

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3920994号
(P3920994)

(45) 発行日 平成19年5月30日(2007.5.30)

(24) 登録日 平成19年2月23日(2007.2.23)

(51) Int.C1.

F 1

G O 6 F 15/16 (2006.01)

G O 6 F 15/16 6 2 O G

請求項の数 11 (全 25 頁)

(21) 出願番号 特願平10-259364
 (22) 出願日 平成10年8月28日(1998.8.28)
 (65) 公開番号 特開平11-154144
 (43) 公開日 平成11年6月8日(1999.6.8)
 審査請求日 平成17年8月26日(2005.8.26)
 (31) 優先権主張番号 08/924,518
 (32) 優先日 平成9年9月5日(1997.9.5)
 (33) 優先権主張国 米国(US)

(73) 特許権者 504199127
 フリースケール セミコンダクター イン
 コーポレイテッド
 アメリカ合衆国 78735 テキサス州
 オースティン ウィリアム キャノン
 ドライブ ウエスト 6501
 (74) 代理人 100083574
 弁理士 池内 義明
 (74) 代理人 100116322
 弁理士 桑垣 衛
 (72) 発明者 ウィリアム・シー・モイヤー
 アメリカ合衆国テキサス州78620、ド
 リッピング・スプリングス、パイア・ブ
 ランチ・ロード 1005

最終頁に続く

(54) 【発明の名称】プロセッサをコプロセッサにインターフェースするための方法および装置

(57) 【特許請求の範囲】

【請求項1】

コプロセッサバス(30)を介して結合されたコプロセッサ(14, 16)と協働する
 よう構成されたプロセッサ(12)を動作させる方法であって、前記コプロセッサバスは
 システムバスから分離されておりかつ前記プロセッサは前記システムバスを使用すること
 なく前記コプロセッサバスを介して前記コプロセッサと通信することが可能であり、前記
 プロセッサ(12)は前記コプロセッサ(14, 16)と、前記コプロセッサに転送される
 べきアーギュメントの数を規定するカウントフィールドおよびコードフィールドを備えた
 少なくとも1つの命令(H_C A L L)の実行において協働し、前記方法は、

前記命令を受ける段階、

前記コプロセッサバスにより第1のサイクルを介して、前記コプロセッサに前記カウント
 およびコードフィールドを提供する段階、もし前記カウントフィールドがゼロより大きなある値、n、を有していれば、前記コプロ
 セッサに、前記コプロセッサバスによって第2のサイクルを介して、第1のオペランド
 を提供する段階、そして

前記命令を完了する段階、

を具備することを特徴とするプロセッサ(12)を動作させる方法。

【請求項2】

コプロセッサバス(30)を介して結合されたコプロセッサ(14, 16)と協働する
 よう構成されたプロセッサ(12)を動作させる方法であって、前記コプロセッサバスは

10

20

システムバスから分離されておりかつ前記プロセッサは前記システムバスを使用することなく前記コプロセッサバスを介して前記コプロセッサと通信可能であり、前記プロセッサ(12)は前記コプロセッサ(14, 16)と、前記コプロセッサから受信すべきアーギュメントの数を規定するカウントフィールドおよびコードフィールドを備えた少なくとも1つの命令(H_RST)の実行において協働し、前記方法は、

前記命令を受ける段階、

前記コプロセッサバスにより第1のサイクルを介して、前記コプロセッサに前記カウントおよびコードフィールドを提供する段階、

もし前記カウントフィールドがゼロより大きなある値、n、を有していれば、前記コプロセッサバスによって第2のサイクルを介して、前記コプロセッサから第1のオペランドを受ける段階、そして

前記命令を完了する段階、

を具備することを特徴とするプロセッサ(12)を動作させる方法。

【請求項3】

複数のレジスタを備えたレジスタファイルを有する、プロセッサによって、コプロセッサ通信バスを介して、前記レジスタファイルにおける全ての書き込み処理を放送する方法であって、

前記レジスタファイルに書き込まれるべきオペランドを受ける段階、

前記レジスタファイルにおける前記複数のレジスタの内の1つを選択する段階、そして

前記レジスタファイルにかつ前記コプロセッサ通信バスを介して、

前記レジスタファイルに書き込まれるべき前記オペランド、

前記レジスタファイルにおける前記複数のレジスタの内の前記選択された1つを示す第1の制御信号、および

前記オペランドが前記複数のレジスタの内の前記選択された1つに書き込まれることを要求する第2の制御信号、

を提供する段階、

を具備することを特徴とする方法。

【請求項4】

第1のプロセッサによって、コプロセッサ通信バスを介して、前記第1のプロセッサとは別個の第2のプロセッサにおける、複数のレジスタを備えた、レジスタファイルへの全ての書き込み処理を監視する方法であって、

前記第2のプロセッサから、前記コプロセッサ通信バスを介して、

前記レジスタファイルに書き込まれるべきオペランド、

前記レジスタファイルにおける前記複数のレジスタの内の選択された1つを示す第1の制御信号、および

前記オペランドが前記複数のレジスタの内の前記選択された1つに書き込まれることを要求する第2の制御信号、

を受ける段階、そして

前記オペランド、前記第1の制御信号および前記第2の制御信号を受けたことに応じて所定の操作を行なう段階、

を具備することを特徴とする方法。

【請求項5】

プロセッサによって、コプロセッサバスを介して、実行のために前記プロセッサによって受けた命令のコプロセッサによる実行を開始する方法であって、

前記命令を受ける段階、

前記命令をデコードする段階、

少なくとも部分的に前記デコードと一致した時間に、前記コプロセッサに対し、

前記コプロセッサバスの第1の部分を介して、前記命令の少なくとも所定の部分、および

前記命令が前記コプロセッサバスの第2の部分を介して前記プロセッサによりデコ

10

20

30

40

50

ードされていることを示す第1の制御信号、
を提供する段階、

前記命令の実行が進行していることを示すために前記コプロセッサバスに第2の制御信号を提供する段階、そして

もし前記第1の制御信号が前記第2の制御信号が肯定される前に否定されれば、前記命令の処理を中断する段階、

を具備することを特徴とする方法。

【請求項6】

プロセッサによって、実行のために前記プロセッサによって受けた命令のコプロセッサによる実行を開始する方法であって、

10

前記命令を受ける段階、

前記命令をデコードする段階、

少なくとも部分的に前記デコードと一致した時間に、前記コプロセッサに対し、

前記コプロセッサバスの第1の部分を介して、前記命令の少なくとも所定の部分、および

前記命令が前記プロセッサによって前記コプロセッサバスの第2の部分を介してデコードされていることを示す第1の制御信号、

を提供する段階、そして

もし前記命令に先立ち、より前の命令の実行が例外を引き起こせば、前記第1の制御信号を否定しつつ前記命令の処理を中断する段階、

20

を具備することを特徴とする方法。

【請求項7】

プロセッサによって、コプロセッサバスを介して、実行のために前記プロセッサによって受けた命令のコプロセッサによる実行を開始する方法であって、

前記命令を受ける段階、

前記命令をデコードする段階、

少なくとも部分的に前記デコードと一致する時間に、前記コプロセッサに対し、

前記コプロセッサバスの第1の部分を介して、前記命令の少なくとも所定の部分、および

前記命令が前記コプロセッサバスの第2の部分を介して前記プロセッサによりデコードされていることを示す第1の制御信号、

30

を提供する段階、そして

もし前記命令が前記プロセッサ内の命令レジスタから放棄されれば、前記第1の制御信号を否定しつつ前記命令の処理を中断する段階、

を具備することを特徴とする方法。

【請求項8】

プロセッサによって、コプロセッサバスを介して、実行のために前記プロセッサによって受けた命令のコプロセッサによる実行を開始する方法であって、

前記命令を受ける段階、

前記命令をデコードする段階、

40

少なくとも部分的に前記デコードと一致する時間に、前記コプロセッサに対し、

前記コプロセッサバスの第1の部分を介して、前記命令の少なくとも所定の部分、および

前記命令が前記コプロセッサバスの第2の部分を介して前記プロセッサによりデコードされていることを示す第1の制御信号、

を提供する段階、そして

前記コプロセッサバスから第2の制御信号を受ける段階であって、該第2の制御信号は前記コプロセッサが前記コプロセッサ内の命令バッファを充填することを可能にする、段階、

を具備することを特徴とする方法。

50

【請求項 9】

プロセッサによって、コプロセッサバスを介して、実行のために前記プロセッサによって受けた命令のコプロセッサによる実行を開始する方法であって、

前記命令を受ける段階、

前記命令をデコードする段階、そして

少なくとも部分的に前記デコードと一致した時間に、前記コプロセッサに対し、

前記コプロセッサバスの第1の部分を介して、前記命令の少なくとも所定の部分、および

前記コプロセッサバスの第2の部分を介して前記プロセッサがスーパバイザモードで動作しているか否かを示す第1の制御信号、

を提供する段階、

を具備することを特徴とする方法。

【請求項 10】

プロセッサによって、コプロセッサバスを介して、実行のために前記プロセッサによって受けた命令のコプロセッサによる実行を開始する方法であって、

前記命令を受ける段階、

前記命令をデコードする段階、

少なくとも部分的に前記デコードと一致した時間に、前記コプロセッサに対し、

前記コプロセッサバスの第1の部分を介して、前記命令の少なくとも所定の部分、および

前記コプロセッサバスの第2の部分を介して前記命令が前記プロセッサによりデコードされていることを示す第1の制御信号、

を提供する段階、そして

前記コプロセッサから、前記命令の前記所定の部分が前記コプロセッサ内で例外を引き起こしたか否かを示す第2の制御信号を受ける段階、

を具備することを特徴とする方法。

【請求項 11】

コプロセッサによって、コプロセッサバスを介して前記コプロセッサに結合されたプロセッサにより受けた命令に応じて操作を行なう方法であって、

前記プロセッサから、前記コプロセッサバスの第1の部分を介して前記命令の少なくとも所定の部分および、前記コプロセッサバスの第2の部分を介して、前記命令が前記プロセッサによりデコードされていることを示す第1の制御信号を受ける段階、

前記命令の実行を開始する段階、そして

前記プロセッサに対し前記命令が例外を引き起こしたか否かを示す制御信号を提供する段階、

を具備することを特徴とする方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は一般的にはプロセッサおよび少なくとも1つのコプロセッサ (c o p r o c e s s o r) を有するデータ処理システムに関し、かつより特定的には、該コプロセッサに対しプロセッサをインタフェースするための方法および装置に関する。

【0002】

【従来の技術】

専用のかつ専門化されたハードウェア機能要素によって基礎 (ベースライン : b a s e l i n e) アーキテクチャプロセッサの機能性を拡張する能力は規模変更可能な (s c a l e a b l e) および拡張性のあるアーキテクチャの重要な観点である。

【0003】

基礎アーキテクチャプロセッサの機能を拡張するための1つの好ましい方法はコプロセッサを使用することによるものである。これらはプロセッサの指令で動作する専用の通常は

10

20

30

40

50

单一目的のプロセッサである。コプロセッサの伝統的な用途の1つは浮動小数点能力をそのようなものを直接サポートしないアーキテクチャに対して選択的に提供するための数学的または数値演算コプロセッサとしてのものであった。そのような数学的コプロセッサのいくつかの例はインテル(Intel)社の8087型および80287型のものである。コプロセッサのいくつかの他の可能性ある用途または形式は、乗算・累算器(mult i p l y - a c c u m u l a t o r s)、変調器/復調器(モデム)、デジタル信号プロセッサ(D S P)、ビタービ計算機(v i t t u r b i c a l c u l a t o r s)、暗号プロセッサ、画像プロセッサ、およびベクトルプロセッサを含む。

【 0 0 0 4 】

【発明が解決しようとする課題】

コプロセッサに対する2つの異なる手法がある。一方においては、デジタル・イクリップメント・コードレイション(D E C)のPDP-11型ファミリのコンピュータのための浮動小数点ユニットはその主プロセッサに対して非常に密接に(t i g h t l y)結合された。発生した1つの問題はこの密接な結合が主プロセッサがコプロセッサの動作についてかなりの程度まで知っていることを必要としたことである。これは新しいコプロセッサを集積または統合システムに付加することが主たる工学技術上の問題になるほど回路設計を複雑にする。

【 0 0 0 5 】

別の構成はコプロセッサを主プロセッサに対してルーズに(l o o s e l y)結合することであった。これは主プロセッサからコプロセッサの動作を引き離し(a b s t r a c t i n g)かつ分離する利点を有し、かつ従って新しいコプロセッサを現存のプロセッサと共に集積または統合するのに必要な努力を実質的に軽減した。しかしながら、これは常にかなりの犠牲をはらって行われた。性能の喪失がこの手法の1つの問題である。このルーズな結合から生じる形式の性能ヒットをとることに伴う1つの問題はそのようなコプロセッサに頼ることに対する損益分岐点が相応じて増大することである。従って、コプロセッサに対するそうでなければ魅力的な数多くの応用はコスト効率がよくない。さらに、そのような手法はしばしば、すべての対応する付加的な回路およびチップ面積と共に、バスを使用することを必要とする。

【 0 0 0 6 】

従って、たとえ非常に簡単な機能に頼ることでも有利になるようにインターフェースを使用することが十分高速となるよう十分密接に結合され、一方プロセッサのアーキテクチャがいすれか与えられたコプロセッサのできるだけ多くの細部から分離できるようにインターフェースを引き離すようなコプロセッサンタフェースをもつことが重要である。この後者の一部はハードウェアに代えてソフトウェアで新しいコプロセッサのアプリケーションをあつらえることができるようにするためインターフェースのプログラマが親しみ易くする(f r i e n d l y)ことを含む。

【 0 0 0 7 】

【課題を解決するための手段】

本発明の一態様では、通信バス(30)を介して結合されたコプロセッサ(14 , 16)とカウントフィールドおよびコードフィールドを備えた少なくとも1つの命令(H_C A L_L)の実行において協働するよう構成されたプロセッサ(12)における、前記命令を実行をする方法が提供され、該方法は、前記命令を受ける段階、前記通信バスにより第1のサイクルを介して、前記コプロセッサに前記カウントおよびコードフィールドを提供する段階、もし前記カウントフィールドがゼロより大きなある値、n、を有していれば、前記コプロセッサに、前記通信バスによって第2のサイクルを介して、第1のオペランドを提供する段階、そして前記命令を完了する段階、を具備することを特徴とする。

【 0 0 0 8 】

本発明の別の態様では、通信バス(30)を介して結合されたコプロセッサ(14 , 16)と実効アドレス計算フィールドを備えた少なくとも1つの命令(H_L D)の実行において協働するよう構成されたプロセッサ(12)における、前記命令を実行する方法が提

10

20

30

40

50

供され、該方法は、前記命令を受ける段階、前記通信バスによって第1のサイクルを介して、前記コプロセッサに前記実効アドレス計算フィールドを提供する段階、前記実効アドレス計算フィールドに従って実効アドレスを計算する段階、前記計算された実効アドレスに格納されたオペランドをフェッチする段階、前記通信バスによって第2のサイクルを介して、前記コプロセッサに前記フェッチされたオペランドを提供する段階、そして前記命令を完了する段階、を具備することを特徴とする。

【0009】

本発明のさらに別の態様では、通信バス(30)を介して結合されたコプロセッサ(14, 16)と実効アドレス計算フィールドを備えた少なくとも1つの命令(H_S_T)の実行において協働するよう構成されたプロセッサ(12)における、前記命令を実行する方法が提供され、該方法は、前記命令を受ける段階、前記通信バスによって第1のサイクルを介して、前記コプロセッサに前記実効アドレス計算フィールドを提供する段階、前記実効アドレス計算フィールドに従って実効アドレスを計算する段階、前記通信バスによって第2のサイクルを介して、前記コプロセッサからオペランドを受ける段階、前記計算された実効アドレスに前記受けたオペランドを格納する段階、そして前記命令を完了する段階、を具備することを特徴とする。

【0010】

【発明の実施の形態】
本発明の特徴および利点は添付の図面と共に以下の詳細な説明を参照することによりさらに明瞭に理解され、図面においては同じまたは同様の参照数字は同じかつ対応する部分を示している。

【0011】

以下の説明においては、数多くの特定の細部は特定のワードまたはバイト長などで説明され本発明の完全な理解を与えている。しかしながら、当業者には本発明はそのような特定の細部なしで実施できることが明らかであろう。他の場合には、本発明を不必要的細部によって不明確にすることのないよう回路がブロック図形式で示されている。たいていの場合、タイミングの考慮事項その他に関する詳細は本発明の完全な理解を得るのに必要でない限り省略されておりかつ関連する技術に習熟した当業者の技術の範囲内にある。

【0012】

用語「バス(bus)」は、データ、アドレス、制御またはステータスのような、1つまたはそれ以上の種々の形式の情報を転送するために使用できる複数の信号または導体に言及している。用語「肯定する(assert)」および「否定する(negate)」は信号、ステータスピットまたは同様の装置をそれぞれその論理的に真のまたは論理的に偽の状態にすることに言及する場合に使用される。もし論理的に真の状態が論理レベル“1”であれば、論理的に偽の状態は論理レベル“0”である。また、もし論理的に真の状態が論理レベル“0”であれば、論理的に偽の状態は論理レベル“1”である。

【0013】

図1は、データ処理システム10の一実施形態を示すブロック図であり、該データ処理システム10はプロセッサ12、コプロセッサ14、コプロセッサ16、メモリ18、他のモジュール20および外部バスインターフェース22を含みこれらはすべてバス28によって双方向的に結合されている。本発明の別の実施形態は1個だけのコプロセッサ14、2つのコプロセッサ14および16、またはさらに多くのコプロセッサ(図示せず)をもつものとすることができます。外部バスインターフェース22は集積回路端子35によって外部バス26に双方向的に結合されている。メモリ24は双方向的に外部バス26に結合されている。プロセッサ12は任意選択的に集積回路端子31によってデータ処理システム10の外部に結合することができる。コプロセッサ14は任意選択的に集積回路端子32によってデータ処理システム10の外部に接続することができる。メモリ18は任意選択的に集積回路端子33を介してデータ処理システム10の外部に接続することができる。他のモジュール20も任意選択的に集積回路端子34を介してデータ処理システム10の外部に結合することができる。プロセッサ12はコプロセッサインターフェース30によって

コプロセッサ14およびコプロセッサ16の双方に双方向的に接続されている。

【0014】

図2は、図1のプロセッサ12の一部を示すブロック図である。一実施形態では、プロセッサ12は制御回路40、命令デコード回路42、命令パイプ44、レジスタ46、演算論理ユニット(ALU)48、ラッチングマルチプレクサ(latching multplexer: MUX)50、ラッチングマルチプレクサ(MUX)52、およびマルチプレクサ(MUX)54を含む。本発明の一実施形態では、コプロセッサインターフェース30は信号60～71を含む。クロック信号60は制御回路40によって発生される。コプロセッサオペレーション信号61は制御回路40によって発生されかつコプロセッサ14および16に与えられる。

10

【0015】

スーパバイザモード信号62は制御回路40によって発生されかつコプロセッサ14および16に提供される。デコード信号63は制御回路40によって発生されかつコプロセッサ14および16に提供される。コプロセッサビジー信号(busy signal)64はコプロセッサ14またはコプロセッサ16から制御回路40によって受信される。実行信号(execute signal)65は制御回路40によって発生されかつコプロセッサ14および16に提供される。例外信号(exception signal)66はコプロセッサ14またはコプロセッサ16から制御回路40によって受信される。レジスタ書き込み(REGWR*)信号67は制御回路40によって発生されかつコプロセッサ14および16に提供される。レジスタ信号(REG{4:0})68は制御回路40によって発生されかつコプロセッサ14および16に提供される。エラー信号(H_ERR*)69は制御回路40によって発生されかつコプロセッサ14および16に提供される。データストローブ信号(H_DS*)70は制御回路40によって発生されかつコプロセッサ14および16に提供される。データアクノレッジ(肯定応答)信号(H_DA*)71は制御回路40によってコプロセッサ14およびコプロセッサ16から受信される。コプロセッサインターフェース70の一部とも考えられる、ハードウェアデータポート信号(HDP{31:0})72はコプロセッサ14および16とプロセッサ12内の内部回路との間で双方向的なものである。

20

【0016】

本発明の一実施形態では、複数の信号がバス28に対しまたはバス28から提供されてメモリ18および/またはメモリ24に対しデータをロードしまたは格納(ストア)する。一実施形態では、これらの信号は制御回路40によって発生されかつバス28に提供される転送要求信号(TREQ*)73を含む。転送エラーアクノレッジ信号(TEA*)74はバス28によって制御回路40に提供される。転送アクノレッジ信号(TA*)75はバス28によって制御回路40に提供される。命令はバス28から導体76によって命令パイプ(instruction pipe)44に提供される。データは導体76によってMUX54に提供される。

30

【0017】

ドライブデータ信号79はトライステートバッファ95をイネーブルして導体88および76によってラッチングMUX52からデータを提供できるようにする。アドレス選択信号78はラッチングMUX50が導体77によってバス28にアドレスを提供できるようになる。MUX54への他の入力はHDP信号(HDP{31:0})72によって提供される。MUX54への他の入力はALU結果導体86によって提供される。MUX54の出力、結果信号(result signal)83、はレジスタ46にかつトライステートバッファ96の入力に提供される。ドライブHDP信号82はトライステートバッファ96がHDP信号72によって結果信号83をドライブできるようにする。トライステートバッファ96の出力はまたラッチングMUX52の入力に結合されている。

40

【0018】

本発明の別の実施形態はレジスタ46に任意の数のレジスタを含めることができる。結果信号83はラッチングMUX50への入力として提供される。結果信号83はMUX54

50

によってレジスタ 4 6 に提供される。結果選択信号（結果_選択：R E S U L T _ S E L E C T ）8 1 はM U X 5 4 のどの入力が結果導体 8 3 にドライブされるべきかを選択する。ソース選択信号（ソース_選択：S O U R C E _ S E L E C T ）8 0 はラッチングM U X 5 2 に提供されてどの信号が導体 8 8 によってトライステートバッファ 9 5 にドライブされるかを選択する。

【 0 0 1 9 】

制御回路 4 0 は導体 9 1 によって制御情報を提供しあつレジスタ 4 6 からステータス情報を受信する。制御回路 4 0 は導体 9 2 によって制御信号を提供しあつ演算論理ユニット 4 8 からステータス信号を受信する。制御回路 4 0 は導体 9 3 によって制御信号を提供しあつ命令パイプ 4 4 および命令デコード回路 4 2 からステータス信号を受信する。命令パイプ 4 4 は導体 8 9 によって命令デコード回路 4 2 に命令を提供するよう結合されている。命令デコード回路 4 2 は導体 9 0 によって制御回路 4 0 に対しデコードされた命令情報を提供する。レジスタ 4 6 は導体 8 4 によってソースオペランドを演算論理ユニット 4 8 に提供する。レジスタ 4 6 は導体 8 0 、ラッチングM U X 5 2 、トライステートバッファ 9 5 および導体 7 6 によってメモリ 1 8 またはメモリ 2 4 に格納されるべきデータを提供する。レジスタ 4 6 は導体 8 4 、ラッチングM U X 5 0 およびアドレス導体 7 7 によってメモリ 1 8 またはメモリ 2 4 にアドレス情報を提供する。レジスタ 4 6 は導体 8 5 によって演算論理ユニット 4 8 に第 2 のソースオペランドを提供する。

【 0 0 2 0 】

図 3 は、コプロセッサ 1 4 の一部の一実施形態を示すブロック図である。一実施形態では、コプロセッサ 1 4 は制御回路 1 0 0 、計算回路 1 0 2 および任意選択的な記憶回路 1 0 4 を含む。制御回路 1 0 0 は信号 6 0 ~ 7 2 を含むコプロセッサインタフェース 3 0 によってプロセッサ 1 2 に双方向的に結合されている。本発明の一実施形態では、制御回路 1 0 0 はデコード回路 1 0 6 を含み、該デコード回路 1 0 6 はプロセッサ 1 2 からオペレーション信号 6 1 およびデコード信号 6 3 を受信する。制御回路 1 0 0 は導体 1 0 8 によって制御情報を提供しあつ任意選択的な記憶回路 1 0 4 からステータス情報を受信する。制御回路 1 0 0 は導体 1 0 9 によって制御情報を提供しあつ計算回路 1 0 2 からステータス情報を受信する。計算回路 1 0 2 および任意選択的な記憶回路 1 0 4 は導体 1 1 0 によって双方向的に結合されている。信号 1 1 0 の 1 つまたはそれ以上はバス 2 8 または集積回路端子 3 2 に対しまたはこれらから提供できる。制御回路 1 0 0 は導体 1 1 2 によってバス 2 8 または集積回路端子 3 2 に対し情報を受信しあるいは情報を提供することができる。信号 7 2 は計算回路 1 0 2 および任意選択的な記憶回路 1 0 4 に双方向的に結合されている。さらに、信号 7 2 は双方向的にバス 2 8 または集積回路端子 3 2 に結合することができる。本発明の別の実施形態では、任意選択的な記憶回路 1 0 4 は実施しなくてもよい。任意選択的な記憶回路 1 0 4 が実施されない本発明の実施形態においては、それはレジスタ、任意の種類のメモリ、ラッチまたはプログラマブル論理アレイ、その他を含む任意の種類の記憶回路を使用して構成できる。本発明の別の実施形態では、計算回路 1 0 2 は任意の種類の論理または計算機能を行うことができる。

【 0 0 2 1 】

本システムは特定の用途に関連する動作のために最適化された外部コプロセッサ 1 4 （またはハードウェアアクセラレータ）によるタスク加速のためのサポートを提供する。これらの外部コプロセッサ 1 4 , 1 6 は人口または母集団（p o p u l a t i o n ）のカウントを行うためのコプロセッサ 1 4 としての簡単なものでもよく、あるいはD S P 加速コプロセッサ 1 4 または高速乗算 / 累算操作が可能なコプロセッサ 1 4 のようなより複雑な機能のものとすることもできる。

【 0 0 2 2 】

データは特定の構成に対して適切な 1 つまたはそれ以上のいくつかのメカニズムによってコプロセッサ 1 2 とコプロセッサ 1 4 との間で転送される。これらはコプロセッサ 1 4 への転送、およびコプロセッサ 1 4 からの転送へと分割できる。

【 0 0 2 3 】

10

20

30

40

50

コプロセッサ14へのデータの転送のための1つのメカニズムはレジスタスヌーピングメカニズム (Register Snooping mechanism) であり、これは命令プリミティブ (instruction primitive) を含まないが、通常のプロセッサ12の動作の副産物である。これはコプロセッサ14が1つまたはそれ以上のプロセッサ12のレジスタへの更新を監視できるようにインターフェースにわたってプロセッサ12の汎用目的のレジスタ (“GPR”) 46に対する更新を反映することを含む。これはもしコプロセッサ14が内部レジスタまたは機能に対してGPR46を「オーバレイ (overlays)」する場合に適切であろう。この場合、プロセッサ12からコプロセッサ14への明確な (explicit) 受け渡しは必要とされないであろう。

【0024】

10

命令プリミティブ (Instruction primitives) は外部コプロセッサ14, 16およびプロセッサ12の間でオペランドおよび命令の明確な転送のためにベースプロセッサ12において提供される。命令およびデータ転送のレートに対する制御を可能にするためハンドシェイクメカニズムが提供される。

【0025】

コプロセッサ14の機能は構成に特有の (implementation specific) ユニットとなるよう設計され、従って与えられたユニットの正確な機能は、同じ命令マッピングが存在しても、異なる構成にわたり自由に変更できる。

【0026】

20

図4は、レジスタスヌーピング動作を示すタイミング図である。コプロセッサ14または外部モニタへのパラメータの受け渡しの性能オーバヘッドを避けるため、レジスタスヌーピングメカニズムが提供される。これはコプロセッサ14がプロセッサ12の汎用レジスタ46の1つまたはそれ以上のシャドウコピー (shadow copy) を実施できるようとする。この能力はプロセッサのGPR46の1つへ書き込まれる値を転送しつどのレジスタ46が各々のGPRの更新に対して更新されるかを指示することによって実施される。ストローブ信号REGWR*67が各々のレジスタの更新に対して肯定される。この値は32ビットの双方向データバスHDP[31:0]72にわたって転送され、かつ5ビットのレジスタナンババスが更新される (REG[4:0]) 68の実際のプロセッサレジスタ46へのポインタを提供する。前記レジスタナンバは通常のファイル (normal file) におけるあるいは代わりのファイル (alternate file) におけるレジスタ46に言及または参照することができる。好ましい実施形態では、オルタネイトまたは代わりのファイルのレジスタはREG[4]==1によって示されかつノーマルまたは通常のファイルのレジスタはREG[4]==0によって示される。しかしながら、この発明はレジスタセットの実際の区分 (partitioning) には依存しないことに注意を要する。

30

【0027】

専用の12ビット命令バス (H_OP[11:0]) 61は外部コプロセッサ14に対して発行されるコプロセッサインターフェース30のオペコード (opcode) を提供する。このバスはプロセッサのオペコードの下位 (low order) 12ビットを反映する。上位 (high-order) 4ビットはそれらは常に0b0100であるため反映されない。スーパバイザモード指示子 (H-SUP) 62もまたPSR(S)ビットの現在の状態を示すために提供され、プロセッサがスーパバイザまたはユーザモードのいずれで動作しているかを示す。これはあるコプロセッサ機能をスーパバイザモードに限定するのに有用なものとすることができます。プロセッサ12と外部コプロセッサ14, 16の間の一組のハンドシェイク信号はコプロセッサインターフェース30の命令実行を調整する。

40

【0028】

プロセッサ12によって発生される制御信号はプロセッサ12の内部パイプライン構造の反映である。プロセッサパイプライン44は命令フェッチ、命令デコード42、実行、および結果の書き戻し (writeback) のステージから構成される。それは1つまたはそれ以上の命令レジスタ (IR) を含む。プロセッサ12はまた命令プリフェッチャバッ

50

ファを含みデコードステージ42の前に命令のバッファリングを可能にする。命令は命令デコードレジスタIRに入ることによりこのバッファから命令デコードステージ42へと進行する。

【0029】

命令デコーダ42はIRから入力を受け、かつIRに保持された値に基づき出力を発生する。これらのデコード42の出力は常に有効であるとは限らず、かつ例外条件または命令フローの変更により捨てられることがある。有効な場合でも、命令はそれらが命令パイプラインの実行ステージへ進行することができるまでIRに保持することができる。これは前の命令が実行を完了するまで（これは複数クロックを必要とするであろう）生じ得ないから、デコーダはIRが更新されるまでIRに含まれる値をデコードし続けることになる。

10

【0030】

図5は、命令ハンドシェイクのための基本的な命令インターフェース動作を示すタイミング図である。命令デコードストローブ（H_DEC*）信号63はプロセッサ12によってコプロセッサインターフェース30のオペコードのデコードを示すために与えられる。この信号は、たとえ命令が実行されずに捨てられる場合でも、コプロセッサインターフェース30のオペコードがIRに存在する場合に肯定されることになる。前記H_DEC*63の出力は命令が実際に発行されまたは捨てられるまで同じ命令に対して複数クロックの間肯定された状態に留まることができる。

【0031】

ビジー信号（H_BUSY*）64は外部コプロセッサ14がコプロセッサインターフェース30の命令を受け入れることができるか否かを判定するためプロセッサ12によって監視され、かつ命令の発行が生じる場合に対し部分的に制御を行う。もし前記H_BUSY*64信号が、H_DEC*63が肯定されている間に、否定されれば、命令実行はインターフェースによってストールされず、かつH_EXEC*65信号が命令実行が進行できるや否や肯定できるようになる。もしH_BUSY*64信号がプロセッサ12がコプロセッサインターフェース30のオペコードをデコードする場合に（H_DEC63*の肯定によって示される）肯定されれば、コプロセッサインターフェース30のオペコードの実行は強制的にストールされることになる。いったんH_BUSY*64信号が否定されると、プロセッサ12はH_EXEC*65を肯定することにより命令を発行することができる。もしコプロセッサ14が命令をバッファリングできれば、H_BUSY*64信号はバッファの充填を助けるために使用することができる。

20

【0032】

図6は、H_BUSY*64がコプロセッサインターフェース30の命令実行を制御するために使用される場合の命令インターフェース動作を示すタイミング図である。いったん何らかの内部インストール条件が解消されると、かつH_BUSY*64信号が否定されると、プロセッサはH_EXEC*65を肯定してコプロセッサインターフェース30の命令がパイプラインの実行ステージに入ったことを示す。外部コプロセッサ14は命令の実際の実行を制御するためにH_EXEC*65信号を監視すべきであり、その理由はプロセッサがある状況において実行する前にその命令を放棄できるからである。もし前の命令の実行が結果として例外（exception）が取られることになれば、H_EXEC*65信号は肯定されず、かつH_DEC*63出力は否定されることになる。同様のプロセスはIRにおける命令がプログラムフローの変化の結果として廃棄されれば生じ得る。

30

【0033】

図7は、命令廃棄または命令放棄（instruction discard）を示すタイミング図である。もしある命令が廃棄されれば、H_DEC*63信号は他のコプロセッサインターフェース30のオペコードがH_OP[11:0]61のバスに与えられる前に否定されることになる。

40

【0034】

図8は、命令パイプラインストールの例を示すタイミング図である。H_DEC*63が

50

肯定されかつ H_BUS_Y * 64 が否定されてもプロセッサ 12 が H_EXEC * 65 の肯定を遅らせることができる状況がある。これは前の命令が完了するのを待機している間に生じる得る。

【0035】

図 9 は、ストールのないバック - バック (back-to-back) 実行の例を示すタイミング図である。バック - バックコプロセッサインターフェース 30 の命令に対しては、H_OP[11:0] 61 バスが新しい命令が IR に入る際に更新されても、否定されることなく肯定された状態に留まることができる。一般に、H_EXEC * 65 の肯定は前のクロックに際してデコードされた命令の実行に対応する。

【0036】

図 10 は、内部パイプラインストールを有するバック - バック動作を示すタイミング図である。この場合、H_BUS_Y * 64 は否定されるが、プロセッサは内部ストール条件が消えるまで第 2 のコプロセッサインターフェース 30 の命令に対して H_EXEC * 65 を肯定しない。

【0037】

図 11 は、H_BUS_Y * 64 ストールを有するバック - バックコプロセッサインターフェース 30 の命令を示すタイミング図である。この例では、外部コプロセッサ 14 はビジーであり、かつ直ちに第 2 の命令を受け入れることはできない。H_BUS_Y * 64 は第 2 の命令がプロセッサ 12 によって発行されることを防止するため肯定する。いったんコプロセッサ 14 が自由になると、H_BUS_Y * 64 が否定され、かつ次のコプロセッサインターフェース 30 の命令が実行ステージへと進む。

【0038】

コプロセッサインターフェース 30 のオペコードのデコードに関連する例外は外部コプロセッサ 14 によって H_EXEC_P * 66 信号によって通知することができる。プロセッサ 12 へのこの入力は H_DEC * 63 が肯定されかつ H_BUS_Y * 64 が否定されるクロックサイクルの間にサンプルされ、かつもしコプロセッサインターフェース 30 のオペコードが前に述べたように廃棄されなければハードウェアコプロセッサ 14 の例外に対して例外処理を生じる結果となる。この例外処理の詳細は以下に説明する。

【0039】

図 12 は、コプロセッサインターフェース 30 のオペコードのデコードおよび試みられた実行に応じてコプロセッサ 14 により肯定される H_EXEC_P * 66 信号の例を示すタイミング図である。H_EXEC_P * 66 信号は H_DEC * 63 が肯定されかつ H_BUS_Y * 64 が否定されるクロックの間にプロセッサ 12 によってサンプルされる。H_EXEC * 65 信号は例外がインターフェースによって通知されるか否かにかかわりなく肯定され、この肯定は例外が取られる場合を命令廃棄の場合と区別する。

【0040】

前記例外は前のクロックサイクルにデコードされた命令に対応し、かつ実際の実行は行われるべきでないことに注意を要する。コプロセッサ 14 は違背 (offending) 命令を受け入れかつそれに対するプロセッサパイプラインの実行ステージが認識される前に例外を通知しなければならない。前記 H_EXEC_P * 66 信号は H_DEC * 63 が否定されあるいは H_BUS_Y * 64 が肯定されるすべてのクロックサイクルに対して無視される。

【0041】

図 13 は、コプロセッサインターフェース 30 のオペコードのデコードおよび試みられた実行に応じてコプロセッサ 14 によって肯定される前記 H_EXEC_P * 66 信号の例を示すタイミング図である。これを図 14 のタイミング図と比較すると、この例では、コプロセッサインターフェース 30 の命令は廃棄され、従って H_EXEC * 65 信号は肯定されず、かつ H_DEC * 63 は否定される。

【0042】

図 14 は、H_BUS_Y * 64 が肯定されて例外を生じる結果となるコプロセッサインターフェース 30 のオペコードのデコードおよび試みられた実行に応じてコプロセッサ 14 によって肯定される前記 H_EXEC_P * 66 信号の例を示すタイミング図である。これを図 13 のタイミング図と比較すると、この例では、コプロセッサインターフェース 30 の命令は廃棄され、従って H_EXEC * 65 信号は肯定されず、かつ H_DEC * 63 は否定される。

10

20

30

40

50

フェース30のオペコードの実行を遅らせる例を示すタイミング図である。

【0043】

H_BUS Y * 64 および H_EXCP * 66 信号はすべてのコプロセッサ14, 16によって共有され、従ってそれらは調整された方法でドライブされなければならない。これらの信号は H_DEC * 63 が肯定されるクロックサイクルに際して H_OP [11:10] 61 に対応してコプロセッサ14, 16 により（ハイまたはローに、いずれか適切なものに）ドライブされるべきである。出力をクロックのローの部分の間にのみドライブすることにより、これらの信号は競合なしに複数のコプロセッサ14, 16 によって共有することができる。ラッチをプロセッサ12の内部に保持することはこの入力に対してそれを何らのユニットもそれをドライブしていない間にクロックのハイのフェーズに対して有効な状態に保持するために提供される。 10

【0044】

コプロセッサインターフェース30の命令プリミティブのいくつかはまたプロセッサ12および外部コプロセッサ14の間でデータ項目またはデータアイテムの転送を行うことを含む。オペランドはコプロセッサインターフェース30にわたって実行されている特定のプリミティブの関数として転送することができる。プロセッサ12のGPRの1つまたはそれ以上を32ビットの双方向のデータパスにわたってコプロセッサ14からあるいはコプロセッサ14へ転送するための備えが設けられている。さらに、データシンク/ソース（sink/source）がコプロセッサインターフェース30として单一データ項目をメモリ18にロードあるいはメモリ18からストアするための備えも設けられている。プロセッサ12はパラメータを CLK60のハイ部分の間に HDP [31:0] 72 を介して外部コプロセッサ14に受け渡し、オペランドはクロックのローフェーズの間にプロセッサ12によってコプロセッサインターフェース30から受信されかつラッチされる。小さな期間のバスハンドオフを可能にするためドライブが行われる前にクロックがハイに遷移する際に遅延が与えられる。コプロセッサ14のインターフェースは立下りクロックエッジにおいて同じ小さな遅延を提供しなければならない。データ項目のハンドシェイクはデータストローブ（H_DS * 70）出力、データアクノレッジ（H_DA * 71）入力、およびデータエラー（H_ERR * 69）出力信号によってサポートされる。 20

【0045】

プロセッサ12はソフトウェアサブルーチンが呼び出され（call ed）または戻る（return ed）のとほぼ同じ方法でコプロセッサインターフェース30に対し呼出したはコール（call）または戻り（return）パラメータのリストを転送する能力を提供する。アーギュメント（arguments）のカウントは渡されるパラメータの数を制御するため H_CALL または H_RET プリミティブに示される。プロセッサ12のレジスタR4の内容で始まるレジスタ値は H_CALL (H_RET) プリミティブの実行の一部として外部コプロセッサ14へ（から）転送される。7つまでのレジスタパラメータが受け渡しできる。この規約はソフトウェアサブルーチンコールの規約と同様のものである。 30

【0046】

オペランド転送のハンドシェイクはデータストローブ（H_DS * 70）出力およびデータアクノレッジ（H_DA * 71）入力信号によって制御される。データストローブは転送の期間の間プロセッサ12によって肯定され、かつ転送はプロセッサ12のインターフェース動作とほぼ同様に、重複した方法で行われることになる。データアクノレッジ（H_DA * 71）はデータエレメントがコプロセッサ14によって受け入れられまたはドライブされたことを示すために使用される。 40

【0047】

図15は、前記 H_CALL プリミティブに関連するレジスタ46の転送の例を示すタイミング図である。命令プリミティブは複数のプロセッサレジスタを転送するために提供されかつ該転送は理想的にはクロックごとに行うことができる。外部コプロセッサ14への転送のために、プロセッサは現在のアイテムのアクノレッジの前に（または同時に）次の 50

オペランドを自動的にドライブし始める（もし必要であれば）。外部論理はデータの喪失がないことを保証するため1つのレベルのバッファリングが可能でなくてはならない。この図はコプロセッサインタフェース30へのH_C A L L転送のシーケンシングまたは順序付けを示しており、この場合2つのレジスタが転送されるべきものである。第2の転送は否定されたデータアクノレッジ（H_D A * ）71により反復される。

【0048】

外部コプロセッサ14からプロセッサレジスタ46への転送のために、プロセッサ12はH_D S * 70が肯定された後のクロックサイクルごとに外部コプロセッサ14から値を受け入れることができ、かつこれらの値はそれらが受信される際にレジスタファイル46へと書き込まれ、従ってバッファリングは必要とされない。

10

【0049】

図16は、前記H_R E Tプリミティブに関連するレジスタ46の転送の例を示すタイミング図である。この例では、2つのレジスタ46の値が転送される。コプロセッサ14はH_E X E C * 65信号の肯定に続くクロックで始まりデータを駆動することができ、それはこれがH_D S * 70が始めに肯定されるクロックであるためである。H_D S * 70の出力はC L K 60の立上がりエッジで遷移し、一方H_D A * 71の入力はC L K 60のローフェーズの間にサンプルされる。

【0050】

プロセッサ12はH_L DまたはH_S T命令プリミティブによってコプロセッサインタフェース30へあるいはから单一命令オペランドを転送する能力を提供する。

20

【0051】

H_L Dプリミティブはメモリ18からコプロセッサ14へデータを転送するために使用される。コプロセッサ14へのオペランド転送のハンドシェイクはデータストローブ（H_D S * ）70信号によって制御される。データストローブはプロセッサ12によって肯定されて有効なオペランドがH_D P [31 : 0] 72バス上に与えられたことを示す。データアクノレッジ（H_D A * ）71入力はこの転送に対しては無視される。

【0052】

図17は、コプロセッサインタフェース30へのH_L D転送のシーケンシングを示すタイミング図である。この場合、1つのノーウエイト状態（no-wait state）のメモリ18のアクセスがある。nのウェイト状態を有するメモリ18のアクセスに対しては、オペランドおよびH_D S * 70はnクロック後にドライブされる。もしベースレジスタ46をロードの実効アドレスによって更新するためのオプションが選択されれば、更新値はそれが計算された後の最初のクロック（H_E X E C * 65の肯定に続くクロック）でH_D P [31 : 0] 72においてドライブされる。

30

【0053】

図18は、メモリ18のアクセスが結果としてアクセス例外になる場合のプロトコルを示すタイミング図である。そのような場合、H_E R R * 69信号は外部コプロセッサ14へと肯定し戻される。

【0054】

H_S Tプリミティブはデータをコプロセッサ14からメモリ18に転送するために使用できる。もし前記ストアの実効アドレスによってベースレジスタ46を更新するオプションが選択されれば、該更新値はそれが計算された後の最初のクロック（H_E X E C * 65の肯定に続くクロック）でH_D P [31 : 0] 72においてドライブされる。

40

【0055】

図19は、H_S Tプリミティブに関連する転送の例を示すタイミング図である。H_S Tプリミティブに関連するハンドシェイクは2つの部分からなり、すなわちストアのためのデータを提供しなければならない、コプロセッサ14からの初期ハンドシェイク、およびいったんメモリ18へのストアが完了した場合のプロセッサ12からの完了ハンドシェイクである。

【0056】

50

前記初期ハンドシェイクはコプロセッサ14がストアデータをプロセッサ12へとドライブしたことを見せるためプロセッサ12へのH_DA*71入力を使用する。H_DA*71信号はコプロセッサ14によってデータがHDP[31:0]72へとドライブされるのと同じクロックで肯定される。ストアデータはハーフワードサイズのストアに対し
てはバスの下半分(lower_half)から取られ、上位16ビットはメモリ18に書き込まれない。前記H_DA*71信号はH_EXEC*65信号が肯定されたクロックと共に始まりサンプルされることになる。メモリサイクルはH_DA*71が認識されるクロックの間要求され、かつストアデータは引き続くクロックに際してメモリ18にドライブされる。いったんストアが完了すると、プロセッサ12はH_DS*70信号を肯定する。

10

【0057】

図20は遅延されたストアデータを備えた転送の例を示すタイミング図である。

【0058】

図21は、ストアがアクセスエラーの結果となる場合のプロトコル信号を示すタイミング図である。ここでは、H_ERR*69信号が肯定されることに注意を要する。もしハードウェアユニットがH_EXEC*65が肯定されるクロックでH_EXCP*66を肯定することにより命令を中止すれば、H_DA*71信号は肯定されるべきではない。

【0059】

図22～26はハードウェアアクセラレータ（またはコプロセッサ）14へのインターフェースのために命令セットの一部として提供される命令を示す。プロセッサ12が前記プリミティブのフィールドのいくつかを解釈し、他のものはコプロセッサ14のみによって解釈される。

20

【0060】

図22は、H_CALLプリミティブに対する命令フォーマットを示す。この命令はコプロセッサ14によって実施される機能を「コール」するまたは「呼び出す」(call)ために使用される。パラダイム(paradigm)は、ハードウェアコンテクストにおけるものを除き、標準的なソフトウェアコール規約と同様である。H_CALLプリミティブはプロセッサ12およびコプロセッサ14の双方によって解釈されて「コールパラメータ(call parameters)」またはアーギュメントのリストをプロセッサ12から転送しあつコプロセッサ14において特定の機能を開始する。

30

【0061】

命令ワードのUUおよびCODE(コード)フィールドはプロセッサ12によって解釈されず、これらはコプロセッサ14の特定の機能を示すために使用される。UUフィールドは特定のコプロセッサ14, 16を特定することができ、かつCODEフィールドは特定の動作または操作を特定することができる。CNTフィールドはプロセッサ12およびコプロセッサ14の双方によって解釈され、かつコプロセッサ14に受け渡すためにレジスターアーギュメントの数を特定する。

【0062】

アーギュメントは汎用レジスタ46からR4で始まりかつR(4+CNT-1)へと続いて受け渡される。7個までのパラメータまたはレジスタ46が単一のH_CALLの実施または発動において受け渡すことができる。

40

【0063】

H_CALL命令はモジュール方式のモジュール発動(modular module invocation)を実施するために使用できる。この種のインターフェースを使用することは長い間より高い信頼性およびより少ないバグを備えたソフトウェアシステムを生み出すために知られてきている。通常、機能パラメータは値(value)によって最もよく受け渡される。これは副作用を大幅に低減する。多くの場合、CおよびC++のようなブロック構造言語に対する近代のコンパイラは短いシーケンスのパラメータまたはアーギュメントをレジスタ46における発動された機能またはサブルーチンに渡す。この技術はH_CALL命令と共に実施することができる。コンパイラは7個までのパラメータま

50

たはアーギュメントを R 4 で始まり引き続くレジスタ 4 6 にロードするよう構成でき、次に H_C A L L 命令を発生し、これは標準的なコンパイラ発生サブルーチンリンクージ命令を置き換える。

【 0 0 6 4 】

図 2 3 は、 H_R E T プリミティブに対する命令フォーマットを示す。この命令はコプロセッサ 1 4 によって実施される機能「からの戻り (r e t u r n f r o m) 」のために使用される。パラダイムは、ハードウェアコンテクストにおけるものを除き、プロセッサ 1 2 によって使用されるソフトウェアコール規約と同様である。 H_R E T プリミティブはプロセッサ 1 2 およびコプロセッサ 1 4 の双方によって解釈されて「戻りパラメータ (r e t u r n p a r a m e t e r s) 」または値のリストをコプロセッサ 1 4 からプロセッサ 1 2 へと転送する。 10

【 0 0 6 5 】

前記命令ワードの U U および C O D E (コード) フィールドはプロセッサ 1 2 によって解釈されず、これらはコプロセッサ 1 4 の特定の機能を示すために使用される。 U U フィールドはハードウェアユニットを特定することができ、かつ C O D E フィールドは戻りのためにコプロセッサ 1 4 における特定の動作または一組のレジスタ 4 6 を特定することができる。 C N T フィールドはプロセッサ 1 2 およびコプロセッサ 1 4 の双方によって解釈され、かつコプロセッサ 1 4 からプロセッサ 1 2 へと渡すレジスタ 4 6 のアーギュメントの数を特定する。 20

【 0 0 6 6 】

アーギュメントは R 4 で始まりかつ R (4 + C N T - 1) へとプロセッサ 1 2 の一般または汎用レジスタ 4 6 へ渡される。 7 個までのパラメータ (またはレジスタ内容) を戻すことができる。

【 0 0 6 7 】

H_C A L L 命令と同様に、 H_R E T 命令はまたモジュール方式のプログラミングを実施するために使用できる。構造プログラミング (s t r u c t u r e d p r o g r a m m i n g) は機能戻り値が値によってコーリングルーチンへと最もよく渡し戻されることを要求する。これはしばしばコンパイラによって 1 つまたはそれ以上の戻り値をサブルーチンまたは機能戻りのためのレジスタに与えることによって効率的に行われる。しかしながら、伝統的な構造プログラミングはサブルーチンまたは機能が該サブルーチンまたは機能の発動の直後に戻ることを期待することに注目すべきである。コプロセッサ 1 4 の場合には、実行はしばしば発動プロセッサ 1 2 のものと非同期である。 H_R E T 命令はプロセッサ 1 2 およびコプロセッサ 1 4 を再同期するために使用できる。従って、プロセッサ 1 2 は 1 つまたはそれ以上のレジスタ 4 6 をロードし、コプロセッサ 1 4 を 1 つまたはそれ以上の H_C A L L 命令によってアクティベイトまたは活性化し、関連のない命令を実行し、かつ次にコプロセッサ 1 4 と再同期し一方で H_R E T 命令を発行することによりコプロセッサ 1 4 から結果としての値 (単数または複数) を受けることができる。 30

【 0 0 6 8 】

図 2 4 は、 H_E X E C プリミティブに対する命令フォーマットを示す。この命令は機能を開始あるいはアクセラレータによって実施される動作モードに入るために使用できる。 H_E X E C 命令は U U フィールドによって特定される特定のコプロセッサ 1 4 , 1 6 における機能を制御するために使用できる。コードフィールドはプロセッサ 1 2 によって解釈されずむしろ指定されたコプロセッサ 1 4 , 1 6 のために確保される。該命令ワードの U U および C O D E フィールドはプロセッサ 1 2 によって解釈または変換処理されず、これらはコプロセッサ 1 4 の特定の機能を特定するために使用される。 U U フィールドは特定のコプロセッサ 1 4 , 1 6 を指定することができ、かつ C O D E フィールドは特定の動作を示すことができる。 40

【 0 0 6 9 】

図 2 5 は、 H_L D 命令のための命令フォーマットを示す。この命令はメモリオペランドを汎用目的のレジスタ (G P R) 4 6 に一時的に格納することなくある値をメモリ 1 8 か 50

らコプロセッサ14に受け渡すために使用される。メモリオペランドはベースポインタおよびオフセットを使用してアドレスされる。

【0070】

H_LD命令はメモリ18へのある値のロードを行い、かつメモリオペランドをそれをレジスタ46に格納することなくコプロセッサ14に渡す。H_LD動作は3つのオプション、w-ワード、h-ハーフワードおよびu-更新(updata)を有する。DISPはIMM2フィールドをロードのサイズでスケーリングし、かつゼロ拡張(zero-extending)を行うことによって得られる。この値はレジスタRXの値に加えられかつ特定されたサイズのロードはこのアドレスから行われ、ロードの結果はハードウェアインターフェース28に渡される。ハーフワードのロードに対しては、フェッチされたデータは32ビットへとゼロ拡張される。もしuオプションが特定されれば、ロードの実効アドレスはそれが計算された後にレジスタRX46に入れられる。10

【0071】

前記命令ワードのUUフィールドはプロセッサ12によって解釈されず、このフィールドは特定のコプロセッサ14, 16を特定することができる。Szフィールドはオペランドのサイズ(ハーフワードまたはワードのみ)を特定する。DispフィールドはRbaseフィールドによって特定されるレジスタの内容に加えられるべき符号なしの(unsigned)オフセット値を特定してロードのための実効アドレスを形成する。Dispフィールドの値は転送されるべきオペランドのサイズによってスケーリングされる。Upフィールドは前記Rbaseレジスタ46がロードの実効アドレスによってそれが計算された後に更新されるべきか否かを特定する。このオプションまたは選択肢は「自動更新(auto-update)」アドレッシングモードを可能にする。20

【0072】

図26は、H_ST命令に対する命令フォーマットを示す。この命令はプロセッサ12のレジスタ46にメモリオペランドを一時的に記憶することなくある値をコプロセッサ14からメモリ18に渡すために使用される。前記メモリオペランドはベースポインタおよびオフセットを使用してアドレスされる。

【0073】

前記命令ワードのUUフィールドはプロセッサ12によって解釈されない。むしろ、このフィールドは特定のコプロセッサ14, 16を特定することができる。Szフィールドはオペランドのサイズ(ハーフワードまたはワードのみ)を特定する。DispフィールドはRbaseフィールドによって特定されるレジスタ46の内容に加えられるべき符号なしのオフセット値を特定しストアのための実効アドレスを形成する。Dispフィールドの値は転送されるべきオペランドのサイズによってスケーリングまたは尺度変更される。Upフィールドは前記Rbaseレジスタが前記ストアの実効アドレスによってそれが計算された後に更新されるべきか否かを特定する。このオプションは「自動更新」アドレッシングモードを可能にする。30

【0074】

H_ST命令はメモリ18へのコプロセッサ14からのオペランドのストアを、それをレジスタ46に格納することなく、達成する。H_ST動作は3つのオプション、w-ワード、h-ハーフワード、およびu-更新を有する。DispはIMM2フィールドをストアのサイズによってスケーリングすることおよびゼロ拡張により得られる。この値はレジスタRXの値に加えられかつ特定されたサイズのストアはこのアドレスに対して行なわれ、該ストアのためのデータはハードウェアインターフェースから得られる。もし前記uオプションが特定されれば、ロードの実効アドレスがそれが計算された後にレジスタRXに入れられる。40

【0075】

前記H_LD命令およびH_ST命令はデータがレジスタ46を通って導き移動されることなくオペランドをメモリ18からコプロセッサ14へかつコプロセッサ14からメモリ18へ移動するための効率的なメカニズムを提供する。オフセットおよびインデクスの備50

えはアレイを効率的に歩進するためのメカニズムを提供する。したがって、これらの命令は特にループ内で有用である。両方の命令はプロセッサ12をコプロセッサ14とロードまたはストアされるそれぞれのオペランドに対して同期させることに注目すべきである。もしこれが必要ではないかあるいは好ましい場合には、あるいは、指定されたレジスタ（単数または複数）46をメモリ18からのデータによって反復的にロードすることによりデータをコプロセッサ14に流し、かつコプロセッサインターフェースバス30もまたレジスタスヌーピングのために使用されるからコプロセッサ14がこれらのロードを検出するようになることができる。

【0076】

当業者は本発明の精神から離れることなく修正および変更を行なうことができることを理解するであろう。したがって、この発明は添付の特許請求の範囲に含まれるすべてのそのような変更および修正を含むものと考える。 10

【0077】

【発明の効果】

以上のように、本発明によれば、プロセッサおよび少なくとも1つのコプロセッサを有するデータ処理システムにおいて、プロセッサとコプロセッサとの間のインターフェースにおける従来技術の欠点を克服し高性能のデータ処理システムが実現できる。

【図面の簡単な説明】

【図1】本発明に係わるデータ処理システムの一実施形態を示すブロック図である。

【図2】図1のプロセッサの一部を示すブロック図である。 20

【図3】図1のコプロセッサの一部の一実施形態を示すブロック図である。

【図4】本発明に係わるレジスタスヌーピング動作を示すタイミング図である。

【図5】命令ハンドシェイクのための基本的な命令インターフェース動作を示すタイミング図である。

【図6】H_BUS Y*信号がコプロセッサインターフェース命令の実行を制御するために使用される場合の命令インターフェース動作を示すタイミング図である。

【図7】命令放棄を示すタイミング図である

【図8】命令パイプラインストールの例を示すタイミング図である。

【図9】ストールのないバック・バック実行の例を示すタイミング図である。

【図10】内部パイプラインストールを有するバック・バック動作を示すタイミング図である。 30

【図11】H_BUS Y*ストールを有するバック・バックコプロセッサインターフェース30の命令を示すタイミング図である。

【図12】コプロセッサインターフェースオペコードのデコードおよび試みられた実行に応じてコプロセッサにより肯定されるH_EXC_P*信号の例を示すタイミング図である。

【図13】コプロセッサインターフェース命令が放棄された場合のコプロセッサインターフェースオペコードのデコードおよび試みられた実行に応じてコプロセッサにより肯定されるH_EXC_P*信号の例を示すタイミング図である。

【図14】コプロセッサインターフェースオペコードの実行を遅延するためにH_BUS Y*が肯定された例を示すタイミング図である。 40

【図15】H_CALLプリミティブに関連するレジスタ転送の例を示すタイミング図である。

【図16】H_RETプリミティブに関連するレジスタ転送の例を示すタイミング図である。

【図17】コプロセッサインターフェースへのH_LD転送のシーケンシングを示すタイミング図である。

【図18】メモリアクセスが結果としてアクセス例外となるプロトコルを示すタイミング図である。

【図19】H_STプリミティブに関連する転送の例を示すタイミング図である。

【図20】遅延されたストアデータを有する転送の例を示すタイミング図である。 50

【図21】ストアが結果としてアクセスエラーとなる場合のプロトコル信号を示すタイミング図である。

【図22】本発明に係わるH_C A L L プリミティブのための命令フォーマットを示す説明図である。

【図23】本発明に係わるH_R E T プリミティブのための命令フォーマットを示す説明図である。

【図24】本発明に係わるH_E X E C プリミティブのための命令フォーマットを示す説明図である。

【図25】本発明に係わるH_L D 命令に対する命令フォーマットを示す説明図である。

【図26】本発明に係わるH_S T 命令のための命令フォーマットを示す説明図である。

10

【符号の説明】

10 データ処理システム

12 プロセッサ

14, 16 コプロセッサ

18 メモリ

20 他のモジュール

22 外部バスインターフェース

24 メモリ

26 外部バス

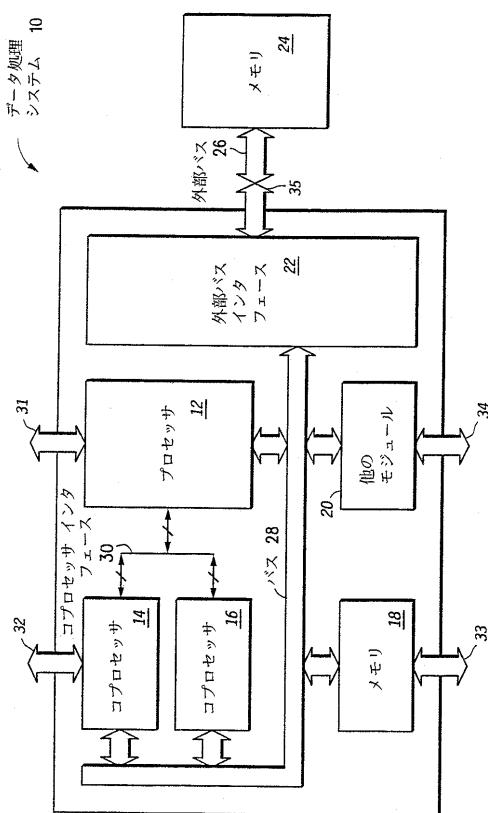
28 バス

30 コプロセッサインターフェース

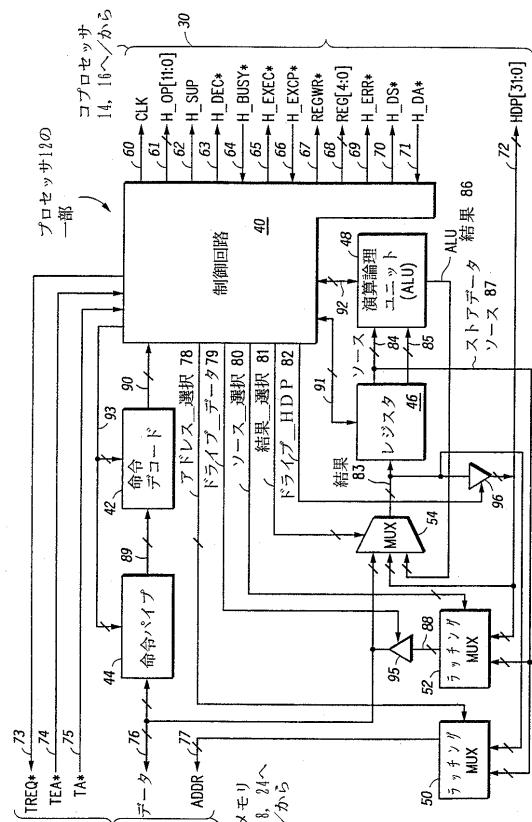
31, 32, 33, 34, 35 集積回路端子

20

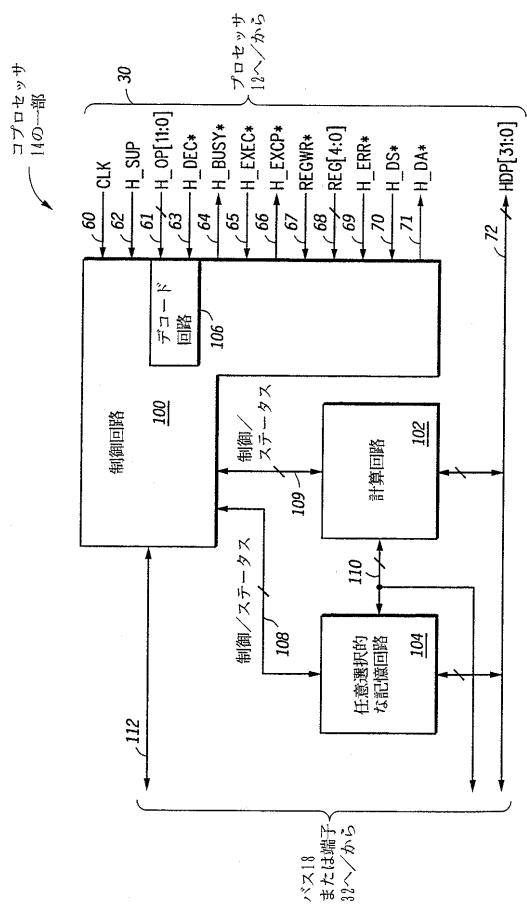
【図1】



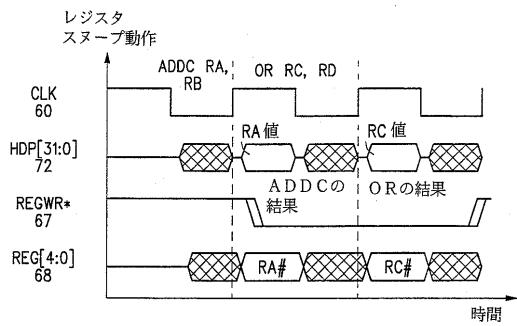
【図2】



【図3】

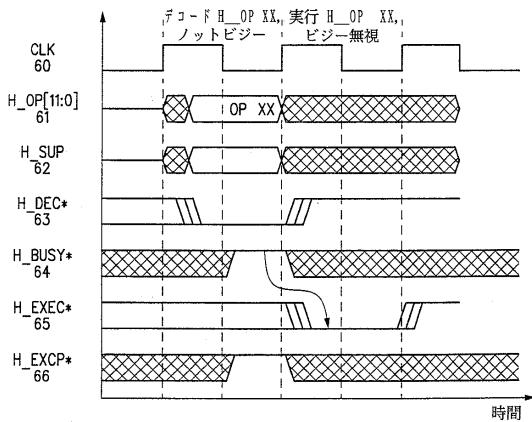


【図4】

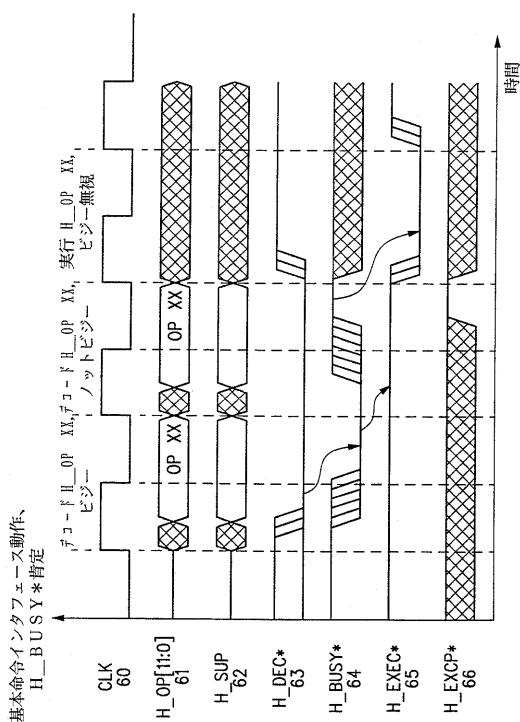


【図5】

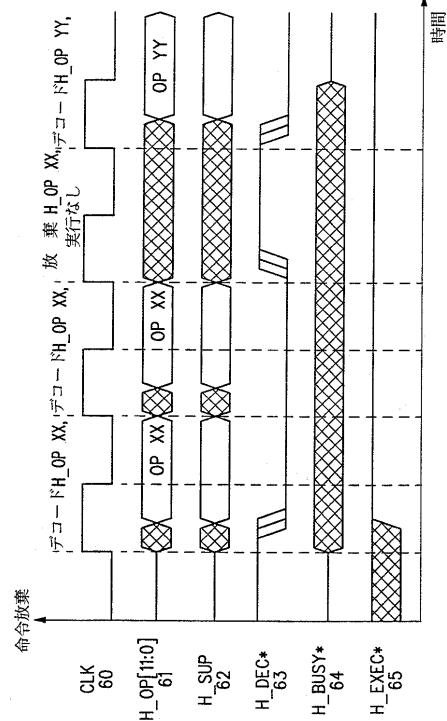
基本命令インターフェース動作、
H_BUSY*否定



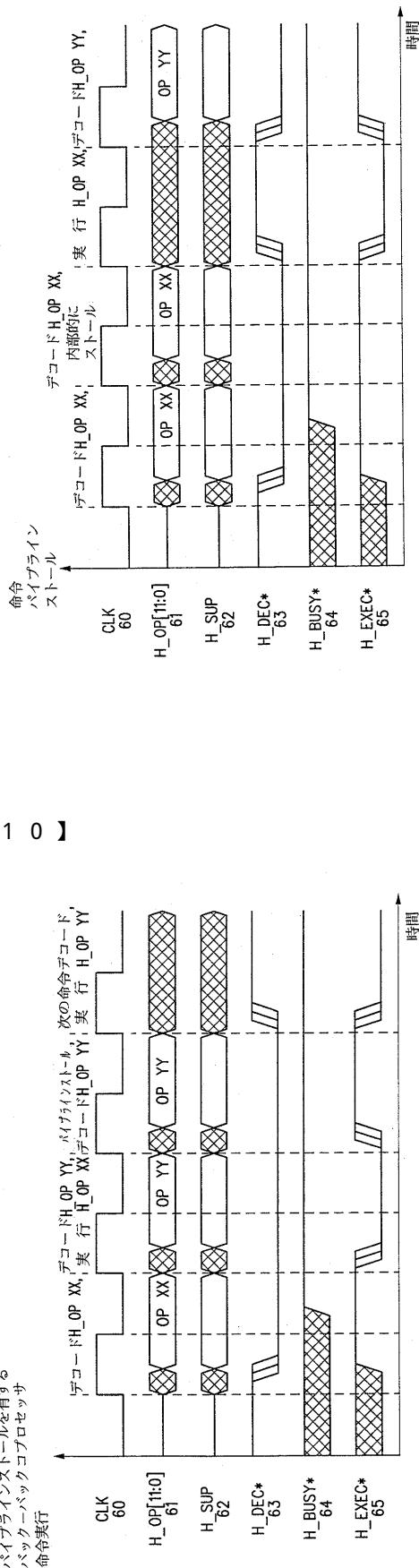
【図6】



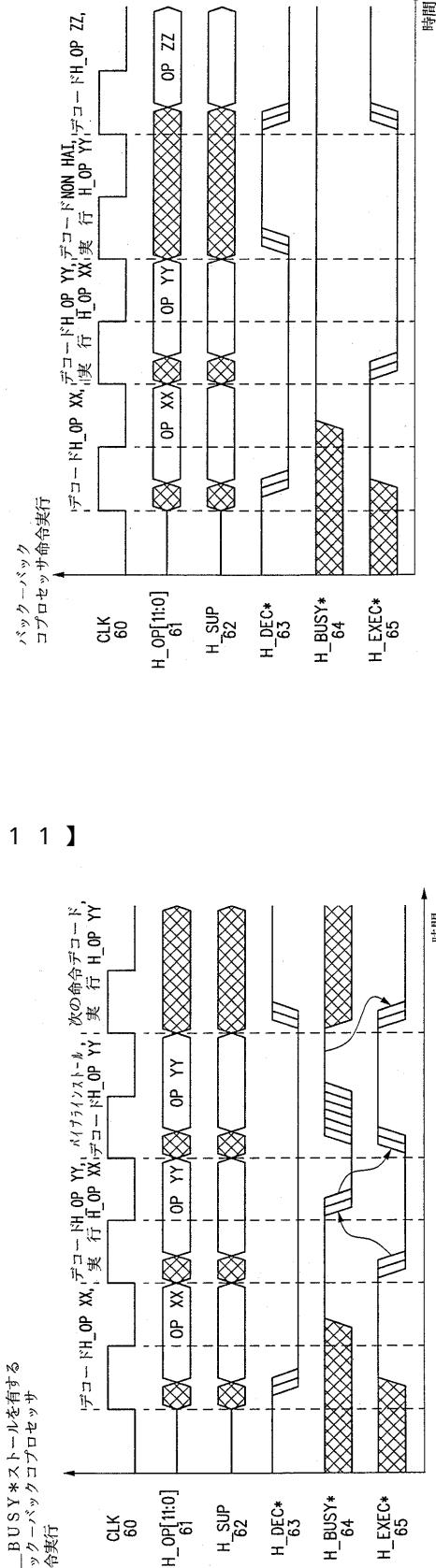
【図7】



【 図 8 】



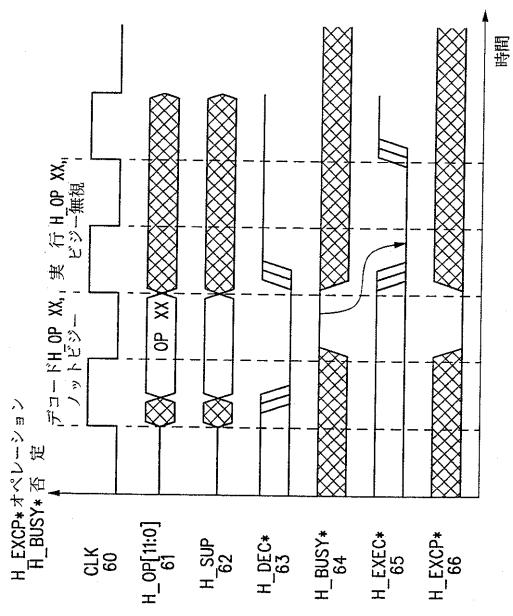
〔 図 9 〕



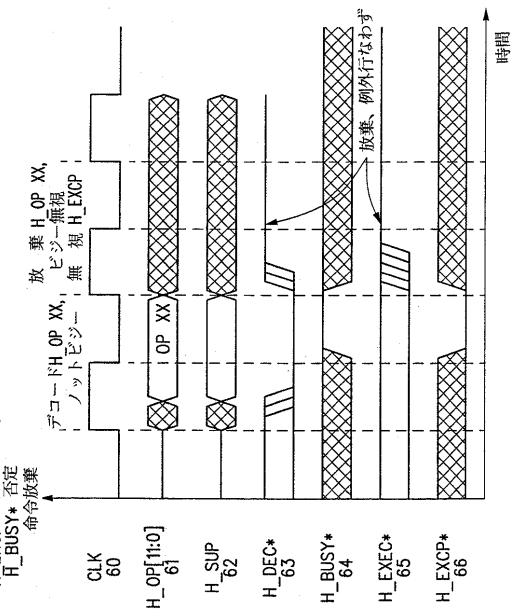
〔 10 〕

【 111 】

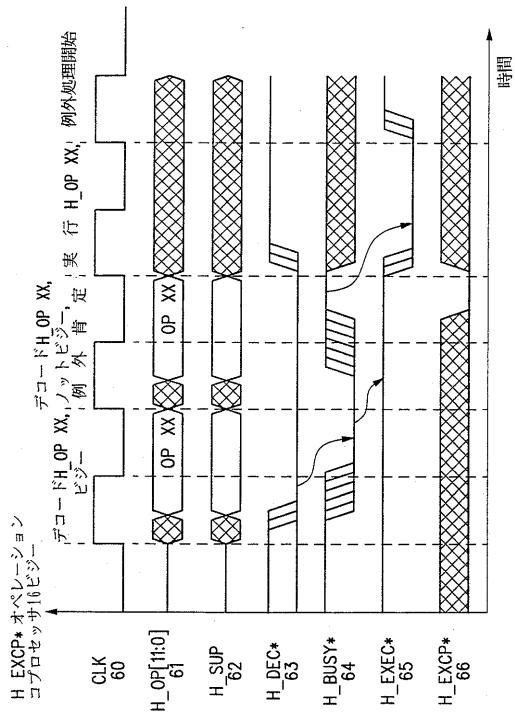
【図 1-2】



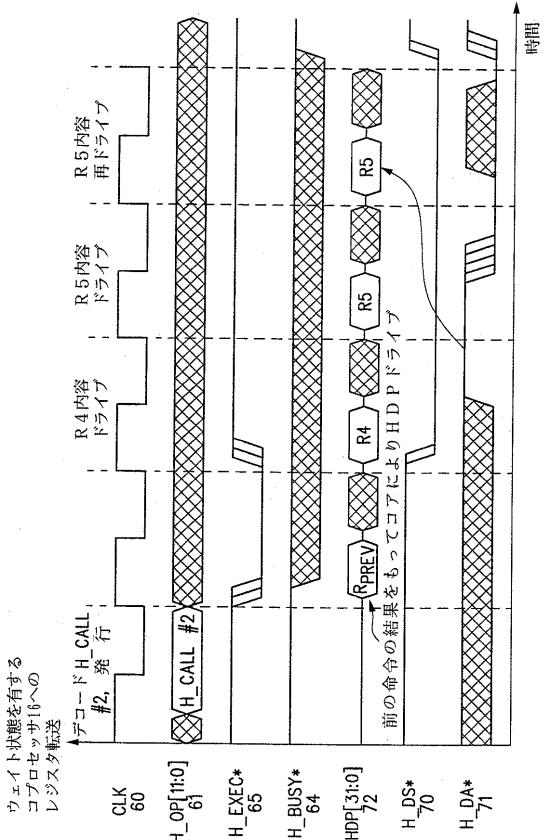
【図 1-3】



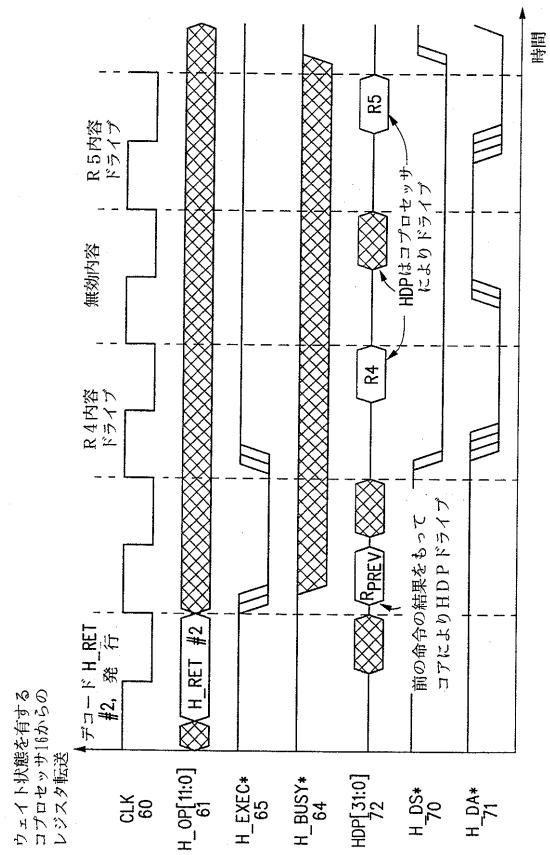
【図 1-4】



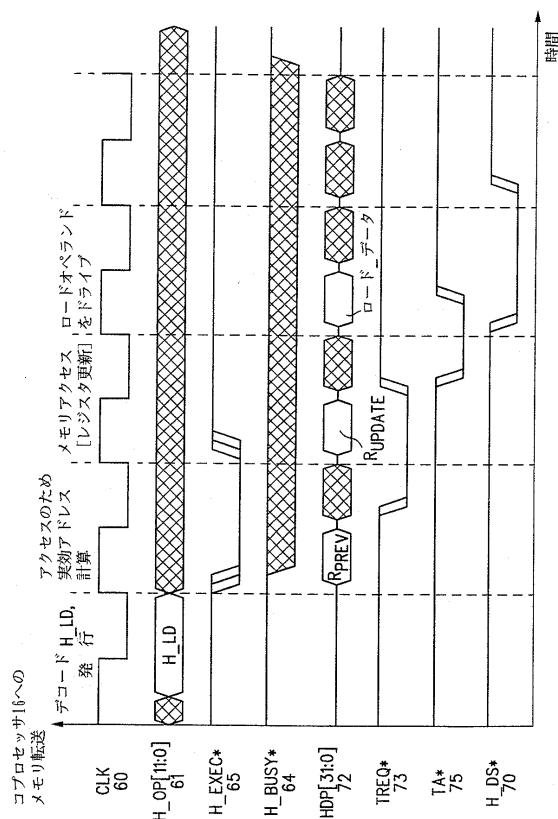
【図 1-5】



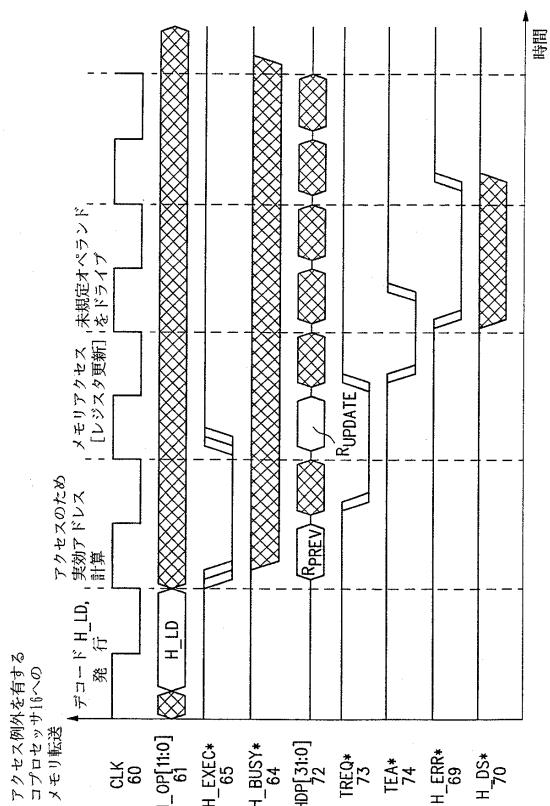
【図16】



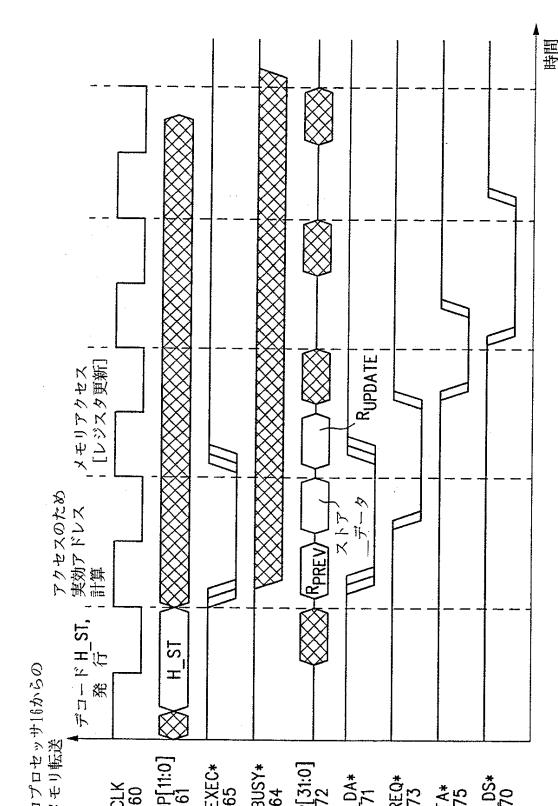
【図17】



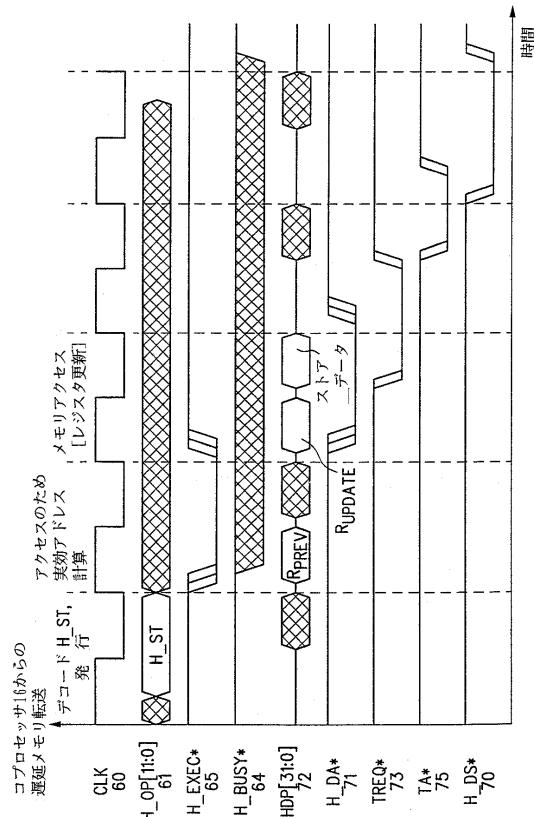
【図18】



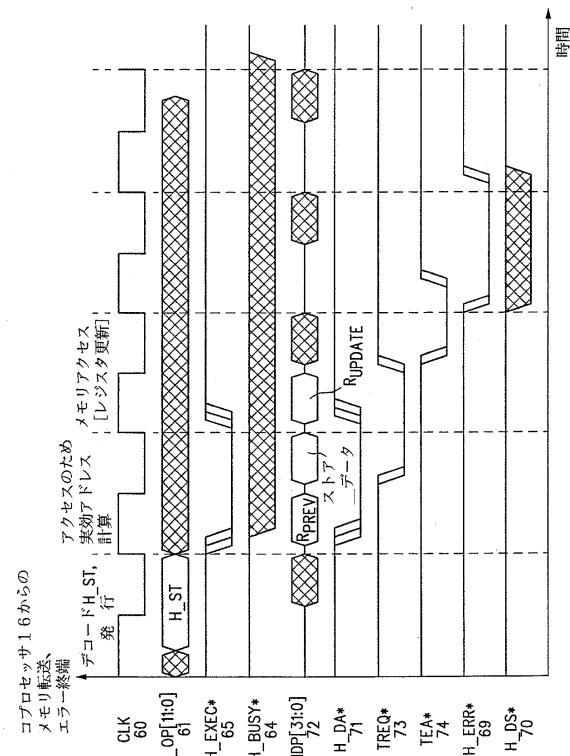
【図19】



【 図 2 0 】



【 図 2 1 】



【 図 2 2 】

H_CALL ハードウェアアクセラレータ(コプロセッサ)コールプリミティブ

動作： ハードウェアアクセラレータへパラメータを渡す

アセンブリ
シンタクス: H_CALL #UU, R4-RLAST, #CODE

記述: H_CALL は1組のレジスタをベースとしたパラメータおよびコードをハードウェアブロック(コプロセッサ) #UU に渡す

条件コード: 影響なし

命令フォーマット:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	UU	0	1	1		CNT					コード	

命令フィールド:

- UU フィールド-ハードウェアブロック(コプロセッサ)を特定
 - 00 - ブロック 0
 - 01 - ブロック 1
 - 10 - ブロック 2
 - 11 - ブロック 3
- CNT フィールド-R4 で始まり、渡すべきレジスタの数を特定
 - 000 - リザーブ、使用しない
 - 001 - パス R4
 - ⋮
 - 111 - パス R4-R10

【 図 2 3 】

【 図 2 4 】

【 図 25 】

【図26】

フロントページの続き

(72)発明者 ジョン・アレンズ

アメリカ合衆国テキサス州78748、オースチン、シャッケルフォード・ドライブ 10709

(72)発明者 ジェフリー・ダブリュ・スコット

アメリカ合衆国テキサス州78759、オースチン、ストーンレイク・ブルバード 9801

#1334

審査官 石川 正二

(56)参考文献 特開平02-294830(JP, A)

特開平08-069377(JP, A)

特開平04-032954(JP, A)

特開平01-118954(JP, A)

特開平01-240932(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 15/16 620