを実行することに応じて少なくとも１つのデバッギングイベントを生成するための手段と、
前記少なくとも１つのデバッギングイベントに応じて前記複数のデバッギング命令を実行するための手段であって、前記デバッギング命令は、前記マルチスレッド化デジタル信号プロセッサの少なくとも１つ以上のスレッドをデバッギングモードに移行させることによって前記マルチスレッド化デジタル信号プロセッサにおいて前記複数の処理命令を前記実行することを非侵入方式でデバッグするためのデバッギング命令である手段と、
前記マルチスレッド化デジタル信号プロセッサの前記少なくとも１つ以上のスレッドにおいて前記複数のデバッギング命令を前記実行することを報告するために前記複数のデバッギング命令を前記実行することからの少なくとも１つのデバッギングリターンを生成するための手段と、を備える、マルチスレッド化デジタル信号プロセッサ。
【請求項１４】
前記マルチスレッド化デジタル信号プロセッサ内において内部生成されたインスタンスに応じて前記デバッギングイベントが発生するための手段をさらに備える請求項１３に記載のデジタル信号プロセッサシステム。
【請求項１５】

前記マルチスレッド化デジタル信号プロセッサ内において外部生成されたインスタンスに応じて前記デバッギングイベントが発生するための手段をさらに備える請求項１３に記載のマルチスレッド化デジタル信号プロセッサ。

【請求項１６】

前記マルチスレッド化デジタル信号プロセッサ内においてプログラムカウンタが予め決められたカウントに達することに応じて前記デバッギングイベントが発生するための手段をさらに備える請求項１３に記載のマルチスレッド化デジタル信号プロセッサ。

【請求項１７】

前記マルチスレッド化デジタル信号プロセッサの前記少なくとも１つ以上のスレッドを前記デバッギングモードから出すための手段をさらに備える請求項１３に記載のマルチスレッド化デジタル信号プロセッサ。

【請求項１８】

前記マルチスレッド化デジタル信号プロセッサの動作に関連する予め決められたデバッギングアルゴリズムを実行するための命令をスタッフィングするための手段をさらに備える請求項１３に記載のマルチスレッド化デジタル信号プロセッサ。

【請求項１９】

前記複数のデバッギング命令を前記実行することと前記複数の処理命令を前記実行することとの間においてＪＴＡＧインタフェースを用いるための手段をさらに備える請求項１３に記載のマルチスレッド化デジタル信号プロセッサ。

【請求項２０】

マルチスレッド化デジタル信号プロセッサを非侵入方式でデバッグするための前記マルチスレッド化デジタル信号プロセッサにおける処理命令に関してここにおいて具現化されたコンピュータによって読み取り可能なプログラムコード手段を有するコンピュータによって使用可能な媒体であって、

前記マルチスレッド化デジタル信号プロセッサの少なくとも１つ以上のスレッドを用いてマルチスレッド化プロセスにおいて複数の処理命令を実行するためのコンピュータによって読み取り可能なプログラムコード手段と、

少なくとも１つのデバッギングイベントを生成するために１つ以上のブレークポイント命令を識別するためのコンピュータによって読み取り可能なプログラムコード手段と、

前記ブレークポイント命令のうちの少なくとも１つを実行することに応じて少なくとも１つのデバッギングイベントを生成するためのコンピュータによって読み取り可能なプログラムコード手段と、

前記少なくとも１つのデバッギングイベントに応じて前記複数のデバッギング命令を実行するためのコンピュータによって読み取り可能なプログラムコード手段であって、前記デバッギング命令は、前記マルチスレッド化デジタル信号プロセッサの少なくとも１つ以上のスレッドをデバッギングモードに移行させることによって前記マルチスレッド化デジタル信号プロセッサにおいて前記複数の処理命令を前記実行することを非侵入方式でデバッグするためのデバッギング命令であるコンピュータによって読み取り可能なプログラムコード手段と、

前記マルチスレッド化デジタル信号プロセッサの前記少なくとも１つ以上のスレッドにおいて前記複数のデバッギング命令を前記実行することを報告するために前記複数のデバッギング命令を前記実行することからの少なくとも１つのデバッギングリターンを生成するためのコンピュータによって読み取り可能なプログラムコード手段と、を備える、コンピュータによって使用可能な媒体。

【請求項２１】

前記少なくとも１つのデバッギングイベントであって、前記マルチスレッド化デジタル信号プロセッサ内において内部生成されたインスタンスに応じて発生するデバッギングイベント、に応じて前記複数のデバッギング命令を実行するためのコンピュータによって読み取り可能なプログラムコード手段をさらに備える請求項２０に記載のコンピュータによって使用可能な媒体。

【発明の詳細な説明】
【技術分野】
【０００１】
開示される主題は、データ通信に関するものである。本開示は、より具体的には、マルチスレッド化デジタル信号プロセッサに関する斬新な及び改良された非侵入型、スレッド選択式デバッギング方法及びシステムに関するものである。
【背景技術】
【０００２】
電気通信及びその他の種類の電子装置とそれをサポートする映像、複雑な音声、テレビ会議及びその他のリッチソフトウェアアプリケーションでは、信号処理を含むことがますます多くなっている。信号処理は、複雑であるが反復的なアルゴリズムにおいて高速な数学計算及びデータ生成を行うことが要求される。多くのアプリケーションは、リアルタイムでの演算が要求され、すなわち、信号は時間の連続関数であり、数値処理のためにサンプリングしてデジタル信号に変換しなければならない。プロセッサは、到着したサンプルに関する個別の演算を行うアルゴリズムを実行しなければならない。デジタル信号プロセッサ（ＤＳＰ）のアーキテクチャは、該アルゴリズムを処理するように最適化される。優れた信号処理エンジンの特徴は、高速で柔軟な算術演算ユニットと、演算ユニットへの又は演算ユニットからの制限されないデータフローと、演算ユニット内における拡張された精密な動的範囲と、デュアルアドレス生成器と、効率的なプログラムシーケンシングと、プログラミングの容易さと、を含む。
【０００３】
ＤＳＰ技術の１つの有望な用途は、通信システム、例えば、衛星又は地上リンクを通じてのユーザー間における音声とデータの通信をサポートする符号分割多元接続（ＣＤＭＡ）システム、を含む。多元接続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 OR TERRESTRIAL REPEATERS”（衛星又は地上中継器を用いた拡散スペクトル多元接続通信システム）という題名を有する米国特許番号４，９０１，３０７及び“SYSTEM AND METHOD FOR GENERATING WAVEFORMS IN A CDMA CELLULAR TELEHANDSET SYSTEM”（ＣＤＭＡセルラーテレハンドセットシステムにおいて波形を生成するためのシステム及び方法）という題名を有する米国特許番号５，１０３，４５９において開示されており、両特許とも、請求される主題の譲受人に譲渡されている。
【０００４】
ＣＤＭＡシステムは、典型的には、１つ以上の基準に準拠するように設計される。１つの該第１世代の基準は、“二重モード広帯域拡散スペクトルセルラーシステムに関するＴＩＡ／ＥＩＡ／ＩＳ－９５端末－基地局互換性基準”であり、以下ではＩＳ－９５基準と呼ばれる。ＩＳ－９５ＣＤＭＡシステムは、音声データ及びパケットデータを送信することができる。第１世代よりも効率的にパケットデータを送信することができるより新しい世代の基準が、“第３世代パートナーシッププロジェクト”（３ＧＰＰ）と呼ばれるコンソーシアムによって提供され、一般人が簡単に入手可能である一組の文書、例えば、文書番号３Ｇ　ＴＳ　２５．２１１、３Ｇ　ＴＳ　２５．２１２、３Ｇ　ＴＳ　２５．２１３、及び３Ｇ　ＴＳ　２５．２１４、において具体化されている。３ＧＰＰ基準は、以下ではＷ－ＣＤＭＡ基準と呼ばれる。
【０００５】
例えばＷ－ＤＣＭＡ基準を採用する複雑なＤＳＰ運用ソフトウェアは、強固な開発ツールが要求される。該開発ツールは、符号の生成、インテグレーション、試験、デバッギング、及びアプリケーション性能の評価のための開発ツールを含むことができる。ソフトウェア又は複雑なＤＳＰアプリケーション、例えば高度な電気通信アプリケーション、を開発及び運用する際には、精巧であるがその一方で非侵入型のデバッギングソフトウェアが必要である。すなわち、デバッギングソフトウェアアプリケーションは、ソフトウェアの欠陥及び運用上の問題の訂正をモニタリング、試験、及びサポートする上で十分に強固で

なければならないだけでなく、デバッギング動作中にコアプロセッサソフトウェアと干渉しないように動作できなければならない。さもないと、コア処理ソフトウェア内のいずれの問題も、該デバッギングソフトウェアを使用中に検出することができず又は適切に検出することができない。

【０００６】

例えば、リアルタイム映像ソフトウェアを最適化及びデバッグするためには、サイクルが正確なプロファイリング及び非侵入型のデバッギングという特長が極めて重要である。さらに、開発ボードは、広範なリアルタイムの試験を可能にするために大量の試験データをプロセッサ内に又はプロセッサから移動させるためのサポートが必要である。これらの及びその他の状況は、非侵入型のコアプロセッサソフトウェアデバッギングを要求する。従って、マルチスレッド化デジタル信号プロセッサにおいては、マルチスレッド化オペレーティングソフトウェアを非侵入的にデバッグする必要がある。さらに、リアルタイムオペレーティングソフトウェアが存在する環境においては、侵入型デバッギングプログラムが引き起こすことがあるソフトウェアの変更は、プロセッサ内において起きることを明らかに変更させ、必要なデバッギング動作に加えてソフトウェアの運用上の問題の決定に対しても悪影響を及ぼす可能性がある。

【０００７】

上記により、対話形式でかつマルチスレッド化デジタル信号プロセッサのリアルタイムの挙動に対して非侵入的に動作することができるＤＳＰデバッギングプロセスが必要であることが明確になる。

【０００８】

マルチスレッド化されたＤＳＰにおいては、１つ以上のスレッド間における対話もコアプロセッサの誤動作を引き起こすことがある。このことは、個々のスレッドがプログラミングされたとおりに及び希望されるとおりに個々に動作できるにもかかわらず当てはまることがある。さらに、動作中のスレッドの異なる組合せも、デバッギングソフトウェアによる解析を行うのが有益である異なる種類のプログラミング上の問題を引き起こす可能性がある。

【０００９】

さらに、マルチスレッド化ＤＳＰにおいては、デバッギング動作を行うことが望まれる数多くのポイント、すなわちブレークポイントが存在することができる。該ブレークポイントは、コアプロセッサのアプリケーションに影響を及ぼすハードウェアの状態、ソフトウェアの状態、外部の状態、及びその他の状態に起因して生じることができる。好ましいことに、柔軟な型のマルチスレッド化ＤＳＰデバッギングソフトウェアアプリケーションは、コアプロセッサアプリケーションのデバッギングを要求する非常に様々な状態に対処する。実際、柔軟であるためには、デバッギングソフトウェアがデバッギングソフトウェアの動作を必要とする状態に従って動的に変化することが要求される。

【００１０】

これらの考慮事項を念頭に置き、個々のスレッドのデバッギングをサポートするマルチスレッド化ＤＳＰデバッギングプロセスが必要であることは明確である。

【００１１】

コア処理アプリケーションのニーズに従って１つ、２つ、又はそれよりも多いスレッドのスレッド選択式デバッギング動作を許可するマルチスレッド化ＤＳＰデバッギングプロセスの必要性も存在する。

【００１２】

マルチスレッド化ＤＳＰがＤＳＰの動作に影響を及ぼす非常に様々な状態、例えば、デバッギングブレークポイントを確立することができるハードウェア状態、ソフトウェア状態、外部状態、及びその他の状態、を有するデバッギングプロセスに従事することを許可する方法及びシステムの必要性も存在する。

【発明の概要】

【００１３】

マルチスレッド化デジタル信号プロセッサに関する非侵入型スレッド選択式デバッギング方法及びシステムを提供するための技法が開示され、該技法は、デジタル信号プロセッサの動作と、パソコン、パーソナルデジタルアシスタント、ワイヤレスハンドセット、及び同様の電子デバイス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になるソフトウェアアプリケーションに関するデジタル信号プロセッサ命令の効率的な使用と、の両方を向上させ、さらに関連するデジタルプロセッサの速度及びサービス品質を向上させる。

【００１４】

開示される主題の一側面により、マルチスレッド化デジタル信号プロセッサの非侵入型デバッギングに関する方法及びシステムが提供される。前記方法及びシステムは、デバッギング命令を第１の組のレジスタ内に格納し、処理命令を第２の組のレジスタ内に格納することを許容する。前記第２の組のレジスタは、前記第１の組のレジスタと異なる。前記方法及びシステムは、前記マルチスレッド化デジタル信号プロセッサの少なくとも１つ以上のスレッドを用いてマルチスレッド化プロセスにおいて処理命令をさらに実行する。前記処理命令の部分組は、少なくとも１つのデバッギングイベントを生成するためのブレークポイント命令である。前記プロセスは、前記ブレークポイント命令のうちの少なくとも１つの前記実行に応じて少なくとも１つのデバッギングイベントを生成し、前記デバッギングイベントに応じてデバッギング命令を実行し、前記デバッギング命令は、前記マルチスレッド化デジタル信号プロセッサの少なくとも１つ以上のスレッドをデバッギング動作モードに移行させることによって前記マルチスレッド化デジタル信号プロセッサ内における処理命令の前記実行を非侵入方式でデバッグすることを許容する。本開示は、前記マルチスレッド化デジタル信号プロセッサのスレッドの前記部分組におけるデバッギング命令の前記実行を報告するために前記複数のデバッギング命令の前記実行からのデバッギングリターンを生成する。

【００１５】

開示される主題のこれらの利点とその他の利点、及び追加の斬新な特長は、ここにおいて提供される説明から明確になるであろう。この発明の概要の意図は、請求される主題に関する包括的な説明を提供することではなく、主題の機能の一部について簡単に概説することである。ここにおいて提供されるその他のシステム、方法、特長及び利点は、以下の図及び発明を実施するための形態を検討し次第当業者に明確になるであろう。これらのすべての追加のシステム、方法、特長及び利点は、この説明の中に含められ、さらに添付される請求項の適用範囲内に含められることが意図される。

【図面の簡単な説明】

【００１６】

開示される主題の特長、性質、及び利点は、以下の発明を実施するための形態を図面と併読することでより明確になるであろう。なお、図面全体において同様の参照文字は同様の要素を識別する。

【図１】本実施形態を実装することができる通信システムの単純化されたブロック図である。

【図２】本実施形態の教示を実行するためのＤＳＰアーキテクチャを示す。

【図３】開示される主題の技術的利点を提供する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】非侵入型デバッギング動作モードにおける動作を含む本開示のモード制御に関する側面の機能ブロック図である。

【図５】本開示のデバッギング動作を達成するためのモード制御レジスタを示した図である。

【図６】本開示の非侵入型デバッギングアルゴリズムの様々な側面に関する流れ図である。

【発明を実施するための形態】

【００１７】

マルチスレッド化デジタル信号プロセッサに関する非侵入型スレッド選択式デバッギング方法及びシステムに関する開示される主題は、ここにおいて提示される利益が有利であることができるあらゆる型のマルチスレッド化処理に関して用途を有する。１つの該用途は、電気通信において現れ、特に、１つ以上のデジタル信号処理回路を採用するワイヤレスハンドセットにおいて現れる。該ワイヤレスハンドセットをどのようにして用いることができるかについて説明するために、図１は、開示される割り込み処理方法及びシステムの提示される実施形態を実装することができる通信システム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において、データは、データ源１４から、１つ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化、符号化、及び処理する送信（ＴＸ）データプロセッサ１６に、典型的にはブロックで送信される。次に、アナログ信号が、ベースバンド信号を変調、フィルタリング、増幅、及びアップコンバージョンして変調された信号を生成する送信機（ＴＭＴＲ）１８に提供される。変調された信号は、アンテナ２０を介して１つ以上の受信機ユニットに送信される。

【００１８】

受信機ユニット２２においては、送信された信号は、アンテナ２４によって受信されて受信機（ＲＣＶＲ）２６に提供される。受信機２６内において、受信された信号が増幅、フィルタリング、ダウンコンバージョン、復調、及びデジタル化されて同相の（Ｉ）及び（Ｑ）サンプルが生成される。これらのサンプルは、受信（ＲＸ）データプロセッサ２８によって復号及び処理されて送信されたデータが復元される。受信機ユニット２２における復号及び処理は、送信機ユニット１２において実行される符号化及び処理を補完する形で実行される。復元されたデータは、データシンク３０に提供される。

【００１９】

上述される信号処理は、音声、映像、パケットデータ、メッセージ送信、及びその他の種類の通信を一方向に送信することをサポートする。双方向通信システムは、2方向データ送信をサポートする。しかしながら、説明を単純化するため、図１には他方の方向に関する信号処理は示されていない。通信システム１０は、地上リンクを通じてのユーザー間における音声及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、時分割多元接続（ＴＤＭＡ）通信システム（例えば、ＧＳＭ（登録商標）システム）、周波数分割多元接続（ＦＤＭＡ）通信システム、又はその他の多元接続通信システムであることができる。１つの特定の実施形態においては、通信システム１０は、Ｗ－ＣＤＭＡ基準に準拠するＣＤＭＡシステムである。

【００２０】

図２は、図１の送信データプロセッサ１６及び受信データプロセッサ２８として働くことができるＤＳＰ４０アーキテクチャを示す。ＤＳＰ４０は、ここにおいて提示される教示及び概念を実効的に用いることができる非常に数多くの可能なデジタル信号プロセッサ実施形態のうちの１つの実施形態を表すにすぎないことを強調する。従って、ＤＳＰ４０においては、スレッドＴ０：Ｔ５（参照番号４２乃至５２）は、異なるスレッドからの命令の組を含む。回路５４は、命令アクセス機構を表し、スレッドＴ０：Ｔ５に関する命令をフェッチするために用いられる。回路５４に関する命令は、命令待ち行列５６内に入れられる。命令待ち行列５６内の命令は、プロセッサパイプライン６６内に発行する準備が整った状態である（下記参照）。命令待ち行列５６から、発行論理回路５８によって単一のスレッド、例えばスレッドＴ０、を選択することができる。選択されたスレッドのレジスタファイル６０が読まれ、読み取られたデータがＳＬＯＴ０：ＳＬＯＴ３に関する実行データ経路６２に送られる。ＳＬＯＴ０：ＳＬＯＴ３は、この例においては、本実施形態において採用されるパケットグループの結合に関するものである。

【００２１】

実行データ経路６２からの出力は、ＤＳＰ４０の動作からの結果を戻すために、同じく個々のスレッドＴ０：Ｔ５を受け入れるように構成されたレジスタファイル書き込み回路６４に向かう。従って、回路５４及びそれよりも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、最大で６つの

スレッドＴ０：Ｔ５を有する単一のプロセッサを採用することができる。プロセッサパイプライン６６は、６つのステージを有し、これは、回路５４からレジスタ６０及び６４にデータ項目をフェッチするために必要な最低限のプロセッササイクル数と一致する。ＤＳＰ４０は、プロセッサパイプライン６６内の異なるスレッドＴ０：Ｔ５の命令を同時並行して実行する。すなわち、ＤＳＰ４０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：Ｔ５の命令を区別するための内部タギング機構、及びスレッドスイッチをトリガーする機構を提供する。

【００２２】

従ってＤＳＰ４０は、非常に様々な信号、画像、及び映像処理用途において高性能及び低電力であるように設計された汎用デジタル信号プロセッサを提供する。図３は、開示される主題の１つの表示に関する関連づけられた命令セットアーキテクチャの幾つかの側面を含むＤＳＰ４０アーキテクチャの概要を示す。ＤＳＰ４０アーキテクチャの実装は、インターリービングされたマルチスレッド化（ＩＭＴ）をサポートする。この実行モデルにおいては、ハードウェアは、パイプライン内の異なるスレッドからの命令をインターリービングすることによって複数のハードウェアスレッドＴ０：Ｔ５の同時並行実行をサポートする。この特長は、ＤＳＰ４０がコアとメモリの高い利用を依然として維持しながら積極的なクロック周波数を含むことを許容する。ＩＭＴは、アウトオブオーダー実行、広範な転送ネットワーク、等の高コストの補償機構を必要とせずに高スループットを提供する。さらに、ＤＳＰ４０は、ＩＭＴの変形、例えば、Ｍ．アーメド、等による、“Variable Interleaved Multithreaded Processor Method and System”（可変のインターリービングされたマルチスレッド化プロセッサに関する方法及びシステム）及び“Method and System for Variable Thread Allocation and Switching in a Multithreaded Processor”（マルチスレッド化されたプロセッサにおける可変スレッド割り当て及び切り換えに関する方法及びシステム）という題名を有する共通譲渡米国特許出願において開示される変形及び斬新な手法、を含むことができる。

【００２３】

図３は、特に、開示される主題の教示を採用することができる単一のスレッドに対して適用されるＤＳＰ４０に関するコア処理アーキテクチャ７０ブロック図を提供する。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を介して命令を受け取る共有命令キャッシュ７２を描き、これらの命令は、混合された１６ビットと３２ビットの命令を含む。これらの命令は、スレッドＴ０：Ｔ５のシーケンサ７６、ユーザー制御レジスタ７８、及びスーパバイザ制御レジスタ８０に届く。開示される主題のコアレベルシステムアーキテクチャは、ＪＴＡＧインタフェース８４を介してコアプロセッサ７０をインタフェースするインシリコンデバッギングシステム（ＩＳＤＢ）８２も含み、以下ではこれらの両方がさらに詳細に説明される。

【００２４】

シーケンサ７６は、ハイブリッド２方向スーパースカラー命令及び４方向ＶＬＩＷ命令をＳ－パイプユニット８６、Ｍ－パイプユニット８８、ＬＤ［Ｌｏａｄ］－パイプ９０、及びＬＤ／ＳＴ［Ｓｔｏｒｅ］－パイプユニット９２に提供し、これらの全ユニットは、汎用レジスタ９４と通信する。ＡＸＩバス７４は、バスＩ／Ｆ７３を介して、スレッドＴ０：Ｔ５への共有データキャッシュ９６ＬＤ／ＳＴ命令とも通信する。オプションのＬ２キャッシュ／ＴＣＭ９８信号は、共有データＴＣＭ１００を有するＬＤ／ＳＴ命令を含み、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専用コントローラ１０４は、Ｔ０：Ｔ５と割り込みを通信し、割り込みコントローラ命令と、デバッギング命令と、タイミング命令と、を含む。グローバル制御レジスタ１０６は、スレッドＴ０：Ｔ５と制御レジスタ命令を通信する。

【００２５】

従って、ＤＳＰ４０は、６つの仮想ＤＳＰコアを含み、各仮想ＤＳＰコアは、グローバル制御レジスタ１０６と、プライベートスーパバイザ制御レジスタ８０と、を含む。グローバル制御レジスタ１０６は、全スレッド間で共有される。各スレッドは、共通データキ

ャッシュ及び共通命令キャッシュを共有する。ロード、ストア、及びフェッチの各動作は、共通のバスインタフェースによって対処される。高性能ＡＸＩバス７４及びそれよりも低性能のＡＨＢバス１０２は、データ及び命令トラフィックをオフコアメモリ及び周辺装置に接続するために用いられる。統合されたレベル２メモリ（キャッシュ及び／又はＴＣＭ）入力９８はオプションである。周辺装置のアクセスは、メモリによってマッピングされるロード及びストアを通じて行うことができる。ＡＨＢとＡＸＩとの間における物理アドレスパーティションは、ＭＳＭレベルで構成することができる。

【００２６】

明確なことであるが、ＤＳＰ４０に関する提示されるアーキテクチャは、経時で発展及び変化することができる。例えば、ＤＳＰ４０が用いることができる命令キャッシュ数は、６から１に、又はその他のキャッシュ数に変更することができる。ＴＣＭ１００におけるスーパースカラーディスパッチＬ１データ、及びその他のアーキテクチャ上の側面は、変更することができる。しかしながら、本主題は、非常に様々なコンフィギュレーションにおいて及びＤＳＰ４０の修正の大規模な系統に関して継続的な関連性を有することができる。

【００２７】

ＩＳＤＢ８２は、ＪＴＡＧインタフェース８４を介して、ＤＳＰ４０に関するハードウェアデバッガを提供する。ＩＳＤＢ８２は、システム又はスーパバイザ専用レジスタを共有することによってＪＴＡＧインタフェース８４を通じてソフトウェアデバッグ機能を提供し、これらのレジスタは、１つのスレッドベースのスーパバイザ制御レジスタ８０、及び全スレッド間におけるグローバル制御レジスタ１０６に分割される。システム制御レジスタは、１つのスレッドごとの割り込みと例外制御及び１つのスレッドごとのメモリ管理活動に関して用いられる。グローバルレジスタは、デバッギング動作のためにＩＳＤＢ８２と対話することを許容する。

【００２８】

ＩＳＤＢ８２は、ＤＳＰ４０が動作する間にソフトウェア開発者が自己のソフトウェアをデバッグするのを可能にする。ＩＳＤＢ８２ハードウェアは、ＩＳＤＢ８２において動作中のソフトウェアデバッガプログラムと組み合わせることで、ＤＳＰ４０オペレーティングシステムソフトウェアをデバッグするために用いることができる。ＩＳＤＢ８２は、デバッギングハードウェアスレッドを個々にサポートする。ユーザーは、スレッド実行を中断すること、スレッドレジスタを閲覧及び変更すること、命令とデータメモリ、単一ステップスレッド、を閲覧及び変更すること、スレッドに命令をスタッフィングすること、及びスレッド実行を再開することができる。信頼されるユーザーは、すべてのＩＳＤＢ８２の特長にアクセス可能であり、信頼されないユーザーは、１つ以上の特長にアクセス可能である。

【００２９】

ＩＳＤＢ８２は、プログラムカウンタ上に常駐するＩＳＤＢ８２デバッギングソフトウェアと通信するためにデバッガインタフェースカードと通信することができ、すべてＪＴＡＧインタフェース８４を通じて行うことができる。ホストデバッガソフトウェアは、ＩＳＤＢ制御レジスタを読み取る及び書き込むことによってＩＳＤＢ８２と対話することができる。通信は、例えば、読み取り／書き込みの対象となるＩＳＤＢレジスタを識別する４０ビットパケット、及び３２ビットデータペイロードを通じて行うことができる。この動作をサポートするパケットフォーマットは、各々の幅が３２ビットであることができる最大で６４の制御レジスタであることができる。

【００３０】

ＩＳＤＢ８２は、デバッギング動作中にセキュリティを制御するための信頼されるレジスタを含む。ＩＳＤＢ８２ｔｒｕｓｔｅｄがセットされた場合は、すべてのＩＳＤＢ８２レジスタがデバッガソフトウェアにとって可視であり、すべてのＩＳＤＢコマンドを使用可能である。ＩＳＤＢ８２ｔｒｕｓｔｅｄがクリアされた場合は、ＩＳＤＢ８２は、制限された一組の動作のみを許可する。

【００３１】
一定のＩＳＤＢ８２レジスタがコアソフトウェアにとって可視であるようにすることができる。これらのレジスタは、スーパバイザ（ＳＵＰＥＲＶＩＳＯＲ）モード制御レジスタ転送命令を介してアクセス可能である。コア命令は、ブレークポイント命令を含む。ＩＳＤＢｔｒｕｓｔｅｄがセットされたときには、この命令は、実行中のスレッドにデバッグ動作モード１２０に入らせる。この移行は、スレッド制御をＩＳＤＢ８２に移行させる。ブレークポイントを実行したスレッドに加えて、その他のスレッドは、ＩＳＤＢ８２プログラミングに従って選択的にデバッグ（ＤＥＢＵＧ）モード１２０に入ることができる。ＩＳＤＢ８２が信頼されないか又はイネーブルにされない場合は、この命令はＮＯＰとして処理される。好ましいことに、ブレークポイント命令は、パケット内における唯一の命令である。

【００３２】
図４は、デバッギングプロセス中におけるＩＳＤＢ８２の動作を含むＤＳＰ４０の様々なモード制御側面に関する処理モード図１１０を示す。図５は、本開示のデバッギング動作を達成させるためのモード制御レジスタ１２２を示す。一実施形態においては、モード制御レジスタ１２２は、開示される動作モードへの／からの移行を援助し、ビット３１乃至２２を占有する予約セクションと、待機ビット２１乃至１６と、予約ビット１６乃至６と、エラービット５乃至０と、を含む。モード制御レジスタ１２２は、数多くの異なる方法で実装できるが、図５の例示される実施形態は、ＩＳＤＢ８２が所有する様々な性質及びＩＳＤＢ８２が可能にする動作を含むＩＳＤＢ８２に関する以下の説明を理解するのに役立つことができる。

【００３３】
次に、図４に関して、ＤＳＰ４０は、全スレッドに対してグローバルであり個々のスレッドに対してローカルである処理モードをサポートする。各ＤＳＰ４０ハードウェアスレッドは、すべて図４において示されるように、２つの実行モード、ユーザー（ＵＳＥＲ）モード１１２とスーパバイザモード１１４、及び３つの非処理モードすなわち待機（ＷＡＩＴ）モード１１６、ＯＦＦモード１１８、及びデバッグモード１２０、を個々にサポートする。スレッドのモードは、その他のスレッドと独立しており、例えば、１つのスレッドは待機モード１１６であることができ、他のスレッドは、ユーザーモード１１２であり、以下同様である。図４の１つのスレッドごとのモード状態図は、様々な命令又はイベントによってサポートされる。これらは、“例外”すなわち内部例外イベントと、“Ｉｎｔ”すなわち外部割り込みイベントと、“ＲＴＥ”すなわち例外モードからのソフトウェアリターン命令と、“ＳＳＲ”すなわちＳＳＲレジスタ命令の更新と、あらゆるモードから入力することができる“停止”すなわちソフトウェア停止命令と、同じくあらゆるモードから入力することができる“開始”すなわちソフトウェア開始命令と、“トラップ”すなわちソフトウェアトラップ命令と、“待機”すなわちソフトウェア待機命令と、“再開”すなわちソフトウェア再開命令と、“ＤＥ”すなわちデバッグイベントと、“ＤＲ”すなわちデバッグ命令と、を含む。請求される主題の異なる実装における機能は、ここにおいて提示される機能とわずかに異なることができる一方で、“開始”、“待機”、“再開”、“ＤＥ”、及び／又は“ＤＲ”の意味は、請求される主題の適用範囲と一致する最も広義の解釈を行うことができる。

【００３４】
レジスタは、ＤＳＰ４０においてユーザーモード１１２及びスーパバイザモード１１４の両方において利用可能である。ユーザーモードレジスタは、一組の汎用レジスタ及び一組の制御レジスタに分割される。汎用レジスタは、アドレス生成、スカラー及びベクトル算術を含むすべての汎用演算に関して用いられる。制御レジスタは、ハードウェアループ、プレディケート、等の特殊目的の機能をサポートする。

【００３５】
汎用レジスタは、幅が３２ビットであり、単一レジスタとして又は２つのレジスタの整列された対としてアクセスすることができる。汎用レジスタファイルは、命令に関する全

オペランドを提供し、ロード／ストアに関するアドレスと、数値命令に関するデータオペランドと、ベクトル命令に関するベクトルオペランドと、を含む。

【００３６】

ＤＳＰ４０レジスタ及び命令は、標準的なＣ言語規約を採用するソフトウェアスタックの効率的な使用をサポートする。スタックは、高アドレスから低アドレスに向かって増大する。スタックポインタレジスタは、スタック最上部における最後の有効要素を指す。プッシュ動作は、最初にスタックポインタの数を減らし次にスタックにデータを書き込み、ポップ動作は、スタックから読み取ってスタックポインタの数を増やす。

【００３７】

スタック上の手順フレームは、関数呼び出しに関するリターンアドレス及び手順によって必要とされるすべてのローカル変数とデータを含む。さらに、フレームポインタがリターンアドレスの後に格納される。このフレームポインタは、スタック上における前手順フレームのアドレスを含む。この目的は、デバッガがメモリ内のスタックを検討して呼び出しシーケンス、関数パラメータ、等を簡単に決定するのを可能にすることによってデバッグを容易にすることである。

【００３８】

デバッグモード１２０は、スレッドがＩＳＤＢ８２からのコマンドを待っている特別な状態である。例えばソフトウェアブレークポイント命令の実行、ＩＳＤＢ８２からのブレークコマンド、又はハードウェアブレークポイントの発生によって、ＩＳＤＢデバッグイベントが生じるごとに、示されたスレッドがデバッグモード１２０に入ることができる。デバッグモード１２０においては、コアは、ＪＴＡＧインタフェース８４からのコマンドを介してＩＳＤＢ８２によって制御される。ＩＳＤＢ８２が再開コマンドの実行に起因してスレッドをリリースすると、スレッドは、現在のモード設定に従って動作を再開することができる。スレッドがデバッグモード１２０にあるときには、ＩＳＤＢ８２によって制御され、その他のスレッドによって制御することはできない。デバッグモード１２０におけるスレッドをターゲットにした、実行中のスレッドからの待機命令、再開命令、開始命令、又は停止命令は、無視することができる。同様に、マスク不可割り込み（ＮＭＩ）は、デバッグモード１２０にあるスレッドは無視することができる。

【００３９】

ハードウェアリセット（ＨＡＲＤＷＡＲＥ　ＲＥＳＥＴ）モード（示されていない）及びデバッグモード１２０は、全スレッドに対してグローバルである。ハードウェアリセットピンがアサートされるごとに、あらゆるスレッドがどのような処理状態であるかにかかわらず、ＤＳＰ４０は、ハードウェアリセットモードになることができる。リセットモードにおいては、全レジスタが各々のリセット値に設定される。どのような処理も、ハードウェアリセットピンがデアサートされるまで行うことができない。リセットピンがアサートされた時点で、プロセッサは、リセットモードに移行することができ、全レジスタをリセット値にリセットすることができる。リセットピンがデアサートされた後は、スレッドＴ０にソフトリセット割り込みを与えることができる。このことは、スレッドＴ０にスーパバイザモード１１４に入らせてリセットされたベクトル位置での実行を開始させることができる。その他の全スレッドは、オフ状態であることができる。この時点においては、ソフトウェアは、各スレッドに関するモード移行を個々に自由に制御することができる。

【００４０】

各スレッドは、そのスレッドに関してＯＦＦモード１１８への及びＯＦＦモード１１８からの移行を制御する１つの移行ビットをモード制御レジスタ１２２において有することができる。停止命令を介して移行ビットに書き込むことは、関連づけられたスレッドをＯＦＦにする。開始命令を介して移行ビットに書き込むことは、スレッドをオンにし、ソフトリセット割り込みをトリガーする。各スレッドは、待機モード１１６への及び待機モード１１６からの移行を制御するための１つの待機ビットをモード制御レジスタ１２２に含むことができる。待機命令を介して待機ビットに書き込むことは、関連づけられたスレッドをアイドル状態にすることができ、他方、再開命令を介しての書き込みは、待機モード

１１６が設定される前に行っていたあらゆることをスレッドに再開させることができる。

【００４１】

ブレークポイントを用いることで、ＤＳＰ４０の６つのスレッドは、個々にデバッグモード１２０に入ること及びデバッグモード１２０から出ることができる。ブレークポイントトリガーは、ＩＳＤＢ８２においてサポートされる５つの異なる種類のブレークポイントに対応する５つのソースから来ることができる。これらのブレークポイントは、ハードウェアブレークポイントと、ソフトウェアブレークポイントと、ＥＴＭブレークポイントと、ＪＴＡＧインタフェースブレークポイントと、外部ブレークポイントと、を含む。スレッドは、ブレークポイントをヒットした時点で、現行モード（例えば、待機／実行（ＲＵＮ））からデバッグモード１２０に移行する。デバッグモード１２０において、スレッドは、ＩＳＤＢ８２からのコマンドを待つ。ＯＦＦモード１１８にあるスレッドはパワーダウンされ、ＩＳＤＢ８２からのコマンドを受け入れることができない。デバッグモード１２０に入るレーテンシーは、実装によって定義される。例えば、本開示においては、イベント電力崩壊に関連するとして定義される。例えば、一実装は、デバッグモード１２０に入る前に、所定の動作を完了させる、例えば未解決のロード要求を終了させる、ことを選択することができる。一実施形態においては、スレッド識別子レジスタは、８ビットの読み取り／書き込みフィールドを含み、ソフトウェアスレッド識別子を保持するために用いられる。このフィールドは、ブレークポイントとマッチさせるためにハードウェアデバッガによって用いられる。

【００４２】

ブレークポイントプロセスに入る方法は、幾つかの異なる方法が存在する。例えば、２つのハードウェアブレークポイントが存在する。レジスタが予め決められた値と等しい場合は、プログラムカウンタ（ＰＣ）が予め決められた値にマッチするときに、プロセスがデバッグモード１２０になる。ＰＣに加えて、その他の修飾子、例えば、アドレス翻訳（物理アドレス又は仮想アドレス）等のスレッドＩＤ、が存在することができる。ＡＳＩＤは、プロセスにおけるプロセスＩＤ又はマルチスレッド化プロセスにおける特定のスレッドと類似のタグである。このため、物理アドレス、仮想アドレス、ＡＳＩＤ、ＰＣ、又はその他の修飾子を用いて、プロセス空間内のプログラムの位置のフィックスを随意に入手することができる。

【００４３】

ＩＳＤＢ８２は、２つの出力割り込みピンも定義する。これらの信号は、ＩＳＤＢ８２を出てＭＳＭ１０４内に入り、ＭＳＭレベルでストラップされる。これらの２つの信号は、ブレークイベント及びＪＴＡＧインタフェース８４コマンドである。ブレークイベントコマンド時においては、ＩＳＤＢ８２は、示されたスレッド番号においてブレークポイントが生じるごとにこの割り込みを呼び出すようにプログラミングすることができる。ＪＴＡＧインタフェース８４コマンド時においては、ＪＴＡＧインタフェース８４は、この割り込みを呼び出すためのコマンドを送る。

【００４４】

ハードウェアブレークポイントは、スレッドプログラムカウンタ、ＡＳＩＤ（アドレス空間識別子）、及びスレッド識別子レジスタのうちの１以上をＩＳＤＢによってプログラミングされた値とマッチさせる。マッチ条件が満たされているときには、スレッドは、デバッグモード１２０に入る。ブレークポイントをヒットしたスレッドに加えて、その他のスレッドもデバッグモード１２０に入るように構成することができる。このことは、例えばブレークポイントコンフィギュレーションレジスタプログラミングを通じて達成させることができる。

【００４５】

ハードウェアブレークポイントは、様々な特長、例えば、物理的又は仮想であることができる３２ビットプログラムカウンタ値をマッチさせる、６ビットＡＳＩＤ値をマッチさせる、８ビットスレッド識別子値をマッチさせる、及び／又はブレークポイントをヒットした時点で強制的にその他のスレッドをデバッグモード１２０に入らせる、等、をサポー

トすることができる。ハードウェアブレークポイントを設定するために、ブレークポイントプログラムカウンタ及びブレークポイントコンフィギュレーションレジスタは、ＪＴＡＧインタフェース８４を通じて、そして次にＩＳＤＢ８２コンフィギュレーションレジスタを介してイネーブル及び構成されたブレークポイントを用いて設定することができる。

【００４６】

開示される主題は、一定のソフトウェアブレークポイントも提供する。例えば、ユーザーレベルブレークポイント命令を用いてハードウェアデバッグモード１２０に入ることができる。この命令が実行されると、コアは、システムコンフィギュレーションＩＳＤＢｔｒｕｓｔｅｄビットを検討する。ＩＳＤＢｔｒｕｓｔｅｄビットがセットされた場合は、スレッドはデバッグモード１２０に入ることができる。ＩＳＤＢｔｒｕｓｔｅｄがクリアされるか又はＩＳＤＢがディスエーブルにされたときには、ブレークポイント命令の実行はＮＯＰとして処理することができる。ブレークポイント命令のプログラムカウンタアドレスに対する制限は存在しない。しかしながら、ブレークポイント命令は、その他の命令とパケット化することができない。

【００４７】

開示される主題は、ＤＳＰ４０コアプロセッサの動作をモニタリングするＥＴＭプロセスを開始するための埋め込まれたトレースマクロ又はＥＴＭブレークポイントにも対処する。ＥＴＭは、非常に様々なトリガー条件をサポートする。従って、ブレークポイントコンフィギュレーションが該移行に関して設定されたときにハードウェアブレークポイントをＥＴＭブレークポイントにリンクすることが生じることができる。ＩＳＤＢ８２は、ＥＴＭ（埋め込まれたトレースマップ）の使用を通じて、プロセッサの動作をモニタリングすることを目的としてプロセッサに隣接するプロセッサハードウェアのセクションを用いる能力を提供する。さらに、開示される主題は、１つ以上のスレッドにおけるデバッギング動作を、１つ以上のその他のスレッドにおける動作にリンクする能力を提供する。例えば、１つのハードウェアスレッドがブレークポイントをヒットした場合は、本開示は、他のスレッドにおける処理を開始又は停止することを許可する。従って、本開示は、いずれかの１つのスレッド又は一組のスレッドを独立してデバッグすること、又は１つのスレッド又は一組のスレッドにおいて生じているイベントが１つのスレッド又は一組のスレッドにおける動作に対してどのような影響を与えるかを制御する能力に対処する。

【００４８】

従って、開示される主題は、デバッグモード１２０に入らせるブレークポイントの場合にデバッグモード１２０に移行するための経路を提供する。開示される主題は、マルチスレッド化デジタル信号プロセッサ内のいずれのスレッド又はスレッドの組がデバッグ１２０に入るかを制御する。ＥＴＭブレークポイントデバッギングは、プロセッサデバッギングに関して用いることができる性能プロファイリング等の動作行う。そのブロックは、デバッギングプロセスに入るためのブレークポイントを提供することができる。

【００４９】

この状況においては、ハードウェアブレークポイント及びＥＴＭブレークポイントスレッド番号ＭＡＳＫの両方が、マッチしているスレッドに関してイネーブルにされる。この例においては、ＤＳＰ４０は、ハードウェアブレークポイントがＥＴＭブレークポイントに引き続いてトリガーされるときのみにデバッグモード１２０に切り換わることができる。ＥＴＭブレークポイントが生じる前に生じるハードウェアブレークポイントトリガーは、無視することができる。ブレークポイントコンフィギュレーションが“０”に設定されるか又は設定されないときには、ハードウェアブレークポイント及びＥＴＭブレークポイントスレッド番号ＭＡＳは通常どおりに挙動する。すなわち、イネーブルにされたときに、対応するブレークポイントトリガーは、スレッドを直ちにデバッグモード１２０に切り換えさせることができる。

【００５０】

ＪＴＡＧインタフェース８４ブレークポイントは、ＩＳＤＢブレークコマンドにおいてトリガーされ、従ってコマンドマスクにおいて示されるスレッドがデバッグモード１２０

に入ることができる。ＩＳＤＢ８２は、外部ブレークポイントを通じてマルチコアデバッグもサポートする。該ブレークポイントは、外部デバッグ要求信号において立ち上がりエッジが検出されたときにトリガーされる。このイベント時において、外部ブレークポイントスレッド番号マスクにおいて示される全スレッドがデバッグモード１２０に入ることができる。

【００５１】

開示される主題の他の特長は、“命令スタッフィング”と呼ばれる。命令スタッフィングは、ホストデバッギングプロセスがコアの状態を検査しようとするときに生じる。従って、ブレークポイントが生じたときには、プロセスは、コアを検討してコアにおいて動作が発生しているかどうかを決定することを試みる。この決定を行う仕組みは、デバッグ動作モード１２０に入りつつあるスレッドにおいてプロセッサ命令を実行することを目的としてそのプロセッサ命令を送ることである。

【００５２】

命令スタッフィング動作においては、命令は、デバッグモード１２０時にすべての影響を受けているレジスタ全部又は一部分を読み取るようにプロセッサに指示することができる。さらに、デバッグモード１２０は、予め決められた組の又は型の命令をロードするようにプロセッサに指示することができる。状態を読み取る又は書き込むことに加えて、基本的にはいずれの命令も、このプロセスにおいて読み取ること又はコアに書き込むことができる。例えば、何らかのアルゴリズム又はプロセスをプロセスコアにおいて実行するのが望ましい場合は、開示される主題は、一組のオプションを許容する。命令スタッフィングプロセスにおいては、記憶場所に分岐し、プロセスは、動作のために符号をリリースすることができる。該符号は、例えば、規定された一組の理由で一定の関数を実行するための符号を含むことが可能である。１つの該理由は、複雑なデータ構造を処理するためであることができる。命令が、所定のデータ構造の全要素を読み出すことである場合は、プロセスは、そのデータ構造を再構築することであることができる。該プロセスは、極端に難しい可能性がある。データ構造を読み出すための一組の命令を備えることで、プロセスは、その命令の組を呼び出して特定の命令を入手することであることができ、特定の命令は、希望される要素（例えば、要素１２）に従って実行することが可能である。このことは、多くの型のデータ取り出し及び同様の動作を有意に単純化する。

【００５３】

命令スタッフィングは、ＩＳＤＢ８２がコアにおいて命令を実行する方法である。命令は、様々な理由で、例えば、コアレジスタ及びメモリを読み取る及び書き込むために、ユーザーに関して要約されたデバッガ動作のために、及びユーザーによって入力された命令に関して、スタッフィングされる。命令をスタッフィングするためには、ユーザーは、最初に、実行対象となる３２ビット命令を用いてスタッフ命令レジスタをプログラミングしなければならない。命令スタッフィングに関して、ＩＳＤＢコマンドレジスタは、最初にコマンドフィールドをスタッフ符号に設定し、次にスレッド番号フィールドを命令を受け取るスレッドに設定することによって書き込むことができる。選択されたスレッドは、命令をスタッフィングできるようになる前にデバッグモード１２０であることができる。スレッド番号内の２つ以上のビットが設定されるか又は選択されたスレッドがデバッグモード１２０にない場合は、結果は未定義である。次に、スタッフィングされた命令（ユーザー又はスーパバイザ）の特権レベルの設定を含む段階が生じる。

【００５４】

スタッフコマンドを発行後は、選択された特権レベルを有する選択されたスレッドにおいて命令を実行することができる。命令スタッフィング中には、プログラムカウンタは数字が増えない。プログラムカウンタを用いるスタッフィングされた命令（分岐、又は例外を生じさせる命令）は、スレッドの現在のプログラムカウンタ値を用いることができる。スタッフィングされた命令が例外を生じさせる場合は、ＩＳＤＢ状態レジスタは、例外が発生したことを示すことができる。スレッドは、デバッグモード１２０にとどまることができる。スレッドの設計されたレジスタは、例外状態を反映させることができる。好まし

いことに、ＩＳＤＢ８２デバッギングソフトウェアは、例外を発生させることができた命令をスタッフィング後にＩＳＤＢ状態レジスタに問い合わせて例外が発生したかどうかを確認する。

【００５５】

例外が認識された時点においては、ここにおいて開示されるプロセスは、状況をどのように取り扱うかに関する幾つかの選択肢を含む。例えば、デバッガソフトウェアは、例外リターンポイントにおいてソフトウェア又はハードウェアブレークポイントをプログラミングすることを選択し、ハンドラーを実行するためにスレッドを再開することができる。次に、デバッガは、ＯＳ“ｈｅｌｐｅｒ”関数にスレッドをリダイレクションすることができる。次に、単一ステップを用いてハンドラーを実行することおよび手作業で問題を解決する（例えば、ＴＬＢを再ロードする）ことを行うことができる。しかしながら、具体的な戦略は、ＯＳ及び／又はソフトウェアデバッガの実装により異なることができる。

【００５６】

レジスタ、キャッシュ、及びメモリは、適切な命令シーケンスをスタッフィングすることによってアクセスすることができる。命令に関するステップのシーケンスは、ＩＳＤＢデバッギングアルゴリズムを用いてレジスタ及びキャッシュを読み取る／書き込むことを含むことができる。デバッガソフトウェアは、コアレジスタとＩＳＤＢメールボックスとの間でデータを移動させるための適切な制御レジスタ転送命令をスタッフィングすることによってスレッドレジスタを読み取る／書き込むことができる。この命令は、例外が発生しないようにするためにスーパバイザ特権レベルを用いてスタッフィングすることができる。

【００５７】

再開コマンドは、デバッグからコアモード制御レジスタにおいてプログラミングされたモードにスレッドを移行させるために用いられる。再開方法は、ＪＴＡＧインタフェース８４コマンド又は外部信号のいずれかから再開させる2つの方法がある。ＪＴＡＧインタフェース８４コマンドからの再開である場合は、コマンドマスクにおいて示された、デバッグモード１２０にあるスレッドが終了し、モード制御レジスタにおいて示されるモードになる。外部信号からの再開である場合は、外部再開スレッド番号ＭＡＳＫにおいて示され、デバッグモード１２０にあるスレッドが、モード制御レジスタにおいて示されるモードに移行する。

【００５８】

ＩＳＤＢリセットコマンドを実行することは、ハードウェアリセットを強制し、ＤＳＰ全体（全スレッド）にリセットさせる。このことは、全レジスタを初期値、例えば電力オフスレッド１乃至５に設定し、リセット割り込みをスレッドＴ０に送る。一定のスレッドのみをリセットすることが希望される場合は、適切なマスク設定を用いて最初に開始命令をスタッフィングする手順で行うことができる。このことは、示されたスレッドに対してリセット割り込みを保留状態にすることができる。次に、プロセスは、希望されるスレッドにおいてＩＳＤＢ再開命令を実行することを含む。

【００５９】

他の型のブレークポイントは、ホストがプロセッサにコマンドを送ってブレークを指示するＪＴＡＧインタフェース８４ブレークポイントである。基本的に入り外部ピンが存在する。この実施形態においては、ＩＳＤＢ８２制御レジスタは、ＪＴＡＧインタフェース８４を介してデバッガホストソフトウェアによってアクセスすることができる。ＩＳＤＢ８２は、異なるデバッグタスクを実行し、ＤＳＰ４０コアプロセッサと通信するようにＩＳＤＢ８２を構成するためにホストシステムによって用いることができる様々な制御レジスタを提供する。例えば、ＩＳＤＢ８２状態レジスタは、ＩＳＤＢ８２の現在の状態を示す。ＩＳＤＢ８２状態レジスタのビットは、いずれのスレッドが“待機”対実行モードにあるかを示し、その他は、いずれのスレッドが“ＯＦＦ”モードであるかを示す。これらは、コアモード制御レジスタのＥビットフィールドを反映したものである。例えばＯＦＦであるスレッドは、一般的にはデバッグできない。従って、ＩＳＤＢコマンドがオフであ

るスレッドに送られる場合は、結果は未定義である。その他のデバッグモード１２０状態ビットは、いずれのスレッドがデバッグモード１２０であるかを示す。スレッドがデバッグモード１２０にあることをこれらのビットが示す場合は、待機／実行モードビットは、デバッグモード１２０に入る前にモードを示す。

【００６０】

さらにその他のビットは、スタッフコマンド状態、すなわちスタッフ命令プロセスが成功しているかどうか又はスタッフ命令が例外を発生させたかどうか、を示す。ＩＳＤＢコマンド状態ビットは、ＩＳＤＢコマンドが成功又は失敗であったかを表す。他の組のビットは、いずれかのスレッドがデバッグ１２０モードにあるときにグローバル割り込みディスエーブルを提供し、従って、デバッグモード１２０にあるスレッドに関しては割り込みがディスエーブルにされ、その他のスレッドに関してはイネーブルにされる。いずれかのスレッドがデバッグモード１２０にあるときには全スレッドに関して割り込みがディスエーブルにされる。

【００６１】

その他のビットは、外部再開信号に基づいていずれのスレッドが再開すべきかを示すフィールドを形成することができる。外部再開信号に基づき、マスクビットが設定されているスレッドに関して、そのスレッドがデバッグモード１２０にある場合は、前モードを再開することができ、その他の場合は影響がない。他のフィールドは、外部ブレークポイント要求時にいずれのスレッドがブレークすべきかを示す。マスクビットが設定されているスレッドに関して、ＩＳＤＢ８２において外部ブレークポイント信号を受信した時点で、そのスレッドがデバッグモード１２０ない場合は、デバッグモードに入ることができ、その他の場合は影響がない。

【００６２】

さらに、ＩＳＤＢコンフィギュレーション命令は、ＩＳＤＢ８２の様々な特長をイネーブル又はディスエーブルにすることができる。いずれかのスレッドがデバッグモード１２０にあるときにグローバル割り込みディスエーブルが発生し、それによりデバッグモード１２０にあるスレッドに関する割り込みをディスエーブルにする。ＩＳＤＢコンフィギュレーションレジスタ内の他のフィールドは、外部再開信号に基づいていずれのスレッドが再開すべきかを示すことができる。外部再開信号に基づき、マスクビットが設定されているスレッドに関して、該スレッドがデバッグモード１２０にある場合は、該スレッドは前モードを再開する。その他の場合は影響がない。

【００６３】

ＩＳＤＢレジスタ内のさらに他のフィールドは、ＩＳＤＢ８２が外部ブレークポイント要求を受け取った時点でいずれのスレッドがブレークすべきかを示すことができる。外部ブレークポイント信号を受け取った時点で、マスクビットが設定されているスレッドに関して、スレッドがデバッグモード１２０にない場合は、デバッグモード１２０に入ることができる。その他の場合は影響がない。

【００６４】

ブレークポイント情報レジスタは、デバッグモード１２０にあるスレッドに関して、いずれのトリガーがブレークポイントを発生させたかを示す。これは、ブレークポイント命令実行時にいずれの追加のスレッドがブレークすべきかを示す６ビットフィールドであることができる。最下位ビットは、スレッド番号０に関するビットであることができ、次のビットはスレッド番号１に関するビットであり、以下同様である。ブレークポイント命令実行時において、ブレークポイントを実行したスレッドは、デバッグモード１２０に入ることができる。さらに、このマスクにおいてビットが設定されているスレッドは、デバッグモード１２０に入ることができる。

【００６５】

ＩＳＤＢ８２からＭＳＭに進む割り込み信号ブレークイベントが存在する。このマスクにおいて示されるスレッド番号がデバッグモード１２０に入るごとに、ブレークイベント割り込みが呼び出される。一実施形態においては、ビット０は、スレッド番号０に関する

ビットであり、ビット１はスレッド番号１に関するビットであり、以下同様である。デバッグモード１２０にあるスレッドに関して、これらのビットは、何がデバッグモード１２０への移行を生じさせたかを示す。デバッグモード１２０にないスレッドに関しては、これらのビットは未定義である。従って、ビットは、ハードウェアブレークポイント、ブレークポイント命令実行、ＥＴＭブレークポイント、ＪＴＡＧインタフェース８４ブレークポイント、外部ブレークポイントの存在を示す。さらに、その他のビットは、ブレークポイントソースを示すことができる。ブレークポイントプログラムカウンタは、ブレークポイントプログラムカウンタと同一のレジスタを含み、これらは、ハードウェアブレークポイントを制御することを除く。ブレークポイントコンフィギュレーションレジスタは、スレッドのプログラムカウンタレジスタと比較するために用いられる。

【００６６】

ＩＳＤＢ８２コマンドレジスタは、いずれのスレッドがデバッグモード１２０に移行できるかを示すためのブレークコマンドを含むことができる。再開コマンドに関して、いずれのスレッドが再開すべきかを示す。ＩＳＴＥＰコマンドは、いずれのスレッドがステップ・バイ・ステッププロセスにおいてステップすべきかを示す。スタッフ命令コマンドは、いずれのスレッドが命令スタッフィング動作を行うためのスタッフ命令を受け取ることができるかを示す。スタッフ命令特権は、幾つかのスタッフィングされた命令がユーザーモードにおいて実行するのを許容し、その他は、スーパバイザモードにおいて実行することができる。

【００６７】

ブレークコマンドに基づき、ＤＳＰ４０は、スレッド番号マスクにおいて示される全スレッドをデバッグモード１２０に移行させることができる。再開コマンドは、スレッド番号マスクにおいて示される全スレッドを実行モードに移行させることをプロセッサに行わせる。ステップコマンドは、デジタル信号プロセッサが１つのパケットに関するスレッド番号マスク内において示される全スレッドをステップさせることを許容する。示されたスレッドがデバッグモード１２０にない場合は影響がない。

【００６８】

スタッフコマンドは、スレッド番号マスクにおいて示されるスレッドにおけるスタッフ命令レジスタ内に含まれる３２ビット命令を実行することをデジタル信号プロセッサに行わせる。マスク内の１つのビットのみをセットすることができる。示されるスレッドがデバッグモード１２０にない場合は、挙動は未定義である。リセットコマンドは、ＤＳＰへのハードウェアリセットを開始する。レジスタは、初期値に設定され、スレッド１乃至５がオフにされ、スレッドＴ０は、リセット割り込みが与えられる。割り込みコマンドに基づき、ＩＳＤＢ８２は、ＪＴＡＧインタフェース８４コマンド割り込みを呼び出す。この信号は、ＩＳＤＢ８を出てＭＳＭ１０４内に入り、ＭＳＭレベルでストラップされる。ＩＳＤＢ８２イネーブルレジスタは、ＩＳＤＢ８２の動作をイネーブルにし、さらに、“セキュリティ”ＩＳＤＢ８２イネーブルビット及びＩＳＤＢ８２クロックの状態を検査する。

【００６９】

ＩＳＤＢ８２の動作をサポートする様々なコマンドについて説明したが、ＩＳＤＢ８２デバッギング動作の典型的プロセスは、さらに有益であることができる。従って、図６は、本開示の様々な非侵入型デバッギングアルゴリズムの側面に関するＩＳＤＢ８２流れ図を示す。図６のＩＳＤＢ８２プロセスの流れは、様々な手法を用いて実行することができるが、提示される開示主題の基本的な流れは、希望される非侵入型デバッギング動作を達成する。従って、図６に関して、ＪＴＡＧインタフェース８４から、ＩＳＤＢ入口ステップ１３０において、プロセスの流れが開始することができる。

【００７０】

ＩＳＤＢ入口ステップ１３０から、非侵入型デバッギングプロセスの流れは、ＩＳＤＢがＤＳＰ４０動作に関してイネーブルにされているかどうかを試験する、ＩＳＤＢによってイネーブルにされるクエリ１３２に進む。ＩＳＤＢがＤＳＰ４０動作に関してイネーブ

ルにされている場合は、プロセスの流れは、ハードウェアブレークポイントクエリ１３４に進む。ハードウェアブレークポイントクエリ１３４は、ハードウェアブレークポイントに遭遇しているかどうかを試験する。遭遇していない場合は、プロセスの流れは、ソフトウェアブレークポイントクエリ１３６まで続くことができる。その他の場合は、プロセスの流れは、デバッギング動作が開始するデバッギング動作ステップ１３８に進む。ソフトウェアブレークポイント１３６は、ソフトウェアブレークポイントの存在の有無を試験し、ソフトウェアブレークポイントが存在する場合はＩＳＤＢ８２プロセスをデバッギング動作ステップ１３８に向ける。ソフトウェアブレークポイントが存在しない場合は、プロセスの流れは、ＥＴＭブレークポイントクエリ１４０まで続く。ＥＴＭブレークポイントクエリ１４０は、ＥＴＭブレークポイントの存在の有無を試験し、ＥＴＭブレークポイントが存在する場合はＩＳＤＢ８２プロセスをデバッギング動作ステップ１３８に向ける。ＥＴＭブレークポイントが存在しない場合は、プロセスの流れは、ＪＴＡＧインタフェース８４ブレークポイントクエリ１４２まで続く。ＪＴＡＧインタフェース８４ブレークポイント１４２は、ＪＴＡＧ８４ブレークポイントの存在の有無を試験し、ＪＴＡＢブレークポイントが存在する場合は、ＩＳＤＢ８２プロセスをデバッギング動作ステップ１３８に向かわせる。ＪＴＡＢブレークポイントが存在しない場合は、プロセスの流れは、外部ブレークポイントクエリ１４４まで続く。外部ブレークポイント１４４は、外部ブレークポイントの存在の有無を試験し、外部ブレークポイントの存在する場合は、ＩＳＤＢ８２プロセスをデバッギング動作ステップ１３８に向かわせる。外部ブレークポイントが存在しない場合は、プロセスの流れは、ＩＳＤＢによってイネーブルにされるクエリ１３２に戻る。このタイプのサイクルは、ＤＳＰ４０の動作中に繰り返すことができる。

【００７１】

ＩＳＤＢ８２プロセスの流れがデバッギング動作ステップ１３８に進んだ時点で、“デバッグ待機”クエリ１４６は、待機モード１１６が有効かどうか試験する。有効である場合は、待機１１６モードが終了するまでデバッギング動作は行われない。待機モード１１６が有効でない場合は、プロセスの流れは、ＩＳＴＥＰデバッギングクエリ１４８に進む。ＩＳＴＥＰデバッギングクエリ１４８は、個々のステップデバッギングがＩＳＤＢ８２動作に関して有効であるかどうかを試験する。有効である場合は、プロセスの流れは、ＩＳＴＥＰデバッギングステップ１５０に進み、このタイプのデバッギング動作を実行する。ＩＳＴＥＰデバッギングが有効でない場合は、プロセスの流れは、スタッフ命令クエリ１５２に進む。スタッフ命令クエリ１５２は、命令スタッフィング動作がＩＳＤＢ８２動作に関して有効であるかどうかを試験する。有効である場合は、プロセスの流れは、ここにおいて説明される命令スタッフィング動作を表すスタッフ命令ステップ１５４に進むことができる。命令スタッフィングが有効でない場合は、プロセスの流れは、クエリ１５６に進む。

【００７２】

クエリ１５６において、コアＤＳＰ４０リセット命令がデバッギング動作によって生成されているかどうかに関する試験が実施される。コアＤＳＰ４０リセット命令が生成されている場合は、プロセスの流れは、コアＤＳＰ４０デジタル信号プロセッサリセットコマンドを引き渡すためにＪＴＡＧインタフェース８４に進む。該リセットコマンドが生成されていない場合は、プロセスの流れは、割り込み存在クエリ１５８に進む。割り込み存在クエリ１５８は、デバッギング動作への割り込みが存在するかどうかを試験する。デバッギング動作への割り込みが存在する場合は、ＩＳＤＢ８２動作に割り込まれ、プロセスの流れは、デバッギング動作に割り込まれていることを示す信号をＤＳＰ４０に引き渡すためにＪＴＡＧ４０インタフェース８４に進む。割り込み信号が存在しない場合は、プロセスの流れは、通常のスレッド動作が開始できるかどうか及びデバッギング動作が停止できるかどうかを試験するために通常スレッド再開クエリ１６０に進む。通常のスレッド動作が開始することができ、デバッギング動作が停止することができる場合は、プロセスの流れは、ＪＴＡＧインタフェース８４に進み、ＤＳＰ４０は、対象スレッドをデバッグ動作モード１２０から通常動作モードに移行させる。デバッギングが継続する場合は、プロセ

スの流れは、デバッギング動作ステップ１３８に戻る。
【００７３】
明確なことであるが、ＩＳＤＢ８２プロセスの流れの動作は、非常に広範であることが可能であるが、これらの動作は開示される主題の適用範囲内であることができる。従って、図６のＩＳＤＢ８２プロセスの流れは、本開示の１つの可能な実施形態として例示することを目的とするものである。
【００７４】
開示される主題の他の側面は、ＤＳＰ４０内における電力崩壊を通じてのデバッギングを含む。ＩＳＤＢコンフィギュレーションレジスタは、デバッガソフトウェア（ＪＴＡＧインタフェース８４を介する）及びスーパバイザコアソフトウェア（ＣＲ転送命令を介する）の両方によって読み取り可能及び書き込み可能である。カーネルソフトウェアは、この特長を用いて電力崩壊時におけるＩＳＤＢコンフィギュレーションをセーブ及びリストアすることができる。これらの共有レジスタを書き込むマスターが多数存在するため、これらのレジスタは首尾一貫した、相互に排他的な形でのみ書き込むことが重要である。
【００７５】
本明細書の方針は、コアがパワーダウン又はパワーアップ中のときには、ＪＴＡＧインタフェース８４はこれらのレジスタを読み取る／書き込むことを許容されないということである。同様に、ＪＴＡＧ８４がこれらのレジスタを修正中であるときには、コアは、パワーダウンすることが許容されない。この方針は、ハードウェアとソフトウェアの組合せに関しても強制される。システムコンフィギュレーションにおける１つのビット、ＩＳＤＢコアレディレジスタビットは、コアスーパバイザソフトウェアのみによって書き込むことができる。このビットは、ＤＳＰ４０のハードウェアリセット時にクリアされる。このビットがクリアされているときには、すべてのＪＴＡＧインタフェース８４読み取り及び書き込みパケットは、無効状態を戻すことができる。コアは、このビットを用いて、いつパワーアップシーケンスを完了していてＩＳＤＢと対話する準備が整っているかをホストソフトウェアに示すことができる。このことは、セーブされたＩＳＤＢ８２コンフィギュレーションをウォームブートパワーアップ（リストア）シーケンスにおいてリストアさせる機会をコアに与える。
【００７６】
電力崩壊時におけるデバッギングの一例は、電力を意識する必要がある携帯電話において見つけることができる。ＤＳＰ４０は、依然としてデバッギングを行う必要がある間にオフになるか又はアイドル状態になることができる。従って、開示される主題は、電力崩壊インスタンスのみにおいて現れることができるブレークポイントを設定する能力を提供する。このことは、コアが動作中でない又は“オン”状態でないときでさえもデバッグする能力を提供する。
【００７７】
開示される実施形態における、電力崩壊時のデバッギングは、ＤＳＰが電力を降下させることに関連づけられたコンフィギュレーションに関する一組のブレークポイントを設定することを含む。ＤＳＰが電力を降下させる前に、ＤＳＰは、コンフィギュレーションを特定のレジスタ内にセーブする。これらの特定のレジスタ及びコンフィギュレーションは、ＲＡＭプロセスの中断を可能にする。従って、ＤＳＰが戻ったときに、コンフィギュレーションは、次のデバッグ動作を実行できる状態にある。
【００７８】
ここにおいて、マルチスレッド化デジタル信号デジタル信号プロセッサにおける非侵入型、スレッド選択式デバッギングに関して説明される処理上の特長及び機能は、様々な方法で実装することができる。例えば、ＤＳＰ４０が上述される動作を実行できるだけでなく、本明細書の実施形態は、ここにおいて説明される機能を果たすように設計された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、又はその他の電子回路において実装することもできる。さらに、ここにおいて説明されるプロセス及び特長は、該様々な信号及び命令処理システムによる読み取り及び実行のために磁

気媒体、光学媒体、又はその他の記録媒体に格納することができる。従って、好まれる実施形態に関する上記の説明は、当業者が請求される主題を製造または使用するのを可能にすることを目的とするものである。これらの実施形態に対する様々な修正が当業者にとって容易に明確になるであろう。さらに、ここにおいて定義される一般原理は、革新的な能力を用いずにその他の実施形態にも適用可能である。以上のように、請求される主題は、ここにおいて示される実施形態に限定されることが意図されるものではなく、ここにおいて開示される原理及び斬新な特長に一致する限りにおいて最も広範な適用範囲が認められるべきである。

【図１】

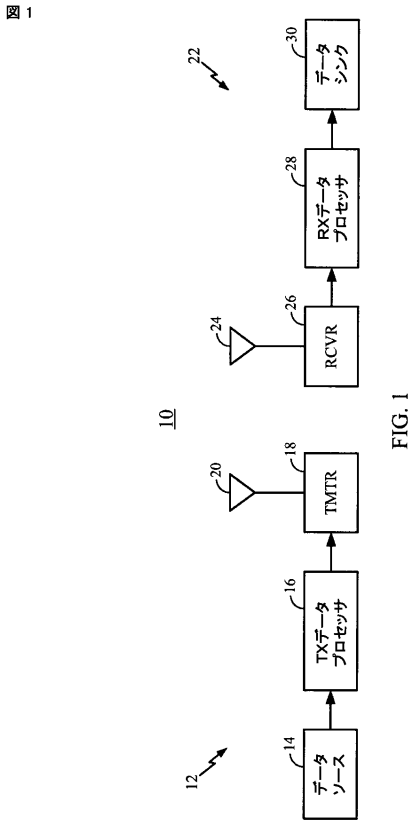

FIG. 1

【図２】

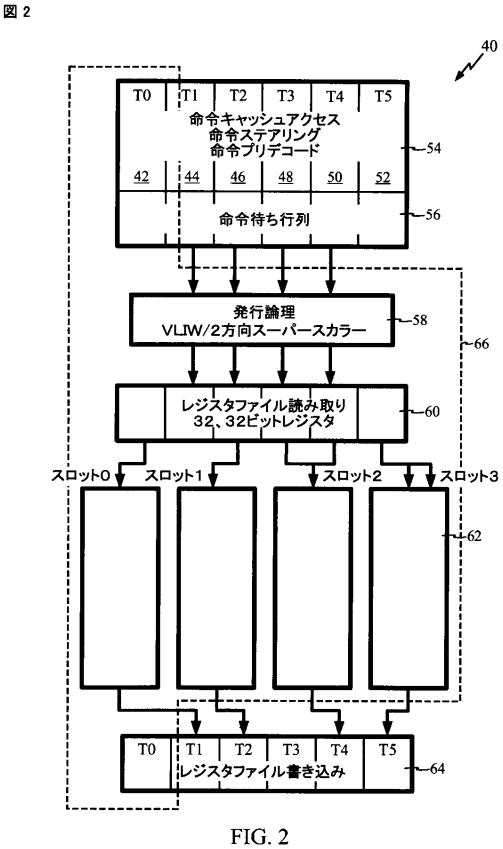

FIG. 2

【図３】

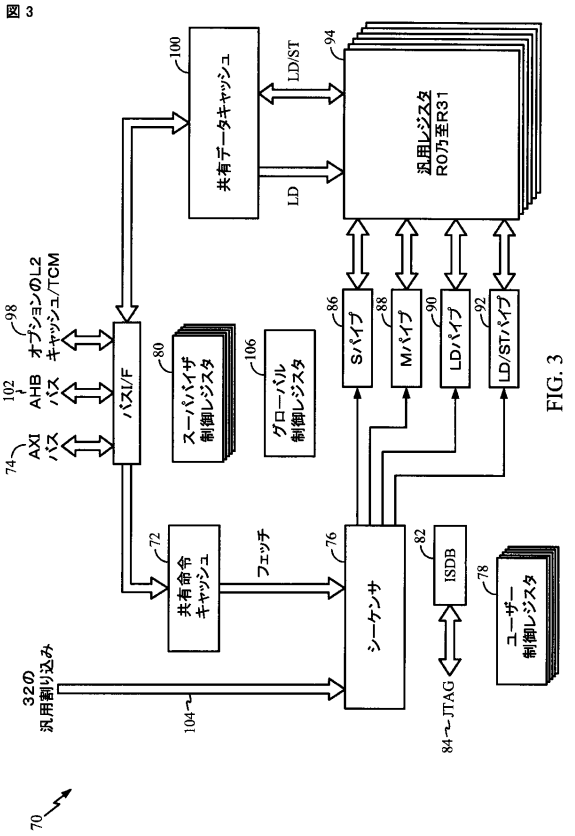

FIG. 3

【図４】

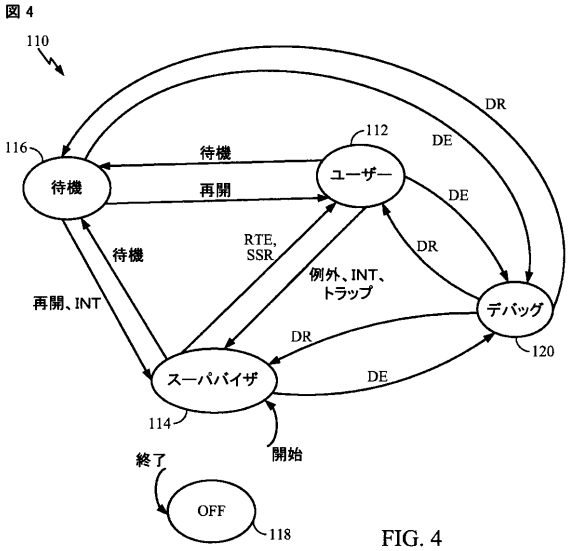

FIG. 4

【図５】

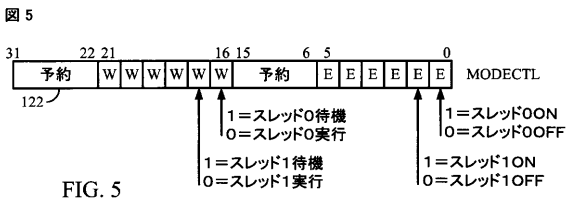

FIG. 5

【図６】

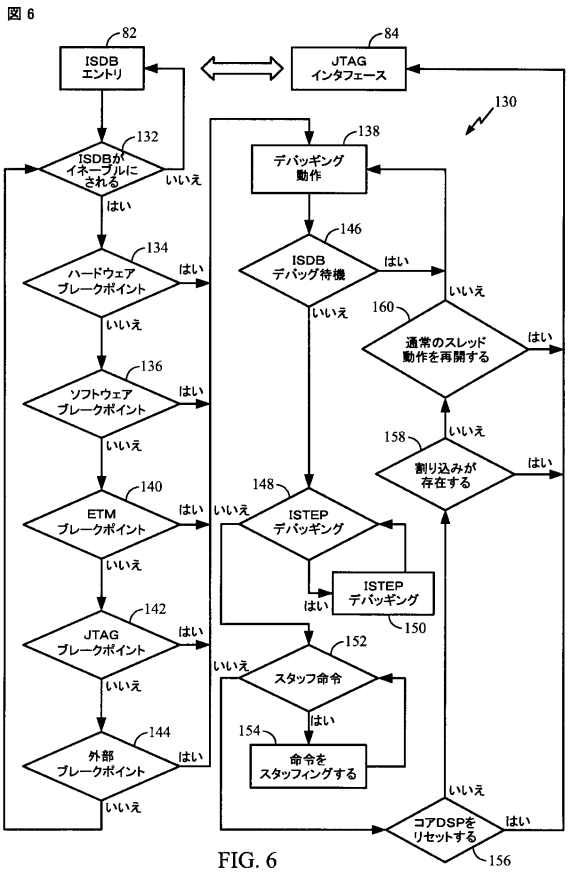

FIG. 6

【手続補正書】
【提出日】平成24年10月24日(2012.10.24)
【手続補正１】
【補正対象書類名】特許請求の範囲
【補正対象項目名】全文
【補正方法】変更
【補正の内容】
【特許請求の範囲】
【請求項１】
マルチスレッド化デジタル信号プロセッサをデバッグするための非侵入型方法であって、
前記マルチスレッド化デジタル信号プロセッサの少なくとも１つ以上のスレッドを用いて複数の処理命令をマルチスレッド化プロセスにおいて実行することと、
少なくとも１つのデバッギングイベントを生成するための１つ以上のブレークポイント命令を識別することと、
前記ブレークポイント命令のうちの少なくとも１つを実行することに応じて前記少なくとも１つのデバッギングイベントを生成することと、
前記少なくとも１つのデバッギングイベントに応じて複数のデバッギング命令を実行することであって、前記デバッギング命令は、前記マルチスレッド化デジタル信号プロセッサの少なくとも１つ以上のスレッドをデバッギングモードに移行させることによって前記マルチスレッド化デジタル信号プロセッサにおいて前記複数の処理命令を前記実行することを非侵入方式でデバッグするためのデバッギング命令であることと、
前記複数のデバッギング命令を前記実行することを報告するために前記複数のデバッギング命令を前記実行することからの少なくとも１つのデバッギングリターンを生成すること、とを備える、非侵入型方法。

フロントページの続き

(74)代理人 100095441
弁理士 白根 俊郎
(74)代理人 100075672
弁理士 峰 隆司
(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 100158805
弁理士 井関 守三
(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
(72)発明者 ルシアン・コドレスキュ
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 ウィリアム・シー．・アンダーソン
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
(72)発明者 スレッシュ・ベンクマハンティ
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 ルイス・アチレ・ジアンニニ
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 マノジクマー・パイラ
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 スーフェン・チェン
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) 5B042 GA23 HH25 HH50 LA10

【外国語明細書】
2013058207000001.pdf