

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6227621号
(P6227621)

(45) 発行日 平成29年11月8日(2017.11.8)

(24) 登録日 平成29年10月20日(2017.10.20)

(51) Int.Cl.			F I		
G06F	9/318	(2006.01)	G06F	9/30	320C
G06F	9/30	(2006.01)	G06F	9/30	310B
G06F	9/305	(2006.01)	G06F	9/30	350F
G06F	9/315	(2006.01)	G06F	9/30	340E
G06F	9/38	(2006.01)	G06F	9/30	340D

請求項の数 24 (全 47 頁) 最終頁に続く

(21) 出願番号	特願2015-251651 (P2015-251651)	(73) 特許権者	591003943 インテル・コーポレーション
(22) 出願日	平成27年12月24日(2015.12.24)		アメリカ合衆国 95054 カリフォル ニア州・サンタクララ・ミッション カレ ッジ ブレーバード・2200
(62) 分割の表示	特願2014-28261 (P2014-28261) の分割	(74) 代理人	110000877 龍華国際特許業務法人
原出願日	平成26年2月18日(2014.2.18)	(72) 発明者	ロクツキン、マキシム
(65) 公開番号	特開2016-103280 (P2016-103280A)		アメリカ合衆国 95054 カリフォル ニア州・サンタクララ・ミッション カレ ッジ ブレーバード・2200 インテル ・コーポレーション内
(43) 公開日	平成28年6月2日(2016.6.2)		
審査請求日	平成28年3月29日(2016.3.29)		
(31) 優先権主張番号	13/842, 754		
(32) 優先日	平成25年3月15日(2013.3.15)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 複数の試験ソースに対するOR試験及びAND試験機能を提供するために命令をフェージングする方法及び装置

(57) 【特許請求の範囲】

【請求項1】

プロセッサ内において命令をフェージングする方法であって、

第1オペランドデスティネーションを規定する第1論理命令と、第2オペランドソースを規定する非破壊的な第2論理命令と、分岐条件を規定する第3命令とを含む複数の命令をフェッチし、

前記複数の命令の一部分を単一のマイクロ演算にフェージングすることを含み、

前記部分は、前記第1オペランドデスティネーションと前記第2オペランドソースが同一であり、且つ、前記分岐条件が前記非破壊的な第2論理命令に依存している場合には、前記第1論理命令及び前記非破壊的な第2論理命令の両方を含む方法。

【請求項2】

前記部分は、前記第1オペランドデスティネーションと前記第2オペランドソースが同一であり、前記非破壊的な第2論理命令及び前記第3命令が連続しており、且つ、前記非破壊的な第2論理命令が試験命令である場合に、前記第1論理命令、前記非破壊的な第2論理命令、及び前記第3命令を含む請求項1に記載の方法。

【請求項3】

前記第1論理命令は、論理OR命令である請求項2に記載の方法。

【請求項4】

前記第1論理命令は、論理AND命令である請求項2に記載の方法。

【請求項5】

10

20

前記部分は、前記第 1 オペランドデスティネーションと前記第 2 オペランドソースが同一であり、前記非破壊的な第 2 論理命令及び前記第 3 命令が連続しており、且つ、前記非破壊的な第 2 論理命令が非破壊的な論理 AND 命令である場合に、前記第 1 論理命令、前記非破壊的な第 2 論理命令、及び前記第 3 命令を含む請求項 1 に記載の方法。

【請求項 6】

前記部分は、前記第 1 オペランドデスティネーションと前記第 2 オペランドソースが同一であり、前記非破壊的な第 2 論理命令及び前記第 3 命令が連続しており、且つ、前記第 1 論理命令が論理 OR 命令である場合に、前記第 1 論理命令、前記非破壊的な第 2 論理命令、及び前記第 3 命令を含む請求項 1 から 3 のいずれか 1 項に記載の方法。

【請求項 7】

前記部分は、前記第 1 オペランドデスティネーションと前記第 2 オペランドソースが同一であり、前記非破壊的な第 2 論理命令及び前記第 3 命令が連続しており、且つ、前記第 1 論理命令が論理 AND 命令である場合に、前記第 1 論理命令、前記非破壊的な第 2 論理命令、及び前記第 3 命令を含む請求項 1、2 および 4 のいずれか 1 項に記載の方法。

【請求項 8】

第 1 オペランドデスティネーションを規定する第 1 論理命令と、第 2 オペランドソースを規定する非破壊的な第 2 論理命令と、分岐条件を規定する第 3 命令とを含む複数の命令をフェッチするための第 1 パイプラインステージと、

前記複数の命令の一部分を単一の第 1 マイクロ演算として復号化するための第 2 パイプラインステージと

を有し、

前記部分は、前記第 1 オペランドデスティネーションと前記第 2 オペランドソースが同一であり、且つ、前記分岐条件が前記非破壊的な第 2 論理命令に依存している場合に、前記第 1 論理命令及び前記非破壊的な第 2 論理命令の両方を含むプロセッサ。

【請求項 9】

前記部分は、前記第 1 オペランドデスティネーションと前記第 2 オペランドソースが同一であり、且つ、前記非破壊的な第 2 論理命令及び前記第 3 命令が連続しており、且つ、前記非破壊的な第 2 論理命令が試験命令である場合に、前記第 1 論理命令、前記非破壊的な第 2 論理命令、及び前記第 3 命令を含む請求項 8 に記載のプロセッサ。

【請求項 10】

前記第 1 論理命令は、論理 OR 命令である請求項 9 に記載のプロセッサ。

【請求項 11】

前記第 1 論理命令は、論理 AND 命令である請求項 9 に記載のプロセッサ。

【請求項 12】

前記部分は、前記第 1 オペランドデスティネーションと前記第 2 オペランドソースが同一であり、前記非破壊的な第 2 論理命令が試験命令であり、且つ、前記第 1 論理命令が論理 OR 命令である場合に、前記第 1 論理命令及び前記非破壊的な第 2 論理命令の両方を含む請求項 8 に記載のプロセッサ。

【請求項 13】

前記分岐条件が前記非破壊的な第 2 論理命令に依存している場合に、前記単一の第 1 マイクロ演算と前記第 3 命令の第 2 マイクロ演算をフュージングするためのマイクロフュージョン論理を含む第 3 パイプラインステージを有する請求項 12 に記載のプロセッサ。

【請求項 14】

前記部分は、前記第 1 オペランドデスティネーションと前記第 2 オペランドソースが同一であり、前記非破壊的な第 2 論理命令及び前記第 3 命令が連続しており、且つ、前記第 1 論理命令が論理 OR 命令である場合に、前記第 1 論理命令、前記非破壊的な第 2 論理命令、及び前記第 3 命令を含む請求項 8 に記載のプロセッサ。

【請求項 15】

前記部分は、前記第 1 オペランドデスティネーションと前記第 2 オペランドソースが同一であり、前記非破壊的な第 2 論理命令及び前記第 3 命令が連続しており、且つ、前記第

10

20

30

40

50

1 論理命令が論理AND命令である場合に、前記第1 論理命令、前記非破壊的な第2 論理命令、及び前記第3命令を含む請求項8に記載のプロセッサ。

【請求項16】

プロセッサ内において命令をフュージングするシステムであって、
複数の命令を保存するためのメモリと、
プロセッサと、
を有し、

前記プロセッサは、
第1オペランドデスティネーションを規定する第1 論理命令と、第2オペランドソースを規定する非破壊的な第2 論理命令と、分岐条件を規定する第3命令とを含む前記複数の命令をフェッチするための第1パイプラインステージと、

前記複数の命令の一部分を単一の第1マイクロ演算として復号化するための第2パイプラインステージと

を有し、

前記部分は、前記第1オペランドデスティネーションと前記第2オペランドソースが同一であり、且つ、前記分岐条件が、前記非破壊的な第2 論理命令に依存する場合に、前記第1 論理命令及び前記非破壊的な第2 論理命令の両方を含むシステム。

【請求項17】

前記部分は、前記第1オペランドデスティネーションと前記第2オペランドソースが同一であり、前記非破壊的な第2 論理命令及び前記第3命令が連続しており、且つ、前記非破壊的な第2 論理命令が試験命令である場合に、前記第1 論理命令、前記非破壊的な第2 論理命令、及び前記第3命令を含む請求項16に記載のシステム。

【請求項18】

前記第1 論理命令は、論理OR命令である請求項17に記載のシステム。

【請求項19】

前記第1 論理命令は、論理AND命令である請求項17に記載のシステム。

【請求項20】

前記プロセッサは、前記分岐条件が前記非破壊的な第2 論理命令に依存している場合に、前記第3命令の第2マイクロ演算を前記単一の第1マイクロ演算とフュージングするためのマイクロフュージョン論理を含む第3パイプラインステージを有する請求項16に記載のシステム。

【請求項21】

ソースデータオペランド並びにそのデスティネーションデータオペランドとして第1データオペランドを規定し、第2ソースデータオペランドを規定する第1 論理命令と、その複数のソースデータオペランドのうちの1つとして前記第1データオペランドを規定し、第3ソースデータオペランドを規定する非破壊的な第2 論理命令と、分岐ターゲットを規定する第3命令と、を復号化するための復号化ステージであって、単一のフュージングされたマイクロ演算としての実行のために前記第1 論理命令及び前記非破壊的な第2 論理命令を前記第3命令とフュージングするための復号化ステージと、

前記単一のフュージングされたマイクロ演算に回答し、前記第1データオペランド及び前記第2ソースデータオペランドからのデータの間において第1論理演算を実行し、前記第3ソースデータオペランドからのデータ及び前記第1論理演算の結果の間において第2論理演算を実行して条件フラグを設定し、且つ、前記条件フラグがどのように設定されたかに応じて前記分岐ターゲットに対する条件付き分岐を実行するための1つ又は複数の実行ユニットと、

を有するプロセッサ。

【請求項22】

前記第2論理演算を実行するステップは、論理TEST演算を有し、且つ、条件付きでゼロフラグを設定する請求項21に記載のプロセッサ。

【請求項23】

10

20

30

40

50

前記第 1 論理命令に従って、前記第 1 データオペランド及び前記第 2 ソースデータオペランドからのデータの間において論理 A N D が実行される請求項 2 2 に記載のプロセッサ。

【請求項 2 4】

前記第 1 論理命令に従って、前記第 1 データオペランド及び前記第 2 ソースデータオペランドからのデータの間において論理 O R が実行される請求項 2 2 に記載のプロセッサ。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、プロセッサ又はその他の処理論理によって実行された際に、論理的、数学的、又はその他の機能的な演算を実行する処理論理、マイクロプロセッサ、及び関連する命令セットアーキテクチャの分野に関する。具体的には、本開示は、複数の試験ソースに対して O R 試験及び A N D 試験機能を提供するために命令をフュージングする方法及び装置に関する。

10

【背景技術】

【0002】

最近のプロセッサは、処理性能を改善すると共に / 又は特殊な命令を利用するためにマイクロアーキテクチャ構造及び技法を含む場合がある。これらのマイクロアーキテクチャ構造及び技法は、例えば、アウトオブオーダー実行、複数のマクロ命令を 1 つのフュージングされた命令にフュージングするデコーダ、マイクロ命令もしくはマイクロ演算（又は

20

【0003】

命令のフュージョンとは、2 つの命令を、実行時におけるプロセッサ内の 1 つの演算、マイクロ演算、シーケンスを結果的にもたらず単一の命令として合成する動的なプロセスである。プロセッサ命令キュー（Instruction Queue: IQ）内に保存されている命令は、IQ からの読出しの後に、且つ、命令デコーダへの送信の前に、或いは、命令デコーダによる復号化の後に、「フュージング」してもよい。通常、命令の復号化の前に行われる命令のフュージョンを「マクロフュージョン」と呼び、命令の（例えば、 $\mu o p$ としての）復号化の後に行われる命令フュージョンを「マイクロフュージョン」と呼んでいる。マクロフュージョンの一例は、比較（「C M P」）命令又は試験命令（「T E S T」）（「C M P / T E S T」）を条件付きジャンプ（「J C C」）命令と合成するというものである。C M P / T E S T と J C C の命令ペアは、例えば、比較を実行し、且つ、比較の結果に基づいて、分岐を実行するか又は実行しないなどのように、ループの末尾においてプログラム内において規則的に発生することになる。マクロフュージョンは、命令のスループットを事実上増大させることになることから、命令をフュージングする機会を可能な限り多く見出すことが望ましいであろう。

30

【0004】

クライアントサイドスクリプトによって、例えば、ユーザーとやり取りし、ブラウザを制御し、非同期式で通信し、且つ、文書コンテンツが表示されるのに伴って文書コンテンツを変更してもよいように、1 つ又は複数のインタープリタ型コンピュータプログラム言語のサポートがウェブブラウザに含まれている場合がある。例えば、J a v a S c r i p t（登録商標）、J S c r i p t、又は E C M A S c r i p t などのインタープリタ型コンピュータプログラム又はスクリプト言語は、動的であると共に弱いタイピング型（w e a k l y t y p e d）である場合があり、且つ、オブジェクト指向の命令型であると共に機能的なプログラミングスタイルをサポートしている場合がある。このようなインタープリタ型コンピュータプログラム言語は、インタープリタ型言語を機械命令のシーケンスとして動的にコンパイルする J I T（J u s t - I n - T i m e）コンパイラの形態におけるブラウザサポートを見出す場合がある。J I T コンパイラは、保存されている値を

40

50

処理の高速化を目的として最適化する場合があるが、この場合には、このような最適化のために、実行時にオブジェクトのタイプを動的に試験する必要がある。これらの動的な試験は、更なる処理のオーバーヘッドを導入し、これにより、相対的に大きなエネルギー消費量が必要になると共にそのような最適化に伴う任意の性能上の利点が制限される可能性がある。

【発明の概要】

【0005】

これまでのところ、このような性能の制限の問題、エネルギー消費の懸念、及びその他の実行時のボトルネックに対する潜在的な解決策が十分に探求されていない。

【0006】

本発明は、限定ではなく、一例として、添付図面の各図に示されている。

【図面の簡単な説明】

【0007】

【図1A】複数の試験ソースに対するOR試験及びAND試験機能を提供するために命令を実行するシステムの一実施形態のブロック図である。

【図1B】複数の試験ソースに対するOR試験及びAND試験機能を提供するために命令を実行するシステムの別の実施形態のブロック図である。

【図1C】複数の試験ソースに対するOR試験及びAND試験機能を提供するために命令を実行するシステムの別の実施形態のブロック図である。

【図2】複数の試験ソースに対するOR試験及びAND試験機能を提供するために命令を実行するプロセッサの一実施形態のブロック図である。

【図3A】一実施形態によるパック型データタイプを示す図である。

【図3B】一実施形態によるパック型データタイプを示す図である。

【図3C】一実施形態によるパック型データタイプを示す図である。

【図3D】一実施形態による複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令の符号化を示す図である。

【図3E】別の実施形態による複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令の符号化を示す図である。

【図3F】別の実施形態による複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令の符号化を示す図である。

【図3G】別の実施形態による複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令の符号化を示す図である。

【図3H】別の実施形態による複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令の符号化を示す図である。

【図4A】複数の試験ソースに対するOR試験及びAND試験機能を提供する命令を実行するためのプロセッサマイクロアーキテクチャの一実施形態の要素を示す図である。

【図4B】複数の試験ソースに対するOR試験及びAND試験機能を提供する命令を実行するためのプロセッサマイクロアーキテクチャの別の実施形態の要素を示す図である。

【図5】複数の試験ソースに対するOR試験及びAND試験機能を提供する命令を実行するためのプロセッサの一実施形態のブロック図である。

【図6】複数の試験ソースに対するOR試験及びAND試験機能を提供する命令を実行するためのコンピュータシステムの一実施形態のブロック図である。

【図7】複数の試験ソースに対するOR試験及びAND試験機能を提供する命令を実行するためのコンピュータシステムの別の実施形態のブロック図である。

【図8】複数の試験ソースに対するOR試験及びAND試験機能を提供する命令を実行するためのコンピュータシステムの別の実施形態のブロック図である。

【図9】複数の試験ソースに対するOR試験及びAND試験機能を提供する命令を実行するためのシステムオンチップの一実施形態のブロック図である。

【図10】複数の試験ソースに対するOR試験及びAND試験機能を提供する命令を実行するためのプロセッサの一実施形態のブロック図である。

10

20

30

40

50

【図11】複数の試験ソースに対するOR試験及びAND試験機能を提供するIPコア開発システムの一実施形態のブロック図である。

【図12】複数の試験ソースに対するOR試験及びAND試験機能を提供するアーキテクチャエミュレーションシステムの一実施形態を示す図である。

【図13】複数の試験ソースに対するOR試験及びAND試験機能を提供する命令を変換するためのシステムの一実施形態を示す図である。

【図14】複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令をフュージングするためのプロセッサマイクロアーキテクチャの代替実施形態の要素を示す図である。

【図15A】複数の試験ソースに対するOR試験及びAND試験機能を提供するためのフュージングされた命令構造の一実施形態のフロー図である。

10

【図15B】複数の試験ソースに対するOR試験及びAND試験機能を提供するためのフュージングされた命令構造の代替実施形態のフロー図である。

【図15C】複数の試験ソースに対するOR試験及びAND試験機能を提供するためのフュージングされた命令構造の別の代替実施形態のフロー図である。

【図16A】複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令をフュージングするためのプロセスの一実施形態のフロー図である。

【図16B】複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令をフュージングするためのプロセスの代替実施形態のフロー図である。

【図16C】複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令をフュージングするためのプロセスの別の代替実施形態のフロー図である。

20

【図16D】複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令をフュージングするためのプロセスの別の代替実施形態のフロー図である。

【図17】複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令を実行するためのプロセスの一実施形態のフロー図である。

【発明を実施するための形態】

【0008】

以下の説明は、プロセッサ、コンピュータシステム、又はその他の処理装置内において又はこれらとの関連において、複数の試験ソースに対するOR試験及びAND試験機能を提供するためのフュージング可能な命令及び論理について開示している。

30

【0009】

ウェブブラウザは、例えば、JavaScript（登録商標）、JScript、又はECMAScriptなどの1つ又は複数のインタープリタ型コンピュータプログラム言語のサポートを含んでいる場合があり、これらは、動的であると共に弱いタイピング型であってもよく、且つ、オブジェクト指向の命令型であり且つ機能的なプログラミングスタイルをサポートしてもよい。このようなインタープリタ型コンピュータプログラム言語は、インタープリタ型言語を機械命令のシーケンスとして動的にコンパイルするJITコンパイラの形態におけるブラウザのサポートを見出す場合がある。JITコンパイラは、保存されている値を処理の高速化を目的として最適化することになるが、この場合には、このような最適化のために、実行時におけるオブジェクトのタイプの動的な試験が必要となる場合がある。例えば、JITコンパイラは、機械語の1つ又は複数のビットを使用することにより、値が、相対的に一般的な浮動小数点値又はストリングではなく、最適化された整数であることを弁別すると共に、/又は表してもよい。これらの1つ又は複数のビットの存在は、値にアクセスする前に、実行時に動的にチェックすることができる。このような動的なタイプ試験は、頻繁に実行されるタイプ試験命令シーケンスのイディオムの形態の更なる処理オーバーヘッドを導入し、これにより、相対的に大きなエネルギー消費量が必要になると共にそのようなコンパイラ最適化の任意の性能上の利点が制限される可能性がある。

40

【0010】

これらの頻繁に実行される命令シーケンスのイディオムは、2つのソースデータオペラ

50

ンドを合成するための論理演算（例えば、いずれかのソース内のビットが1であるかどうかを試験する際のOR、いずれかのソース内のビットがゼロであるかどうかを試験する際のAND）と、比較又は試験命令（例えば、マスクに照らして論理演算の結果をTESTするためのもの）と、条件付き分岐（例えば、ゼロフラグがTESTによって設定されている／設定されていない場合に相対的に低速の一般的なコードシーケンスにジャンプするためのJZ/JNZなど）と、を含む場合がある。プロセッサ復号化論理は、比較命令又は試験命令（例えば、TEST）を条件付き分岐命令（例えば、JZ）と合成する際に、マクロフュージョンを利用してもよい。

【0011】

複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令をフュージングする方法及び装置の実施形態が開示される。いくつかの実施形態は、フェッチ命令を含み、前記命令は、第1オペランドデスティネーションを規定する第1命令と、第2オペランドソースを規定する第2命令と、分岐条件を規定する第3命令と、を含む。複数の命令の一部が、単一のマイクロ演算としてフュージングされ、この部分は、前記第1オペランドデスティネーションと前記第2オペランドソースが同一であり、且つ、前記分岐条件が第2命令に依存している場合に、第1及び第2命令の両方を含む。いくつかの実施形態は、1つの論理命令を従来技術の試験命令とフュージングすることにより、新しい試験命令を生成する。その他の実施形態は、動的に、但し、JITコンパイラを通じて、試験命令を生成する。又、いくつかの実施形態は、試験命令を後続の条件付き分岐命令（例えば、JZ）とフュージングし、且つ、条件フラグが設定された方式に従って分岐を実行する。従って、頻繁に実行される命令シーケンスのイディオムは、実行時においてプロセッサ内における1つのマイクロ演算シーケンスを結果的にもたらず単一の命令としてフュージング可能であってもよい。

【0012】

複数の試験ソースに対するOR試験及び／又はAND試験機能を提供するためのフュージング可能な命令及び論理は、ほとんど、性能の制限の問題、エネルギー消費の懸念、及びJITコンパイラによって生成される頻繁に実行されるタイプ試験命令のイディオムと関連するその他の実行時のボトルネックを除去することになることを理解されたい。

【0013】

以下の説明には、本発明の実施形態の相対的に十分な理解を提供するべく、処理論理、プロセッサタイプ、マイクロアーキテクチャ条件、イベント、イネーブルメントメカニズム、及びこれらに類似したものなどの多数の具体的な詳細が記述されている。但し、当業者には、これらの具体的な詳細を伴うことなしに、本発明を実施してもよいことが理解されよう。更には、本発明の実施形態を不必要に曖昧にすることを回避するべく、いくつかの周知の構成、回路、及びこれらに類似したものについては、詳細な図示が省略されている。

【0014】

以下の実施形態は、プロセッサを参照して記述されているが、その他の実施形態は、その他のタイプの集積回路及び論理装置に適用可能である。本発明の実施形態の類似の技法及び教示は、更に大きなパイプラインスループット及び改善された性能の利益を享受することができるその他のタイプの回路又は半導体装置に適用することができる。本発明の実施形態の教示内容は、データ操作を実行する任意のプロセッサ又は機械に対して適用可能である。但し、本発明は、512ビット、256ビット、128ビット、64ビット、32ビット、又は16ビットのデータ演算を実行するプロセッサ又は機械に限定されるものではなく、且つ、データの操作又は管理を実行する任意のプロセッサ及び機械に対して適用することができる。更には、以下の記述内容は、例を提供するものであり、且つ、添付図面は、例示を目的として、様々な例を示している。但し、これらの例は、本発明の実施形態の可能な実装形態のすべてを網羅したリストの提供ではなく、本発明の実施形態の例を提供することを目的としたものに過ぎないことから、これらの例は、限定の意味において解釈してはならない。

10

20

30

40

50

【0015】

以下の例は、実行ユニット及び論理回路の文脈において命令の処理及び配布について記述しているが、本発明のその他の実施形態は、機械によって実行された際に本発明の少なくとも1つの実施形態による機能を機械に実行させる有体の機械可読媒体上に保存されたデータ及び/又は命令によって実現することができる。一実施形態においては、本発明の実施形態と関連する機能は、機械実行可能命令として実施されている。命令を使用することにより、命令によってプログラムされた汎用又は専用プロセッサに本発明のステップを実行させることができる。本発明の実施形態は、コンピュータプログラムプロダクト又はソフトウェアとして提供してもよく、このコンピュータプログラムプロダクト又はソフトウェアは、本発明の実施形態に従って1つ又は複数の演算を実行するようにコンピュータ（又は、その他の電子装置）をプログラムするべく使用してもよい命令をその上部に保存された状態において有する機械又はコンピュータ可読媒体を含んでもよい。或いは、この代わりに、本発明の実施形態のステップは、ステップを実行するための固定機能論理を含む特定のハードウェアコンポーネントにより、或いは、プログラムされたコンピュータコンポーネントと固定機能ハードウェアコンポーネントの任意の組合せにより、実行してもよいであろう。

10

【0016】

本発明の実施形態を実行するように論理をプログラムするべく使用される命令は、DRAM、キャッシュ、フラッシュメモリ、又はその他のストレージなどのシステム内のメモリに保存することができる。更には、命令は、ネットワークを介して、又はその他のコンピュータ可読媒体により、配布することができる。従って、機械可読媒体は、限定を伴うことなしに、フロッピー（登録商標）ディスク、光ディスク、コンパクトディスク、読出し専用メモリ（CD-ROM）、並びに、磁気光ディスク、読出し専用メモリ（ROM）、ランダムアクセスメモリ（RAM）、EPROM（Erasable Programmable Read-Only Memory）、EEPROM（Electrically Erasable Programmable Read-Only Memory）、磁気又は光カード、フラッシュメモリ、又は伝播する信号の電氣的、光学的、音響的、又はその他の形態（例えば、搬送波、赤外線信号、デジタル信号など）を介したインターネット上における情報の伝送において使用される有体の機械可読ストレージなどの機械（例えば、コンピュータ）によって読み取り可能な形態で情報を保存又は伝送するための任意のメカニズムを含んでもよい。従って、コンピュータ可読媒体は、電子的な命令又は情報を機械（例えば、コンピュータ）によって読み取り可能な形態で保存又は送信するのに適した任意のタイプの有体の機械可読媒体を含む。

20

30

【0017】

設計は、生成からシミュレーションを経て製造にまで至る様々な段階を経ることになる。設計を表すデータは、設計をいくつかの方式によって表してもよい。第1に、シミュレーションにおいて有用なように、ハードウェア記述言語又は別の機能記述言語を使用してハードウェアを表してもよい。更には、論理及び/又はトランジスタゲートを有する回路レベルのモデルを設計プロセスのいくつかの段階において生成してもよい。更には、大部分の設計は、なんらかの段階において、ハードウェアモデル内における様々な装置の物理的な配置を表す所定のレベルのデータに到達する。従来の半導体製造法を使用する場合には、ハードウェアモデルを表すデータは、集積回路を製造するために使用されるマスクの異なるマスク層上における様々な機能の存在又は不存在を規定するデータであってもよい。設計の任意の表現においては、データは、任意の形態の機械可読媒体内に保存してもよい。メモリ又はディスクなどの磁氣的又は光学的ストレージは、このような情報を送信するべく変調又はその他の方法で生成された光学的又は電氣的な波を介して伝送される情報を保存するための機械可読媒体であってもよい。電氣的信号の複写、バッファリング、又は再伝送が実行される程度に符号又は設計を通知又は搬送する電氣的な搬送波が伝送される際には、新しい複写が生成される。従って、通信プロバイダ又はネットワークプロバイダは、本発明の実施形態の技法を実施する搬送波として符号化された情報などの項目を有

40

50

体の機械可読媒体上に少なくとも一時的に保存してもよい。

【0018】

最近のプロセッサにおいては、いくつかの異なる実行ユニットを使用して様々なコード及び命令を処理及び実行している。いくつかのものは、相対的に迅速に完了し、その他のものは、完了するのにいくつかのクロックサイクルを要する可能性があることから、すべての命令が等しく生成されるわけではない。命令のスループットが大きいほど、プロセッサの全体的な性能が良好になる。従って、可能な限り高速で多くの命令を実行させることが有利であろう。但し、相対的に大きな複雑性を有すると共に実行時間及びプロセッサリソースの観点において相対的に多くのものを必要とする特定の命令が存在している。例えば、浮動小数点命令、ロード/ストア動作、データ移動などが存在している。

10

【0019】

益々多くのコンピュータシステムがインターネット、テキスト、及びマルチメディアアプリケーションにおいて使用されようになるのに伴って、時間と共に更なるプロセッササポートが導入されている。一実施形態においては、命令セットは、データタイプ、命令、レジスタアーキテクチャ、アドレス指定モード、メモリアーキテクチャ、割込み及び例外処理、及び外部入出力(I/O)を含む1つ又は複数のコンピュータアーキテクチャと関連するものであってもよい。

【0020】

一実施形態においては、命令セットアーキテクチャ(Instruction Set Architecture: ISA)は、1つ又は複数の命令セットを実装するために使用されるプロセッサ論理及び回路を含む1つ又は複数のマイクロアーキテクチャによって実装してもよい。従って、異なるマイクロアーキテクチャを有するプロセッサが共通命令セットの少なくとも一部分を共有することができる。例えば、Intel(登録商標) Pentium(登録商標) 4プロセッサ、Intel Core(商標)プロセッサ、及びカリフォルニア州Sunnyvaleに所在するAdvanced Micro Devices, Inc.社からのプロセッサは、略同一のバージョンのx86命令セット(相対的に新しいバージョンに伴って追加されたいくつかの拡張を有するもの)を実装しているが、異なる内部設計を有する。同様に、ARMホールディングス社、MIPS社、又は彼らのライセンシー又は採用者などのその他のプロセッサ開発会社によって設計されたプロセッサも、共通命令セットの少なくとも一部分を共有してもよいが、異なるプロセッサ設計を含んでもよい。例えば、専用の物理レジスタ、レジスタリネームメカニズム(例えば、RAT(Register Alias Table)の使用)を使用する1つ又は複数の動的に割り当てられた物理レジスタ、ROB(Reorder Buffer)、及びリタイアメントレジスタファイルを含む新しい又は周知の技法を使用する異なるマイクロアーキテクチャにおいて異なる方法でISAの同一のレジスタアーキテクチャを実装してもよい。一実施形態においては、レジスタは、ソフトウェアプログラムによるアドレス指定が可能であっても又は可能でなくてもよい1つ又は複数のレジスタ、レジスタアーキテクチャ、レジスタファイル、又はその他のレジスタセットを含んでもよい。

20

30

【0021】

一実施形態においては、命令は、1つ又は複数の命令フォーマットを含んでもよい。一実施形態においては、命令フォーマットは、例えば、実行を要する演算及びその演算が実行の対象とする1つ又は複数のオペランドを規定するための様々なフィールド(ビットの数やビットの場所など)を含んでもよい。いくつかの命令フォーマットは、命令テンプレート(又は、サブフォーマット)によって定義された状態で、更に分割してもよい。例えば、所与の命令フォーマットの命令テンプレートは、命令フォーマットのフィールドの異なるサブセットを有するように定義してもよいと共に、又は、異なる方式によって解釈される所与のフィールドを有するように定義してもよい。一実施形態においては、命令は、命令フォーマットを使用して(且つ、定義されている場合には、その命令フォーマットの命令テンプレートのうちの所与の命令テンプレートにおいて)表現され、且つ、演算及び

40

50

その演算が操作する対象のオペランドを規定又は通知している。

【0022】

科学的な、財務的な、自己ベクトル化された汎用のRMS (Recognition , Mining , and Synthesis) の視覚的なマルチメディアアプリケーション (例えば、2D / 3Dグラフィックス、画像処理、ビデオ圧縮 / 圧縮解除、音声認識アルゴリズム、及びオーディオ操作) は、同一の演算を多数のデータ項目に対して実行することを要する場合がある。一実施形態においては、SIMD (Single Instruction Multiple Data) は、プロセッサに複数のデータ要素に対して演算を実行させるタイプの命令を意味している。SIMD技術は、レジスタ内のビットをそれぞれが別個の値を表すいくつかの固定サイズ又は可変サイズのデータ要素に論理的に分割することができるプロセッサ内において使用してもよい。例えば、一実施形態においては、64ビットレジスタ内のビットは、それぞれが別個の16ビット値を表す4つの別個の16ビットのデータ要素を収容するソースオペランドとして編成してもよい。このタイプのデータは、「パック型」データタイプ又は「ベクトル」データタイプと呼んでもよく、且つ、このデータタイプのオペランドは、パック型データオペランド又はベクトルオペランドと呼ばれる。一実施形態においては、パック型データ項目又はベクトルは、単一のレジスタ内に保存されたパック型データ要素のシーケンスであってもよく、且つ、パック型データオペランド又はベクトルオペランドは、SIMD命令 (又は、「パック型データ命令」又は「ベクトル命令」) のソース又はデスティネーションオペランドであってもよい。一実施形態においては、SIMD命令は、同一の又は異なる数のデータ要素と共に、且つ、同一の又は異なるデータ要素の順序において、同一の又は異なるサイズのデスティネーションベクトルオペランド (結果ベクトルオペランドとも呼ばれる) を生成するべく、2つのソースベクトルオペランドに対して実行される単一のベクトル演算を規定している。

【0023】

x86、MMX (商標)、SSE (Streaming SIMD Extensions)、SSE2、SSE3、SSE4.1、及びSSE4.2命令を含む命令セットを有するIntel (登録商標) Core (商標) プロセッサ、VFP (Vector Floating Point) 及び / 又はNEON命令を含む命令セットを有するARM Cortex (登録商標) ファミリーのプロセッサなどのARMプロセッサ、並びに、Chinese Academy of SciencesのICT (Institute of Computing Technology) によって開発されたLoongsonファミリーのプロセッサなどのMIPSプロセッサによって利用されているものなどのSIMD技術は、アプリケーション性能の大幅な改善を可能にする (Core (商標) 及びMMX (商標) は、カリフォルニア州Santa Claraに所在するインテル社の登録商標又は商標である)。

【0024】

一実施形態においては、デスティネーション及びソースレジスタ / データは、対応するデータ又は演算のソース及びデスティネーションを表す一般的な用語である。いくつかの実施形態においては、これらは、レジスタ、メモリ、又は図示されているもの以外のその他の名称又は機能を有するその他のストレージエリアによって実装してもよい。例えば、一実施形態においては、「DEST1」は、一時的なストレージレジスタ又はその他のストレージエリアであってもよく、「SRC1」及び「SRC2」は、第1及び第2ソースストレージレジスタ又はその他のストレージエリアであってもよく、以下同様である。その他の実施形態においては、SRC及びDESTストレージエリアのうちの複数のものは、同一のストレージエリア内の異なるデータストレージ要素 (例えば、SIMDレジスタ) に対応してもよい。又、一実施形態においては、例えば、第1及び第2ソースデータに対して実行された演算の結果を、デスティネーションレジスタとして機能する2つのソースレジスタのうちの1つに書き戻すことにより、ソースレジスタのうちの1つがデスティネーションレジスタとして機能してもよい。

【 0 0 2 5 】

図 1 A は、本発明の一実施形態による命令を実行するための実行ユニットを含むプロセッサを有するように形成された例示用のコンピュータシステムのブロック図である。システム 100 は、本明細書に記述されている実施形態と同様に、本発明に従ってデータを処理するためのアルゴリズムを実行する論理を含む実行ユニットを利用するためのプロセッサ 102 などのコンポーネントを含む。システム 100 は、カリフォルニア州 Santa Clara に所在するインテル社から入手可能な P E N T I U M (登録商標) I I I、P E N T I U M (登録商標) 4、X e o n (商標)、I t a n i u m (登録商標)、X S c a l e (商標)、及び/又は S t r o n g A R M (商標) マイクロプロセッサに基づいた処理システムを表しているが、その他のシステム(その他のマイクロプロセッサを有する P C、エンジニアリングワークステーション、セットトップボックス、及びこれらに類似したものを含む)を使用してもよい。一実施形態においては、サンプルシステム 100 は、ワシントン州 Redmond に所在する M i c r o s o f t C o r p o r a t i o n 社から入手可能な所定のバージョンの W I N D O W S (登録商標) オペレーティングシステムを実行してもよいが、その他のオペレーティングシステム(例えば、U N I X (登録商標) 及び L i n u x (登録商標))、埋込み型のソフトウェア、及び/又はグラフィカルユーザーインターフェイスを使用してもよい。従って、本発明の実施形態は、ハードウェア回路及びソフトウェアの任意の特定の組合せに限定されるものではない。

10

【 0 0 2 6 】

実施形態は、コンピュータシステムに限定されるものではない。本発明の代替実施形態は、ハンドヘルド型装置及び埋込み型アプリケーションなどのその他の装置において使用することもできる。ハンドヘルド装置のいくつかの例は、セルラー電話機、インターネットプロトコル装置、デジタルカメラ、パーソナルデジタルアシスタント(PDA)、及びハンドヘルドPCを含む。埋込み型アプリケーションは、マイクロコントローラ、デジタル信号プロセッサ(DSP)、システムオンチップ、ネットワークコンピュータ(NetPC)、セットトップボックス、ネットワークハブ、ワイドエリアネットワーク(WAN)スイッチ、又は少なくとも1つの実施形態に従って1つ又は複数の命令を実行することができる任意のその他のシステムを含むことができる。

20

【 0 0 2 7 】

図 1 A は、本発明の一実施形態に従って少なくとも1つの命令を実行するアルゴリズムを実行するための1つ又は複数の実行ユニット 108 を含むプロセッサ 102 を有するように形成されたコンピュータシステム 100 のブロック図である。一実施形態は、シングルプロセッサのデスクトップ又はサーバーシステムの文脈において記述されることになるが、代替実施形態は、マルチプロセッサシステム内に含むこともできる。システム 100 は、「ハブ」システムアーキテクチャの一例である。コンピュータシステム 100 は、データ信号を処理するためのプロセッサ 102 を含む。プロセッサ 102 は、例えば、C I S C (Complex Instruction Set Computer) マイクロプロセッサ、R I S C (Reduced Instruction Set Computing) マイクロプロセッサ、V L I W (Very Long Instruction Word) マイクロプロセッサ、命令セットの組合せを実装するプロセッサ、又はデジタル信号プロセッサなどの任意のその他のプロセッサ装置であってもよい。プロセッサ 102 は、プロセッサバス 110 に結合されており、プロセッサバス 110 は、プロセッサ 102 とシステム 100 内のその他のシステムコンポーネントの間においてデータ信号を送信することができる。システム 100 の要素は、当業者には周知のそれぞれの従来の機能を実行する。

30

40

【 0 0 2 8 】

一実施形態においては、プロセッサ 102 は、レベル 1 (L1) 内部キャッシュメモリ 104 を含む。アーキテクチャに応じて、プロセッサ 102 は、単一の内部キャッシュ又はマルチレベルの内部キャッシュを有することができる。或いは、この代わりに、別の実施形態においては、キャッシュメモリは、プロセッサ 102 の外部に存在することもでき

50

る。又、その他の実施形態は、特定の実装形態及びニーズに応じて、内部及び外部キャッシュの両方の組合せを含むこともできる。レジスタファイル106は、異なるタイプのデータを、整数レジスタ、浮動小数点レジスタ、状態レジスタ、及び命令ポインタレジスタを含む様々なレジスタ内に保存することができる。

【0029】

又、整数及び浮動小数点演算を実行する論理を含む実行ユニット108も、プロセッサ102内に存在している。又、プロセッサ102は、特定のマイクロ命令のマイクロコードを保存するマイクロコード(μcode)ROMをも含んでいる。一実施形態においては、実行ユニット108は、パック型命令セット109を処理するための論理を含む。命令を実行するための関連する回路と共にパック型命令セット109を汎用プロセッサ102の命令セットに含むことにより、汎用プロセッサ102内のパック型データを使用し、多くのマルチメディアアプリケーションによって使用される演算を実行してもよい。従って、パック型データに対して演算を実行するためのプロセッサのデータバスの幅全体を使用することにより、多くのマルチメディアアプリケーションの速度を向上させると共にそれらのアプリケーションを相対的に効率的に実行することができる。これにより、一度に1つのデータ要素に対して1つ又は複数の演算を実行するためにデータを複数の相対的に小さな単位でプロセッサのデータバスに跨って転送する必要性を除去することができる。

【0030】

又、実行ユニット108の代替実施形態は、マイクロコントローラ、埋込み型プロセッサ、グラフィックス装置、DSP、及びその他のタイプの論理回路内において使用することもできる。システム100は、メモリ120を含む。メモリ120は、ダイナミックランダムアクセスメモリ(DRAM)装置、スタティックランダムアクセスメモリ(SRAM)装置、フラッシュメモリ装置、又はその他のメモリ装置であってもよい。メモリ120は、プロセッサ102によって実行することができるデータ信号によって表された命令及び/又はデータを保存することができる。

【0031】

システム論理チップ116がプロセッサバス110及びメモリ120に結合されている。図示の実施形態におけるシステム論理チップ116は、メモリコントローラハブ(Memory Controller Hub: MCH)である。プロセッサ102は、プロセッサバス110を介してMCH116に対して通信することができる。MCH116は、命令及びデータ保存のために、且つ、グラフィックスコマンド、データ、及びテクスチャの保存のために、メモリ120に対する高帯域幅のメモリ経路118を提供している。MCH116は、データ信号をプロセッサ102、メモリ120、及びシステム100内のその他のコンポーネントの間において移送し、且つ、データ信号をプロセッサバス110、メモリ120、及びシステムI/O122の間において橋絡するためのものである。いくつかの実施形態においては、システム論理チップ116は、グラフィックスコントローラ112に結合するためのグラフィックスポートを提供することができる。MCH116は、メモリインターフェイス118を通じてメモリ120に結合されている。グラフィックスカード112は、AGP(Accelerated Graphics Port)相互接続114を通じてMCH116に結合されている。

【0032】

システム100は、プロプライエタリなI/Oインターフェイスバス122を使用してMCH116をI/Oコントローラハブ(ICH)130に結合している。ICH130は、ローカルI/Oバスを介していくつかのI/O装置に対する直接的な接続を提供している。ローカルI/Oバスは、周辺装置をメモリ120、チップセット、及びプロセッサ102に対して接続するための高速I/Oバスである。いくつかの例は、オーディオコントローラ、ファームウェアハブ(フラッシュBIOS)128、ワイヤレストランシーバ126、データストレージ124、ユーザー入力及びキーボードインターフェイスを含むレガシーI/Oコントローラ、USB(Universal Serial Bus)などのシリアル拡張ポート、及びネットワークコントローラ134である。データストレージ

10

20

30

40

50

装置 1 2 4 は、ハードディスクドライブ、フロッピー（登録商標）ディスクドライブ、C D - R O M 装置、フラッシュメモリ装置、又はその他のマストレージ装置を有することができる。

【 0 0 3 3 】

システムの別の実施形態においては、一実施形態による命令をシステムオンチップと共に使用することができる。システムオンチップの一実施形態は、プロセッサと、メモリと、を有する。このような1つのシステム用のメモリは、フラッシュメモリである。フラッシュメモリは、プロセッサ及びその他のシステムコンポーネントと同一のダイ上に配置することができる。更には、メモリコントローラ又はグラフィックスコントローラなどのその他の論理ブロックをシステムオンチップ上に配置することもできる。

10

【 0 0 3 4 】

図 1 B は、本発明の一実施形態の原理を実装したデータ処理システム 1 4 0 を示している。当業者には、本明細書に記述されている実施形態は、本発明の実施形態の範囲を逸脱することなしに、代替処理システムと共に使用することができることが容易に理解されよう。

【 0 0 3 5 】

コンピュータシステム 1 4 0 は、一実施形態に従って少なくとも1つの命令を実行する能力を有する処理コア 1 5 9 を有する。一実施形態においては、処理コア 1 5 9 は、限定を伴うことなしに、C I S C、R I S C、又はV L I Wタイプのアーキテクチャを含む任意のタイプのアーキテクチャの処理ユニットを表している。又、処理コア 1 5 9 は、1つ又は複数のプロセス技術による製造に適することにもなり、且つ、機械可読媒体上において十分詳細に表現されることにより、前記製造の円滑な実行にも適することになる。

20

【 0 0 3 6 】

処理コア 1 5 9 は、実行ユニット 1 4 2 と、1つ又は複数のレジスタファイルの組 1 4 5 と、デコーダ 1 4 4 と、を有する。又、処理コア 1 5 9 は、本発明の実施形態の理解には不要な更なる回路（図示されてはいない）をも含む。実行ユニット 1 4 2 は、処理コア 1 5 9 が受け取った命令を実行するために使用される。通常のプロセッサ命令を実行するのに加えて、実行ユニット 1 4 2 は、パック型データフォーマットに対する演算を実行するためのパック型命令セット 1 4 3 内の命令を実行することができる。パック型命令セット 1 4 3 は、本発明の実施形態及びその他のパック型命令を実行するための命令を含む。実行ユニット 1 4 2 は、内部バスによってレジスタファイル 1 4 5 に結合されている。レジスタファイル 1 4 5 は、データを含む情報を保存するための処理コア 1 5 9 上のストレージエリアを表している。上述のように、パック型データを保存するために使用されるストレージエリアは、必須ではないことを理解されたい。実行ユニット 1 4 2 は、デコーダ 1 4 4 に結合されている。デコーダ 1 4 4 は、処理コア 1 5 9 が受け取った命令を制御信号及び/又はマイクロコードエントリポイントとして復号化するために使用される。これらの制御信号及び/又はマイクロコードエントリポイントに回答し、実行ユニット 1 4 2 は、適切な演算を実行する。一実施形態においては、デコーダを使用することにより、命令内に示されている対応するデータに対して実行するべき演算を通知することになる命令の `opcode` を解釈している。

30

40

【 0 0 3 7 】

処理コア 1 5 9 は、様々なその他のシステム装置と通信するためのバス 1 4 1 と結合されており、その他のシステム装置は、限定を伴うことなしに、例えば、S D R A M (S y n c h r o n o u s D y n a m i c R a n d o m A c c e s s M e m o r y) 制御部 1 4 6、S R A M (S t a t i c R a n d o m A c c e s s M e m o r y) 制御部 1 4 7、バーストフラッシュメモリインターフェイス 1 4 8、P C M C I A (P e r s o n a l C o m p u t e r M e m o r y C a r d I n t e r n a t i o n a l A s s o c i a t i o n) / C F (C o m p a c t F l a s h) カード制御部 1 4 9、液晶ディスプレイ (L i q u i d C r y s t a l D i s p l a y : L C D) 制御部 1 5 0、ダイレクトメモリアクセス (D i r e c t M e m o r y A c c e s s : D M

50

A) コントローラ 151、及びオルタネートバスマスタインターフェイス 152 を含んでもよい。又、一実施形態においては、データ処理システム 140 は、I/Oバス 153 を介して様々な I/O 装置と通信するための I/Oブリッジ 154 を有してもよい。このような I/O 装置は、限定を伴うことなしに、例えば、UART (Universal Asynchronous Receiver/Transmitter) 155、USB (Universal Serial Bus) 156、Bluetooth (登録商標) 無線 UART 157、及び I/O 拡張インターフェイス 158 を含んでもよい。

【0038】

データ処理システム 140 の一実施形態は、モバイル、ネットワーク、及び/又は無線通信と、テキストストリング比較演算を含む SIMD 演算を実行する能力を有する処理コア 159 と、を提供している。処理コア 159 は、Walsh-Hadamard 変換、高速フーリエ変換 (Fast Fourier Transformation: FFT)、離散コサイン変換 (Discrete Cosine Transform: DCT) などの離散変換、及びこれらの個々の逆変換、色空間変換、ビデオ符号化モーション推定、又はビデオ復号化モーション補償などの圧縮/圧縮解除法、及び PCM (Pulse Coded Modulation) などの変調/復調 (MODEM) 機能を含む様々なオーディオ、ビデオ、撮像、及び通信アルゴリズムを有するようにプログラムしてもよい。

10

【0039】

図 1C は、複数の試験ソースに対する OR 試験及び AND 試験機能を提供するための命令を実行する能力を有するデータ処理システムの別の代替実施形態を示している。一代替実施形態によれば、データ処理システム 160 は、主プロセッサ 166 と、SIMD コプロセッサ 161 と、キャッシュメモリ 167 と、入出力システム 168 と、を含んでもよい。入出力システム 168 は、任意選択により、無線インターフェイス 169 に結合してもよい。SIMD コプロセッサ 161 は、一実施形態による命令を含む演算を実行する能力を有する。処理コア 170 は、1つ又は複数のプロセス技術による製造に適することになり、且つ、機械可読媒体上において十分詳細に表現されることにより、処理コア 170 を含むデータ処理システム 160 のすべて又は一部の製造を円滑に実行するのに適することとなる。

20

【0040】

一実施形態においては、SIMD コプロセッサ 161 は、実行ユニット 162 と、1つ又は複数のレジスタファイルの組 164 と、有する。主プロセッサ 166 の一実施形態は、実行ユニット 162 による実行のために一実施形態による命令を含む命令セット 163 の命令を認識するためのデコーダ 165 を有する。又、代替実施形態においては、SIMD コプロセッサ 161 は、命令セット 163 の命令を復号化するためにデコーダ 165B の少なくとも一部分を有する。又、処理コア 170 は、本発明の実施形態の理解のためには不要な更なる回路 (図示されてはいない) をも含む。

30

【0041】

動作の際には、主プロセッサ 166 は、キャッシュメモリ 167 及び入出力システム 168 とのやり取りを含む一般的なタイプのデータ処理演算を制御するデータ処理命令のストリームを実行する。データ処理命令のストリームには、SIMD コプロセッサ命令が組み込まれている。主プロセッサ 166 のデコーダ 165 は、これらの SIMD コプロセッサ命令を、装着されている SIMD コプロセッサ 161 によって実行すべきタイプであるものとして認識する。相応して、主プロセッサ 166 は、これらの SIMD コプロセッサ命令 (又は、SIMD コプロセッサ命令を表す制御信号) をコプロセッサバス 171 上において発行し、任意の装着されている SIMD コプロセッサが、このバスから、これらの命令を受け取る。このケースにおいては、SIMD コプロセッサ 161 は、受け取った自身を宛先とする任意の SIMD コプロセッサ命令を受け付けると共に実行することになる。

40

【0042】

50

SIMDコプロセッサ命令による処理のために、データを無線インターフェイス169を介して受け取ってもよい。一例として、音声通信をデジタル信号の形態で受け取ってもよく、このデジタル信号をSIMDコプロセッサ命令によって処理し、音声通信を表すデジタルオーディオサンプルを再生してもよい。別の例として、圧縮されたオーディオ及び/又はビデオをデジタルビットストリームの形態で受け取ってもよく、このデジタルビットストリームをSIMDコプロセッサ命令によって処理し、デジタルオーディオサンプル及び/又はモーションビデオフレームを再生してもよい。処理コア170の一実施形態においては、主プロセッサ166及びSIMDコプロセッサ161を単一の処理コア170として統合しており、この処理コア170は、実行ユニット162と、1つ又は複数のレジスタファイルの組164と、一実施形態による命令を含む命令セット163の命令を認識するためのデコーダ165と、を有する。

10

【0043】

図2は、本発明の一実施形態による命令を実行するための論理回路を含むプロセッサ200のマイクロアーキテクチャのブロック図である。いくつかの実施形態においては、一実施形態による命令は、バイト、ワード、ダブルワード、クアドワード(quadword)などのサイズと、単精度及び倍精度の整数及び浮動小数点データタイプなどのデータタイプと、を有するデータ要素に対して動作するように実装することができる。一実施形態においては、インオーダーフロントエンド201は、プロセッサ200の一部であり、実行対象の命令をフェッチし、且つ、後からプロセッサパイプライン内において使用されるようにそれらの命令の準備を行う。フロントエンド201は、いくつかのユニットを含んでもよい。一実施形態においては、命令プリフェッチャ226が、メモリから命令をフェッチし、且つ、それらの命令を命令デコーダ228に供給し、次いで、命令デコーダ228が、それらの命令を復号化及び解釈する。例えば、一実施形態においては、デコーダは、受け取った命令を、機械が実行することができる「マイクロ命令」又は「マイクロ演算」と呼ばれる(micro-op又は μop とも呼ばれる)1つ又は複数の演算として復号化する。その他の実施形態においては、デコーダは、命令を、一実施形態に従って演算を実行するためにマイクロアーキテクチャによって使用されるopcode及び対応するデータ及び制御フィールドとして解析する。一実施形態においては、トレースキャッシュ230が、復号化された μop を取得し、且つ、実行のために、それらの μop を μop キュー234内においてプログラム順序付けシーケンス又はトレースとして取り纏める。トレースキャッシュ230が複雑な命令に遭遇した際には、マイクロコードROM232が、演算を完成させるのに必要な μop を提供する。

20

30

【0044】

いくつかの命令は、単一のmicro-opに変換され、その他のものは、演算の全体を完了させるために、いくつかのmicro-opを必要としている。一実施形態においては、命令を完成させるために4つを上回る数のmicro-opが必要である場合には、デコーダ228は、マイクロコードROM232にアクセスし、命令を完成させる。一実施形態においては、命令は、命令デコーダ228における処理のために、少数のmicro-opとして復号化することができる。別の実施形態においては、演算を実現するためにいくつかのmicro-opが必要とされる場合には、命令をマイクロコードROM232内に保存することができる。トレースキャッシュ230は、一実施形態による1つ又は複数の命令を完成させるためにマイクロコードROM232からマイクロコードシーケンスを読み取るべく正しいマイクロ命令ポインタを判定するためのエントリポイントPLA(Programmable Logic array)を表している。マイクロコードROM232が1つの命令のmicro-opの順序付けを終了した後に、機械のフロントエンド201は、トレースキャッシュ230からのmicro-opのフェッチを再開する。

40

【0045】

アウトオブオーダー実行エンジン203は、実行のために命令の準備が行われるところである。アウトオブオーダー実行論理は、命令がパイプラインを下ると共に実行のために

50

スケジューリングされるのに伴って、性能の最適化のために命令の流れを滑らかにすると共に再順序付けするために、いくつかのバッファを有する。アロケータ論理は、実行のためにそれぞれの μop が必要としている機械バッファ及びリソースを割り当てる。レジスタリネーム論理は、論理レジスタをレジスタファイル内のエン트리上にリネームする。又、アロケータは、メモリスケジューラ、高速スケジューラ 202、低速/一般的浮動小数点スケジューラ 204、及び単純浮動小数点スケジューラ 206 からなる命令スケジューラの前にあって、メモリ演算用のキューと非メモリ演算用のキューからなる 2 つの μop キューのうち 1 つのキュー内のそれぞれの μop ごとにエントリを割り当てる。 μop スケジューラ 202、204、206 は、 μop の実行の準備が整うタイミングを、その従属する入力レジスタオペランドソースの準備完了状態と、 μop がその演算を完了させるのに必要とする実行リソースの可用性と、に基づいて、判定する。一実施形態の高速スケジューラ 202 は、主クロックサイクルのそれぞれの半分ごとに、スケジューリングすることが可能であり、その他のスケジューラは、主プロセッサクロックサイクルごとに、スケジューリングするのみである。スケジューラは、 μop の実行をスケジューリングするためにディスパッチポートの間の調整を行う。

【0046】

レジスタファイル 208、210 は、スケジューラ 202、204、206 と実行ブロック 211 内の実行ユニット 212、214、216、218、220、222、224 の間に位置している。それぞれ、整数演算と浮動小数点演算のために別個のレジスタファイル 208、210 が存在している。又、一実施形態のそれぞれのレジスタファイル 208、210 は、パイパスネットワークをも含んでおり、パイパスネットワークは、レジスタファイル内にまだ書き込まれていない完了したばかりの結果をパイパスするか又は新しい従属 μop に対して転送することができる。又、整数レジスタファイル 208 及び浮動小数点レジスタファイル 210 は、他方との間においてデータを通信する能力をも有する。一実施形態においては、整数レジスタファイル 208 は、データの下位の 32 ビット用の 1 つのレジスタファイルと、データの上位の 32 ビット用の第 2 レジスタファイルと、からなる 2 つの別個のレジスタファイルに分割される。浮動小数点命令は、通常、幅が 64 ~ 128 ビットのオペランドを有しているため、一実施形態の浮動小数点レジスタファイル 210 は、128 ビット幅のエントリを有する。

【0047】

実行ブロック 211 は、実行ユニット 212、214、216、218、220、222、224 を収容しており、命令は、これらの実行ユニットにおいて実際に実行される。このセクションは、マイクロ命令が実行を要する整数及び浮動小数点データオペランド値を保存するレジスタファイル 208、210 を含む。一実施形態のプロセッサ 200 は、アドレス生成ユニット (Address Generation Unit: AGU) 212、AGU 214、高速 ALU 216、高速 ALU 218、低速 ALU 220、浮動小数点 ALU 222、及び浮動小数点移動ユニット 224 からなるいくつかの実行ユニットから構成されている。一実施形態においては、浮動小数点実行ブロック 222、224 は、浮動小数点、MMX、SIMD、及び SSE、或いは、その他の動作を実行する。一実施形態の浮動小数点 ALU 222 は、除算、平方根、及び残りの $micro-op$ を実行するための 64 ビット \times 64 ビットの浮動小数点除算器を含む。本発明の実施形態においては、浮動小数点値を伴う命令は、浮動小数点ハードウェアによって処理してもよい。一実施形態においては、ALU 演算は、高速 ALU 実行ユニット 216、218 に進む。一実施形態の高速 ALU 216、218 は、クロックサイクルの半分の有効レイテンシーを伴って高速演算を実行することができる。一実施形態においては、低速 ALU 220 が、乗算器、シフト、フラグ論理、及び分岐処理などの長レイテンシータイプの演算用の整数実行ハードウェアを含んでいることから、大部分の複雑な整数演算は、低速 ALU 220 に進む。メモリロード/ストア演算は、AGU 212、214 によって実行される。一実施形態においては、整数 ALU 216、218、220 は、64 ビットのデータオペランドに対する整数演算を実行するという文脈において記述されている。代替実施形態におい

10

20

30

40

50

ては、ALU216、218、220は、16、32、128、256などを含む様々なデータビットをサポートするように実装することができる。同様に、浮動小数点ユニット222、224は、様々な幅のビットを有する一連のオペランドをサポートするように実装することができる。一実施形態においては、浮動小数点ユニット222、224は、SIMD及びマルチメディア命令との関連においては、128ビット幅のパック型データオペランドに対して実行することができる。

【0048】

一実施形態においては、popスケジューラ202、204、206は、親の読み込みの実行が終了する前に、従属演算をディスパッチする。又、popは、プロセッサ200内において推測によってスケジューリング及び実行されることから、プロセッサ200は、メモリ誤りを処理するための論理をも含む。データキャッシュ内におけるデータの読み込みに誤りが発生した場合には、一時的に誤ったデータをスケジューラが有する状態でパイプラインにおいて稼働中である従属演算が存在する可能性がある。再生メカニズムは、誤ったデータを使用している命令を追跡し、且つ、それらの命令を再実行する。再生を要するのは、従属演算のみであり、且つ、独立演算は、完了することが許容される。又、プロセッサの一実施形態のスケジューラ及び再生メカニズムは、複数の試験ソースに対するOR試験及びAND試験機能を提供する命令を捕捉するように設計されている。

【0049】

「レジスタ」という用語は、オペランドを識別するための命令の一部として使用されるオンボードプロセッサのストレージ場所を意味してもよい。換言すれば、レジスタは、(プログラムの観点において)プロセッサの外部から使用可能なものであってもよい。但し、一実施形態のレジスタは、意味において、特定のタイプの回路に限定されるべきではない。むしろ、一実施形態のレジスタは、データを保存及び提供する能力を有し、且つ、本明細書に記述されている機能を実行する能力を有する。本明細書に記述されているレジスタは、専用の物理レジスタ、レジスタのリネームを使用して動的に割り当てられる物理レジスタ、専用の且つ動的に割り当てられる物理レジスタの組合せなどの任意の数の異なる技法を使用するプロセッサ内の回路によって実装することができる。一実施形態においては、整数レジスタは、32ビットの整数データを保存している。又、一実施形態のレジスタファイルは、パック型データ用の8つのマルチメディアSIMDレジスタを収容している。以下の説明においては、レジスタは、カリフォルニア州Santa Claraに所在するインテル社のMMX技術に対応したマイクロプロセッサ内の64ビット幅のMMX(商標)レジスタ(いくつかの例においては、「mm」レジスタとも呼ばれる)などのパック型データを保持するように設計されたデータレジスタであるものとして理解されたい。整数の形態と浮動小数点の形態の両方において入手可能なこれらのMMXレジスタは、SIMD及びSSE命令を伴うパック型データ要素と共に動作することができる。同様に、SSE2、SSE3、SSE4、又はこれを上回る次数の(一般に、「SSE x 」と呼ばれる)技術に関係する128ビット幅のXMMレジスタを使用することにより、このようなパック型データオペランドを保持することもできる。一実施形態においては、パック型データ及び整数データの保存の際に、レジスタは、2つのデータタイプを弁別する必要がない。一実施形態においては、整数及び浮動小数点は、同一のレジスタファイル内に、又は異なるレジスタファイル内に、収容される。更には、一実施形態においては、浮動小数点及び整数データは、異なるレジスタ内に、又は同一のレジスタ内に、保存してもよい。

【0050】

以下の図面の例には、いくつかのデータオペランドが示されている。図3Aは、本発明の一実施形態によるマルチメディアレジスタ内における様々なパック型データタイプの表現を示している。図3Aは、128ビット幅のオペランドにおけるパック型バイト310、パック型ワード320、及びパック型ダブルワード(dword)330のデータタイプを示している。この例のパック型バイトフォーマット310は、長さが128ビットであり、且つ、16個のパック型バイト要素を収容している。1バイトは、ここでは、デー

10

20

30

40

50

タの8ビットとして定義されている。それぞれのバイトデータ要素の情報は、バイト0の場合には、ビット7～ビット0内に、バイト1の場合には、ビット15～ビット8内に、バイト2の場合には、ビット23～ビット16内に、且つ、最後に、バイト15の場合には、ビット120～ビット127内に、保存される。従って、すべての利用可能なビットがレジスタ内において使用されている。このストレージ構成によれば、プロセッサのストレージ効率が向上する。同様に、16個のデータ要素に対してアクセスが実行されることにより、この場合には、1つの演算を並行して16個のデータ要素に対して実行することができる。

【0051】

一般に、データ要素とは、同一の長さのその他のデータ要素と共に、単一のレジスタ又はメモリ場所内に保存される個々のデータ片である。SSE技術に関するパック型データシーケンスにおいては、XMMレジスタ内に保存されているデータ要素の数は、128ビットを個々のデータ要素のビット長によって除算したものである。同様に、MMX及びSSE技術に関するパック型データシーケンスにおいては、MMXレジスタ内に保存されているデータ要素の数は、64ビットを個々のデータ要素のビット長によって除算したものである。図3Aに示されているデータタイプは、128ビット長であるが、本発明の実施形態は、64ビット幅、256ビット幅、512ビット幅、又はその他のサイズのオペランドと共に動作することもできる。この例のパック型ワードフォーマット320は、128ビット長であり、且つ、8つのパック型ワードデータ要素を収容している。それぞれのパック型ワードは、16ビットの情報を収容している。図3Aのパック型ダブルワードフォーマット330は、128ビット長であり、且つ、4つのパック型ダブルワードデータ要素を収容している。それぞれのパック型ダブルワードデータ要素は、32ビットの情報を収容している。パック型クアドワードは、長さが128ビットであり、且つ、2つのパック型クアドワードデータ要素を収容している。

【0052】

図3Bは、代替インレジスタデータ保存フォーマットを示している。それぞれのパック型データは、複数の独立したデータ要素を含むことができる。パック型ハーフ341、パック型シングル342、及びパック型ダブル343という3つのパック型データフォーマットが図示されている。パック型ハーフ341、パック型シングル342、及びパック型ダブル343の一実施形態は、固定小数点データ要素を収容している。代替実施形態においては、パック型ハーフ341、パック型シングル342、及びパック型ダブル343のうちの1つ又は複数は、浮動小数点データ要素を収容してもよい。パック型ハーフ341の代替実施形態は、128ビット長であり、8つの16ビットのデータ要素を収容している。パック型シングル342の一実施形態は、128ビット長であり、且つ、4つの32ビットのデータ要素を収容している。パック型ダブル343の一実施形態は、128ビット長であり、且つ、2つの64ビットのデータ要素を収容している。このようなパック型データフォーマットは、例えば、96ビット、160ビット、192ビット、224ビット、256ビット、512ビット、又はこれを上回るものなどのその他のレジスタ長に更に拡張してもよいことを理解されたい。

【0053】

図3Cは、本発明の一実施形態によるマルチメディアレジスタ内における様々な符号付き及び符号なしパック型データタイプの表現を示している。符号なしパック型バイト表現344は、SIMDレジスタ内の符号なしパック型バイトの保存状態を示している。それぞれのバイトデータ要素の情報は、例えば、バイト0の場合には、ビット7～ビット0内に、バイト1の場合には、ビット15～ビット8内に、バイト2の場合には、ビット23～ビット16内に、且つ、最後に、バイト15の場合には、ビット120～ビット127内に、保存される。従って、すべての利用可能なビットがレジスタ内において使用されている。このストレージ構成によれば、プロセッサの保存効率を向上させることができる。同様に、16個のデータ要素に対してアクセスが実行されることにより、この場合には、1つの演算を並行して16個のデータ要素に対して実行することができる。符号付きパッ

10

20

30

40

50

ク型バイト表現 3 4 5 は、符号付きパック型バイトの保存状態を示している。すべてのバイトデータ要素の第 8 ビットが符号インジケータであることに留意されたい。符号なしパック型ワード表現 3 4 6 は、ワード 7 ~ ワード 0 が SIMD レジスタ内において保存される方式を示している。符号付きパック型ワード表現 3 4 7 は、符号なしパック型ワードインレジスタ表現 3 4 6 に類似している。それぞれのワードデータ要素の第 1 6 ビットが符号インジケータであることに留意されたい。符号なしパック型ダブルワード表現 3 4 8 は、ダブルワードデータ要素が保存される方式を示している。符号付きパック型ダブルワード表現 3 4 9 は、符号なしパック型ダブルワードインレジスタ表現 3 4 8 に類似している。必要な符号ビットは、それぞれのダブルワードデータ要素の第 3 2 ビットであることに留意されたい。

10

【 0 0 5 4 】

図 3 D は、3 2 個以上のビットと、intel.com/products/processor/manuals/においてワールドワイドウェブ(www)上においてカリフォルニア州 Santa Clara に所在するインテル社から入手可能な「Intel (R) 6 4 and IA-3 2 Intel Architecture Software Developer's Manual Combined Volumes 2 A and 2 B: Instruction Set Reference A-Z」に記述されているタイプの opcode と対応するレジスタ/メモリオペランドアドレス指定モードと、を有する演算符号化(opcode)フォーマット 3 6 0 の一実施形態の図である。一実施形態においては、命令は、フィールド 3 6 1 及び 3 6 2 のうちの 1 つ又は複数によって符号化してもよい。1 命令当たり最大で 2 つのオペランド場所を識別してもよく、これには、最大で 2 つのソースオペランド識別子 3 6 4 及び 3 6 5 が含まれる。一実施形態においては、デスティネーションオペランド識別子 3 6 6 は、ソースオペランド識別子 3 6 4 と同一であり、その他の実施形態においては、これらは異なる。一代替実施形態においては、デスティネーションオペランド識別子 3 6 6 は、ソースオペランド識別子 3 6 5 と同一であり、その他の実施形態においては、これらは異なる。一実施形態においては、ソースオペランド識別子 3 6 4 及び 3 6 5 によって識別されるソースオペランドのうちの 1 つは、命令の結果によって上書きされ、その他の実施形態においては、識別子 3 6 4 は、ソースレジスタ要素に対応しており、且つ、識別子 3 6 5 は、デスティネーションレジスタ要素に対応している。一実施形態においては、オペランド識別子 3 6 4 及び 3 6 5 を使用し、3 2 ビット又は 6 4 ビットのソース及びデスティネーションオペランドを識別してもよい。

20

30

【 0 0 5 5 】

図 3 E は、4 0 個以上のビットを有する別の代替演算符号化(opcode)フォーマット 3 7 0 の図である。opcode フォーマット 3 7 0 は、opcode フォーマット 3 6 0 に対応しており、且つ、任意選択の接頭辞バイト 3 7 8 を有する。一実施形態による命令は、フィールド 3 7 8、3 7 1、及び 3 7 2 のうちの 1 つ又は複数によって符号化してもよい。ソースオペランド識別子 3 7 4 及び 3 7 5 により、且つ、接頭辞バイト 3 7 8 により、1 命令当たり最大で 2 つのオペランド場所を識別してもよい。一実施形態においては、接頭辞バイト 3 7 8 を使用し、3 2 ビット又は 6 4 ビットのソース及びデスティネーションオペランドを識別してもよい。一実施形態においては、デスティネーションオペランド識別子 3 7 6 は、ソースオペランド識別子 3 7 4 と同一であり、その他の実施形態においては、これらは異なる。一代替実施形態においては、デスティネーションオペランド識別子 3 7 6 は、ソースオペランド識別子 3 7 5 と同一であり、その他の実施形態においては、これらは異なる。一実施形態においては、命令は、オペランド識別子 3 7 4 及び 3 7 5 によって識別されたオペランドのうちの 1 つ又は複数に対して動作し、且つ、オペランド識別子 3 7 4 及び 3 7 5 によって識別された 1 つ又は複数のオペランドは、命令の結果によって上書きされ、その他の実施形態においては、識別子 3 7 4 及び 3 7 5 によって識別されたオペランドは、別のレジスタ内の別のデータ要素に書き込まれる。opcode フォーマット 3 6 0 及び 3 7 0 は、MOD フィールド 3 6 3 及び 3 7 3 により、

40

50

且つ、任意選択のSIB (Scale - Index - Base) 及び変位バイトにより、部分的に規定された、レジスタからレジスタの、メモリからレジスタの、メモリによるレジスタの、レジスタによるレジスタの、即値によるレジスタ (register by Immediate) の、レジスタからメモリのアドレス指定を許容する。

【0056】

次に図3Fを参照すれば、いくつかの代替実施形態においては、64ビット (又は128ビット、又は256ビット、又は512ビット以上) のSIMD (Single Instruction Multiple Data) 算術演算は、CDP (Coprocessor Data Processing) 命令を通じて実行してもよい。演算符号化 (opcode) フォーマット380は、CDPのopcodeフィールド382及び389を有するこのような1つのCDP命令を示している。代替実施形態においては、CDP命令のタイプ及び演算は、フィールド383、384、387、及び388のうちの1つ又は複数によって符号化してもよい。1命令当たり最大で3つのオペランド場所を識別してもよく、これには、最大で2つのソースオペランド識別子385及び390及び1つのデスティネーションオペランド識別子386が含まれる。コプロセッサの一実施形態は、8、16、32、及び64ビット値に対して動作することができる。一実施形態においては、命令は、整数データ要素に対して実行される。いくつかの実施形態においては、命令は、条件フィールド381を使用することにより、条件付きで実行してもよい。いくつかの実施形態においては、ソースデータサイズは、フィールド383によって符号化してもよい。いくつかの実施形態においては、ゼロ (Z)、負 (N)、桁上げ (C)、及びオーバーフロー (V) の検出をSIMDフィールドに対して実行することができる。いくつかの命令においては、飽和のタイプをフィールド384によって符号化してもよい。

【0057】

次に図3Gを参照すれば、これは、intel.com/products/processor/manuals/においてワールドワイドウェブ (www) 上においてカリフォルニア州Santa Claraに所在するインテル社から入手可能な「Intel (R) Advanced Vector Extensions Programming Reference」に記述されているopcodeフォーマットのタイプと対応する、別の実施形態による複数の試験ソースに対してOR試験及びAND試験機能を提供するための、別の代替演算符号化 (opcode) フォーマット397の図である。

【0058】

オリジナルのx86命令セットは、その存在が第1「opcode」バイトから判明している更なるバイト内に収容されたアドレスシラブル及び即値オペランドの様々なフォーマットを有する1バイトのopcodeについて提供されている。更には、opcodeに対する修飾子として予約された特定のバイト値が存在していた (命令の前に置かれなければならないことから、これらは、接頭辞と呼ばれる)。256個のopcodeバイトからなるオリジナルのパレット (これらの特殊な接頭辞値を含む) が尽きた際には、単一バイトを256個のopcodeの新しい組に対するエスケープとして振り向けている。ベクトル命令 (例えば、SIMD) の追加に伴って、更なるopcodeに対する必要性が生じ、且つ、接頭辞の使用を通じて拡張した際にも、「2バイト」のopcodeマップでも不十分であった。この目的のために、2バイトと任意選択の接頭辞を加えたものを識別子として使用する更なるマップにおいて新しい命令を追加している。

【0059】

更には、64ビットモードにおいて追加レジスタを促進するために、接頭辞とopcode (並びに、opcodeの判定に必要な任意のエスケープバイト) の間において、更なる接頭辞 (「REX」と呼ばれる) を使用してもよい。一実施形態においては、REXは、64ビットモードにおける追加のレジスタの使用を通知するための4個の「ペイロード」ビットを有してもよい。その他の実施形態においては、これは、4つを下回る又は上回る数のビットを有してもよい。(一般的には、フォーマット360及び/又はフォーマット370に対応している) 少なくとも1つの命令セットの一般的なフォーマットは、以

10

20

30

40

50

下のフォーマットによって一般的に示される。

【0060】

[prefixes][rex]escape[escape2]opcode modrm(etc.)

【0061】

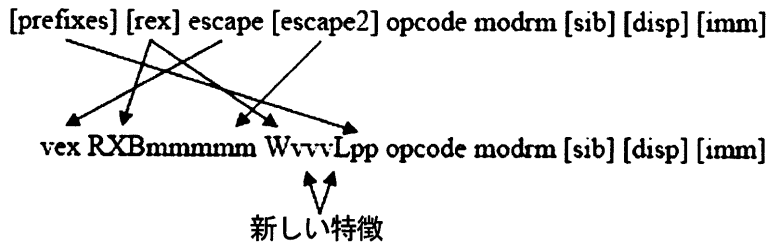
opcodeフォーマット397は、opcodeフォーマット370に対応しており、且つ、大部分のその他の一般に使用されるレガシー命令接頭辞バイト及びエスケープコードを置換するための任意選択のVEX接頭辞バイト391（一実施形態においては、16進のC4によって始まっている）を有する。例えば、以下のものは、2つのフィールドを使用して命令を符号化する実施形態を示しており、これは、第2エスケープコードがオリジナルの命令内に存在している際に、又はRFXフィールド内の余分なビット（例えば、XB及びWフィールド）を使用する必要がある際に、使用してもよい。以下に示されている実施形態においては、レガシーエスケープは、新しいエスケープ値によって表されており、レガシー接頭辞は、「ペイロード」バイトの一部として十分に圧縮されており、レガシー接頭辞は、リクレーンされると共に将来の拡張のために利用可能であり、第2エスケープコードは、利用可能な将来マップ及び特徴空間と共に、「マップ」フィールド内において圧縮され、且つ、新しい特徴（例えば、ベクトル長の増大及び追加のソースレジスタスペシファイア）が追加されている。

10

【0062】

【数1】

20



【0063】

一実施形態による命令は、フィールド391及び392のうちの1つ又は複数によって符号化してもよい。ソースオペランド識別子374及び375との組合せにおいて、且つ任意選択のSIB(Scale-Index-Base)識別子393、任意選択の変位識別子394、及び任意選択の即値バイト(immediate byte)395との組合せにおいて、フィールド391により、1命令当たり最大で4つのオペランド場所を識別してもよい。一実施形態においては、VEX接頭辞バイト391を使用し、32ビット又は64ビットのソース及びデスティネーションオペランド及び/又は128ビット又は256ビットのSIMDレジスタ又はメモリオペランドを識別してもよい。一実施形態においては、opcodeフォーマット397によって提供される機能は、opcodeフォーマット370と冗長な関係にあるものであってもよく、その他の実施形態においては、これらは異なっている。opcodeフォーマット370及び397は、MODフィールド373により、且つ、任意選択の(SIB)識別子383、任意選択の変位識別子394、及び任意選択の即値バイト395により、部分的に規定されたレジスタからレジスタの、メモリからレジスタの、メモリによるレジスタの、レジスタによるレジスタの、即値によるレジスタの、レジスタからメモリのアドレス指定を許容する。

30

40

【0064】

次に図3Hを参照すれば、これは、別の実施形態による複数の試験ソースに対するOR試験及びAND試験機能を提供するための別の代替演算符号化(opcode)フォーマット398の図である。opcodeフォーマット398は、opcodeフォーマット370及び397と対応しており、且つ、大部分のその他の一般的に使用されるレガシー命令接頭辞バイト及びエスケープコードを置換すると共に追加機能を提供するための（一実施形態においては、16進の62によって始まる）任意選択のEVE X接頭辞バイト3

50

96を有する。一実施形態による命令は、フィールド396及び392のうちの1つ又は複数によって符号化してもよい。ソースオペランド識別子374及び375との組合せにおいて、且つ、任意選択のSIB(Scale-Index-Base)識別子393、任意選択の変位識別子394、及び任意選択の即値バイト395との組合せにおいて、フィールド396により、1命令当たり最大で4つのオペランド場所と1つのマスクを識別してもよい。一実施形態においては、EVEX接頭辞バイト396を使用し、32ビット又は64ビットのソース及びデスティネーションオペランド及び/又は128ビット、256ビット、又は512ビットのSIMDレジスタ又はメモリオペランドを識別してもよい。一実施形態においては、opcodeフォーマット398によって提供される機能は、opcodeフォーマット370又は397と冗長な関係にあるものであってもよく、その他の実施形態においては、これらは異なっている。opcodeフォーマット398は、MODフィールド373により、且つ、任意選択の(SIB)識別子393、任意選択の変位識別子394、及び任意選択の即値バイト395により、部分的に規定された、マスクを伴う、レジスタからレジスタの、メモリからレジスタの、メモリによるレジスタの、レジスタによるレジスタの、即値によるレジスタの、レジスタからメモリのアドレス指定を許容する。(一般的には、フォーマット360及び/又はフォーマット370に対応している)少なくとも1つの命令セットの一般的なフォーマットは、以下のものによって一般的に示される。

【0065】

```
evex1  R X B m m m m m  W v v v L p p  evex4  opcode  mod 20
rm[ sib ][ disp ][ imm ]
```

【0066】

一実施形態においては、EVEXフォーマット398に従って符号化された命令は、追加の「ペイロード」ビットを有してもよく、この追加の「ペイロード」ビットは、例えば、ユーザー構成可能マスクレジスタ、又は追加のオペランド、或いは、128ビット、256ビット、又は512ビットのベクトルレジスタ、又は選択対象の更なるレジスタなどの中からの選択などの追加の新しい機能と共に、複数の試験ソースに対するOR試験及びAND試験機能を提供するために使用してもよい。

【0067】

例えば、VEXフォーマット397を使用して黙示的なマスクを伴う複数の試験ソースに対するOR試験及びAND試験機能を提供してもよい場合には、EVEXフォーマット398を使用して明示的なユーザー構成可能なマスクを伴う複数の試験ソースに対するOR試験及びAND試験機能を提供してもよい。更には、VEXフォーマット397を使用して128ビット又は256ビットのベクトルレジスタ上において複数の試験ソースに対するOR試験及びAND試験機能を提供してもよい場合には、EVEXフォーマット398を使用して128ビット、256ビット、512ビット、或いは、更に大きな(又は、更に小さな)ビット数のベクトルレジスタ上において複数の試験ソースに対するOR試験及びAND試験機能を提供してもよい。

【0068】

複数の試験ソースに対するOR試験及びAND試験機能を提供するための例示用のフェーシング可能な命令は、以下の例によって示される。

【0069】

10

20

30

40

【表 1】

命令	デスティネーション	ソース 1	ソース 2	ソース 3	説明
OR 試験	ゼロフラグ	Reg 1	Reg 2 / Mem	Imm 8 / Imm 3 2	Reg 1 及び Reg 2 / Mem オペランドからのデータの間においてビット論理 OR を実行し、且つ、論理 OR の結果と Imm 8 / Imm 3 2 オペランドの間においてビット論理 AND を実行し、論理 AND の結果に従ってゼロフラグを設定する。
AND 試験	ゼロフラグ	Reg 1	Reg 2 / Mem	Imm 8 / Imm 3 2	Reg 1 及び Reg 2 / Mem オペランドからのデータの間においてビット論理 AND を実行し、且つ、Imm 8 / Imm 3 2 オペランドと第 1 論理 AND の結果の間において別のビット論理 AND を実行し、第 2 論理 AND の結果に従ってゼロフラグを設定する。

10

20

【 0 0 7 0 】

上述の例と同様に、複数の試験ソースに対する OR 試験及び / 又は AND 試験機能を提供するためのフュージング可能な命令及び論理を使用し、動的であると共に弱いタイピング型である場合があり且つオブジェクト指向の命令型であると共に機能的なプログラムスタイルをサポートしてもよい JavaScript (登録商標)、JScript、又は ECMAScript などの 1 つ又は複数のインタープリタ型コンピュータプログラム言語をサポートする JIT コンパイラによって生成される頻繁に実行されるタイプ試験命令イディオムと関連する性能の制限の問題、エネルギー消費の懸念、及びその他の実行時のボトルネックをほとんど除去してもよいことを理解されたい。

30

【 0 0 7 1 】

図 4 A は、本発明の少なくとも 1 つの実施形態によるインオーダーパイプライン及びレジスタリネームステージアウトオブオーダー発行 / 実行パイプラインを示すブロック図である。図 4 B は、本発明の少なくとも 1 つの実施形態によるプロセッサ内に含まれるインオーダーアーキテクチャコア及びレジスタリネーム論理アウトオブオーダー発行 / 実行論理を示すブロック図である。図 4 A の実線のボックスは、インオーダーパイプラインを示しており、破線のボックスは、レジスタリネームアウトオブオーダー発行 / 実行パイプラインを示している。同様に、図 4 B の実線のボックスは、インオーダーアーキテクチャ論理を示しており、破線のボックスは、レジスタリネーム論理及びアウトオブオーダー発行 / 実行論理を示している。

40

【 0 0 7 2 】

図 4 A においては、プロセッサパイプライン 400 は、フェッチステージ 402 と、長さ復号化ステージ 404 と、復号化ステージ 406 と、割当ステージ 408 と、リネームステージ 410 と、スケジューリング (ディスパッチ又は発行とも呼ばれる) ステージ 412 と、レジスタ読取り / メモリ読取りステージ 414 と、実行ステージ 416 と、書戻し / メモリ書込みステージ 418 と、例外処理ステージ 422 と、コミットステージ 424 と、を含む。

【 0 0 7 3 】

図 4 B においては、矢印は、複数のユニットの間の結合を表記しており、且つ、矢印の

50

方向は、それらのユニットの間におけるデータの流れの方向を示している。図4Bは、実行エンジンユニット450に結合されたフロントエンドユニット430を含むプロセッサコア490を示しており、且つ、これらは、いずれも、メモリユニット470に結合されている。

【0074】

コア490は、RISC(Reduced Instruction Set Computing)コア、CISC(Complex Instruction Set Computing)コア、VLIW(Very Long Instruction Word)コア、或いは、ハイブリッド又は代替コアタイプであってもよい。更に別の選択肢として、コア490は、例えば、ネットワーク又は通信コア、圧縮エンジン、グラフィックスコア、又はこれらに類似したものなどの特殊目的コアであってもよい。

10

【0075】

フロントエンドユニット430は、命令キャッシュユニット434に結合された分岐予測ユニット432を含み、分岐予測ユニット432は、命令TLB(Translation Lookaside Buffer)436に結合され、命令TLB436は、命令フェッチユニット438に結合され、命令フェッチユニット438は、復号化ユニット440に結合されている。復号化ユニット又はデコーダは、命令を復号化してもよく、且つ、出力として、1つ又は複数のマイクロ演算、マイクロコードエントリポイント、マイクロ命令、その他の命令、又はその他の制御信号を生成し、これらは、オリジナルの命令から復号化されるか、或いは、さもなければ、オリジナルの命令を反映しているか、或いは、オリジナルの命令から導出される。デコーダは、様々な異なるメカニズムを使用して実装してもよい。適切なメカニズムの例は、限定を伴うことなしに、ルックアップテーブル、ハードウェア実装形態、PLA(Programmable Logic Array)、マイクロコードROM(Read Only Memory)などを含む。命令キャッシュユニット434は、メモリユニット470内のレベル2(L2)キャッシュユニット476に更に結合されている。復号化ユニット440は、実行エンジンユニット450内のリネーム/アロケータユニット452に結合されている。

20

【0076】

実行エンジンユニット450は、リネーム/アロケータユニット452を含み、リネーム/アロケータユニット452は、リタイアメントユニット454及び1つ又は複数のスケジューラユニットの組456に結合されている。1つ又は複数のスケジューラユニット456は、予約ステーションや中央命令ウィンドウなどを含む任意の数の異なるスケジューラを表している。1つ又は複数のスケジューラユニット456は、1つ又は複数の物理レジスタファイルユニット458に結合されている。1つ又は複数の物理レジスタファイルユニット458のそれぞれは、1つ又は複数の物理レジスタファイルを表しており、これらのうちの異なるものは、スカラー整数、スカラー浮動小数点、パック型整数、パック型浮動小数点、ベクトル整数、ベクトル浮動小数点などのような1つ又は複数の異なるデータタイプや状態(例えば、実行対象の次の命令のアドレスである命令ポインタ)などを保存する。1つ又は複数の物理レジスタファイルユニット458は、レジスタリネーム及びアウトオブオーダー実行を実装してもよい様々な方法(例えば、1つ又は複数のリオーダーバッファ及び1つ又は複数のリタイアメントレジスタファイルを使用するもの、1つ又は複数の将来ファイル、1つ又は複数の履歴ファイル、及び1つ又は複数のリタイアメントレジスタファイルを使用するもの、レジスタマップ及びレジスタのプールを使用するものなど)を示すために、リタイアメントユニット454によってオーバーラップされている。一般に、アーキテクチャレジスタは、プロセッサの外部から、又はプログラムの観点において、可視状態にある。レジスタは、任意の既知の特定のタイプの回路に限定されるものではない。本明細書に記述されているようにデータを保存及び提供する能力を有している限り、様々な異なるタイプのレジスタが適している。適切なレジスタの例は、限定を伴うことなしに、専用の物理レジスタ、レジスタリネームを使用する動的に割り当てられた物理レジスタ、専用の且つ動的に割り当てられた物理レジスタの組合せなどを含む。

30

40

50

【 0 0 7 7 】

リタイアメントユニット 4 5 4 及び 1 つ又は複数の物理レジスタファイルユニット 4 5 8 は、1 つ又は複数の実行クラスタ 4 6 0 に結合されている。1 つ又は複数の実行クラスタ 4 6 0 は、1 つ又は複数の実行ユニットの組 4 6 2 と、1 つ又は複数のメモリアクセスユニットの組 4 6 4 と、を含む。実行ユニット 4 6 2 は、様々な演算（例えば、シフト、加算、減算、乗算）を様々なタイプのデータ（例えば、スカラー浮動小数点、パック型整数、パック型浮動小数点、ベクトル整数、ベクトル浮動小数点）に対して実行してもよい。いくつかの実施形態は、特定の機能又は機能の組に対して専用であるいくつかの実行ユニットを含んでもよいが、その他の実施形態は、1 つの実行ユニットのみを含んでもよく、或いは、すべてがすべての機能を実行する複数の実行ユニットを含んでもよい。特定の
10
実施形態は、特定のタイプのデータ / 演算について別個のパイプライン（例えば、スカラー整数パイプライン、スカラー浮動小数点 / パック型整数 / パック型浮動小数点 / ベクトル整数 / ベクトル浮動小数点パイプライン、及び / 又はその独自のスケジューラユニット、1 つ又は複数の物理レジスタファイルユニット、及び / 又は実行クラスタをそれぞれが有するメモリアクセスパイプライン）を生成するために（別個のメモリアクセスパイプラインの場合には、そのパイプラインの実行クラスタのみが 1 つ又は複数のメモリアクセス
20
ユニット 4 6 4 を有する特定の実施形態が実装される）、1 つ又は複数のスケジューラユニット 4 5 6、1 つ又は複数の物理レジスタファイル 4 5 8、及び 1 つ又は複数の実行クラスタ 4 6 0 が複数となる可能性が高いものとして示されている。又、別個のパイプラインが使用される場合には、それらのパイプラインのうちの 1 つ又は複数は、アウトオブオー
20
ダー発行 / 実行であってもよく、且つ、残りは、インオーダーであってもよいことを理解されたい。

【 0 0 7 8 】

メモリアクセスユニットの組 4 6 4 は、メモリユニット 4 7 0 に結合されており、メモリユニット 4 7 0 は、データ TLB ユニット 4 7 2 を含み、データ TLB ユニット 4 7 2 は、データキャッシュユニット 4 7 4 に結合されており、データキャッシュユニット 4 7 4 は、レベル 2 (L 2) キャッシュユニット 4 7 6 に結合されている。例示用の一実施形態においては、メモリアクセスユニット 4 6 4 は、ロードユニット、ストアアドレスユニット、及びストアデータユニットを含んでもよく、これらのそれぞれは、メモリユニット
30
4 7 0 内のデータ TLB ユニット 4 7 2 に結合されている。L 2 キャッシュユニット 4 7 6 は、1 つ又は複数のその他のレベルのキャッシュに、且つ、最終的には、主メモリに、結合されている。

【 0 0 7 9 】

一例として、例示用のレジスタリネームアウトオブオーダー発行 / 実行コアアーキテクチャは、パイプライン 4 0 0 を以下のように実装してもよい。即ち、1) 命令フェッチ 4 3 8 がフェッチ及び長さ復号化ステージ 4 0 2 及び 4 0 4 を実行し、2) 復号化ユニット 4 4 0 が復号化ステージ 4 0 6 を実行し、3) リネーム / アロケータユニット 4 5 2 が割当ステージ 4 0 8 及びリネームステージ 4 1 0 を実行し、4) 1 つ又は複数のスケジューラユニット 4 5 6 がスケジュールステージ 4 1 2 を実行し、5) 1 つ又は複数の物理レジスタファイルユニット 4 5 8 及びメモリユニット 4 7 0 がレジスタ読取り / メモリ読取り
40
ステージ 4 1 4 を実行し、実行クラスタ 4 6 0 が実行ステージ 4 1 6 を実行し、6) メモリユニット 4 7 0 及び 1 つ又は複数の物理レジスタファイルユニット 4 5 8 が書戻し / メモリ書込みステージ 4 1 8 を実行し、7) 様々なユニットが例外処理ステージ 4 2 2 に関与してもよく、且つ、8) リタイアメントユニット 4 5 4 及び 1 つ又は複数の物理レジスタファイルユニット 4 5 8 がコミットステージ 4 2 4 を実行する。

【 0 0 8 0 】

コア 4 9 0 は、1 つ又は複数の命令セット（例えば、（相対的に新しいバージョンに伴って追加されるいくつかの拡張を有する）x 8 6 命令セット、カリフォルニア州 Sunnyvale に所在する MIPS Technologies 社の MIPS 命令セット、カリフォルニア州 Sunnyvale に所在する ARM ホールディングス社の（NEON な
50

どの任意選択の更なる拡張を伴う)ARM命令セット)をサポートしてもよい。

【0081】

コアは、(演算の複数の並列セット又はスレッドを実行する)マルチスレッディングをサポートしてもよく、且つ、タイムスライス型マルチスレッディング、(物理コアが同時にマルチスレッディングしているスレッドのそれぞれごとに、単一の物理コアが論理コアを提供する)同時マルチスレッディング、或いは、これらの組合せ(例えば、Intel(登録商標)Hyperthreading技術のように、タイムスライス型のフェッチ及び復号化、並びに、その後の同時マルチスレッディング)を含む様々な方法によってこれを実行してもよいことを理解されたい。

【0082】

レジスタのリネームは、アウトオブオーダー実行の文脈において記述されているが、レジスタリネームは、インオーダーアーキテクチャにおいて使用してもよいことを理解されたい。又、図示のプロセッサの実施形態は、別個の命令及びデータキャッシュユニット434/474及び共有L2キャッシュユニット476をも含んでいるが、代替実施形態は、例えば、レベル1(L1)内部キャッシュ又はマルチレベルの内部キャッシュなどの命令とデータの両方のための単一の内部キャッシュを有してもよい。いくつかの実施形態においては、システムは、内部キャッシュ並びにコア及び/又はプロセッサの外部に位置した外部キャッシュの組合せを含んでもよい。或いは、この代わりに、キャッシュのすべてがコア及び/又はプロセッサの外部に位置してもよい。

【0083】

図5は、本発明の実施形態による統合型のメモリコントローラ及びグラフィックスを有するシングルコアプロセッサ及びマルチコアプロセッサ500のブロック図である。図5の実線のボックスは、シングルコア502A、システムエージェント510、1つ又は複数のバスコントローラユニットの組516を有するプロセッサ500を示しており、破線のボックスの任意選択の追加は、複数のコア502A~N、システムエージェントユニット510内の1つ又は複数の統合型メモリコントローラユニットの組514、及び統合型のグラフィックス論理508を有する代替プロセッサ500を示している。

【0084】

メモリ階層は、コア内の1つ又は複数のレベルのキャッシュと、1つ又は複数の共有キャッシュユニットの組506と、統合型のメモリコントローラユニットの組514に結合された外部メモリ(図示されてはいない)と、を含む。共有キャッシュユニットの組506は、レベル2(L2)、レベル3(L3)、レベル4(L4)、又はその他のレベルのキャッシュなどの1つ又は複数の中間レベルキャッシュ、最終レベルキャッシュ(Last Level Cache: LLC)、及び/又はこれらの組合せを含んでもよい。一実施形態においては、リングに基づいた相互接続ユニット512が統合型グラフィックス論理508、共有キャッシュユニットの組506、及びシステムエージェントユニット510を相互接続しているが、代替実施形態は、このようなユニットを相互接続するための任意の数の周知の技法を使用してもよい。

【0085】

いくつかの実施形態においては、コア502A~Nのうちの1つ又は複数は、マルチスレッディングの能力を有する。システムエージェント510は、コア502A~Nを調整すると共に動作させるコンポーネントを含む。システムエージェントユニット510は、例えば、電力制御ユニット(Power Control Unit: PCU)と、ディスプレイユニットと、を含んでもよい。PCUは、コア502A~N及び統合型グラフィックス論理508の電力状態を調節するのに必要な論理及びコンポーネントであってもよく、或いは、これらを含んでもよい。ディスプレイユニットは、1つ又は複数の外部接続されたディスプレイを駆動するためのものである。

【0086】

コア502A~Nは、アーキテクチャ及び/又は命令セットの観点において同種のものであっても又は異種のものであってもよい。例えば、コア502A~Nのいくつかは、イ

10

20

30

40

50

ンオーダーであってもよく、その他のものは、アウトオブオーダーであってもよい。別の例として、コア502A~Nのうちの複数のものは、同一の命令セットを実行する能力を有してもよく、その他のものは、その命令セット又は異なる命令セットのサブセットのみを実行する能力を有してもよい。

【0087】

プロセッサは、カリフォルニア州Santa Claraに所在するインテル社から入手可能なCore(商標) i3、i5、i7、2 Duo、及びQuad、Xeon(商標)、Itanium(商標)、XScale(商標)、又はStrongARM(商標)プロセッサなどの汎用プロセッサであってもよい。或いは、この代わりに、プロセッサは、ARMホールディングス社やMIPS社などのような別の会社からのものであってもよい。プロセッサは、例えば、ネットワーク又は通信プロセッサ、圧縮エンジン、グラフィックスプロセッサ、コプロセッサ、埋込み型プロセッサ、又はこれらに類似したものなどの特殊目的プロセッサであってもよい。プロセッサは、1つ又は複数のチップ上において実装してもよい。プロセッサ500は、1つ又は複数の基板の一部であってもよいと共に/又は、例えば、BiCMOS、CMOS、又はNMOSなどのいくつかのプロセス技術のうちのいずれかを使用することにより、これらの基板上に実装してもよい。

10

【0088】

図6~図8は、プロセッサ500を含むのに適した例示用のシステムであり、図9は、コア502のうちの1つ又は複数を含んでもよい例示用のシステムオンチップ(System on a Chip: SoC)である。ラップトップ、デスクトップ、ハンドヘルド型PC、パーソナルデジタルアシスタント、エンジニアリングワークステーション、サーバー、ネットワーク装置、ネットワークハブ、スイッチ、埋込み型プロセッサ、デジタル信号プロセッサ(Digital Signal Processor: DSP)、グラフィックス装置、ビデオゲーム装置、セットトップボックス、マイクロコントローラ、セル電話機、携帯型メディアプレーヤー、ハンドヘルド型装置、及び様々なその他の電子装置用の当技術分野において既知のその他のシステム設計及び構成も適している。一般論として、本明細書に開示されているプロセッサ及び/又はその他の実行論理を内蔵する能力を有する様々なシステム又は電子装置は、一般に適している。

20

【0089】

次に図6を参照すれば、本発明の一実施形態によるシステム600のブロック図が示されている。システム600は、グラフィックスメモリコントローラハブ(Graphics Memory Controller Hub: GMCH)620に結合された1つ又は複数のプロセッサ610、615を含んでもよい。図6には、追加のプロセッサ615の任意選択である特性が破線によって表記されている。

30

【0090】

それぞれのプロセッサ610、615は、なんらかのバージョンのプロセッサ500であってもよい。但し、統合型グラフィックス論理及び統合型メモリ制御ユニットがプロセッサ610、615内に存在することになる可能性は低いことに留意されたい。図6は、GMCH620がメモリ640に結合されてもよく、メモリ640は、例えば、ダイナミックランダムアクセスメモリ(DRAM)であってもよいことを示している。DRAMは、少なくとも一実施形態においては、不揮発性キャッシュと関連するものであってもよい。

40

【0091】

GMCH620は、チップセットであってもよく、又はチップセットの一部であってもよい。GMCH620は、1つ又は複数のプロセッサ610、615と通信してもよく、且つ、1つ又は複数のプロセッサ610、615とメモリ640の間のやり取りを制御してもよい。又、GMCH620は、1つ又は複数のプロセッサ610、615とシステム600のその他の要素の間における高速バスインターフェイスとして機能してもよい。少なくとも一実施形態においては、GMCH620は、フロントサイドバス(Front Side Bus: FSB)695などのマルチドロップバスを介して、1つ又は複数のプ

50

ロセッサ610、615と通信している。

【0092】

更には、GMCH620は、ディスプレイ645（フラットパネルディスプレイなど）に結合されている。GMCH620は、統合型グラフィックスアクセラレータを含んでもよい。GMCH620は、様々な周辺装置をシステム600に結合するために使用してもよい入出力（I/O）コントローラハブ（Input/output Controller Hub：ICH）650に更に結合されている。図6の実施形態には、例えば、外部グラフィックス装置660が示されており、外部グラフィックス装置660は、別の周辺装置670と共に、ICH650に結合された個別のグラフィックス装置であってもよい。

10

【0093】

或いは、この代わりに、追加の又は異なるプロセッサがシステム600内に存在してもよい。例えば、1つ又は複数の追加のプロセッサ615は、プロセッサ610と同一である1つ又は複数の追加のプロセッサ、プロセッサ610とは異種である又は非対称である1つ又は複数の追加のプロセッサ、アクセラレータ（例えば、グラフィックスアクセラレータ又はデジタル信号処理（DSP）ユニットなど）、フィールドプログラマブルゲートアレイ、又は任意のその他のプロセッサを含んでもよい。物理リソース610、615の間には、アーキテクチャ特性、マイクロアーキテクチャ特性、熱特性、電力消費特性、及びこれらに類似したものを含む所定範囲の価値測定基準の観点において、様々な差が存在する可能性がある。事実上、これらの差が、プロセッサ610、615の中におけるそれぞれの非対称性及び異種性を明示することになる。少なくとも1つの実施形態においては、様々なプロセッサ610、615は、同一のダイパッケージ内に存在してもよい。

20

【0094】

次に図7を参照すれば、本発明の一実施形態による第2システム700のブロック図が示されている。図7に示されているように、マルチプロセッサシステム700は、ポイントツーポイント相互接続システムであり、且つ、第1プロセッサ770と、ポイントツーポイント相互接続750を介して結合された第2プロセッサ780と、を含む。プロセッサ770及び780のそれぞれは、プロセッサ610、615のうちの1つ又は複数のものとして、なんらかのバージョンのプロセッサ500であってもよい。

【0095】

2つのプロセッサ770、780のみを有するように示されているが、本発明の範囲は、これに限定されるものではないことを理解されたい。その他の実施形態においては、1つ又は複数の更なるプロセッサが所与のプロセッサ内に存在してもよい。

30

【0096】

プロセッサ770及び780は、統合型のメモリコントローラユニット772及び782をそれぞれ含むものとして示されている。又、プロセッサ770は、そのバスコントローラユニットの一部として、ポイントツーポイント（P-P）インターフェイス776及び778をも含んでおり、同様に、第2プロセッサ780も、P-Pインターフェイス786及び788を含む。プロセッサ770、780は、P-Pインターフェイス回路778、788を使用することにより、ポイントツーポイント（P-P）インターフェイス750を介して情報を交換してもよい。図7に示されているように、IMC772及び782は、プロセッサを個々のメモリに、即ち、メモリ732及びメモリ734に、結合しており、メモリ732及びメモリ734は、個々のプロセッサにローカルに装着された主メモリの一部分であってもよい。

40

【0097】

プロセッサ770、780は、ポイントツーポイントインターフェイス回路776、794、786、798を使用することにより、それぞれ、個々のP-Pインターフェイス752、754を介して、チップセット790との間において情報を交換してもよい。又、チップセット790は、高性能グラフィックスインターフェイス739を介して高性能グラフィックス回路738との間において情報を交換してもよい。

50

【0098】

1つのプロセッサが低電力モードに配置された場合に、いずれかの又は両方のプロセッサのローカルキャッシュ情報を共有キャッシュ内に保存してもよいように、共有キャッシュ（図示されてはいない）をいずれかのプロセッサ内に、又は両方のプロセッサの外部に、但し、P-Pインターフェイスを介してプロセッサと接続された状態において、含んでもよい。

【0099】

チップセット790は、インターフェイス796を介して第1バス716に結合してもよい。一実施形態においては、第1バス716は、PCI (Peripheral Component Interconnect) バスであってもよく、或いは、PCI Express バス又は別の第3世代I/O相互接続バスなどのバスであってもよいが、本発明の範囲は、これに限定されるものではない。

【0100】

図7に示されているように、第1バス716を第2バス720に結合するバスブリッジ718と共に、様々なI/O装置714を第1バス716に結合してもよい。一実施形態においては、第2バス720は、LPC (Low Pin Count) バスであってもよい。一実施形態においては、例えば、キーボード及び/又はマウス722、通信装置727、及び、命令/コード及びデータ730を含んでもよいディスクドライブ又はその他のマストレージ装置などのストレージユニット728を含む様々な装置を第2バス720に結合してもよい。更には、オーディオI/O724を第2バス720に結合してもよい。その他のアーキテクチャも可能であることに留意されたい。例えば、図7のポイントツーポイントアーキテクチャの代わりに、システムは、マルチドロップバス又はその他の類似のアーキテクチャを実装してもよい。

【0101】

次に図8を参照すれば、本発明の一実施形態による第3システム800のブロック図が示されている。図7及び図8の同一の要素は、同一の参照符号を有しており、且つ、図8においては、図8のその他の側面を曖昧にすることを回避するべく、図7の特定の側面が省略されている。

【0102】

図8は、プロセッサ870、880が統合型のメモリ及び/又はI/O制御論理（「Control Logic: CL」）872及び882をそれぞれ含んでもよいことを示している。少なくとも1つの実施形態においては、CL872、882は、図5及び図7との関連において上述したものなどの統合型メモリコントローラユニットを含んでもよい。更には、CL872、882は、I/O制御論理を含んでもよい。図8は、メモリ832、834がCL872、882に結合されているのみならず、I/O装置814も、制御論理872、882に結合されることを示している。レガシーI/O装置815は、チップセット890に結合されている。

【0103】

次に図9を参照すれば、本発明の一実施形態によるSoC900のブロック図が示されている。図5の類似の要素は、同一の参照符号を有する。又、破線のボックスは、更に高度なSoC上における任意選択の機能である。図9においては、1つ又は複数の相互接続ユニット902が、1つ又は複数のコアの組502A~N及び1つ又は複数の共有キャッシュユニット506を含むアプリケーションプロセッサ910と、システムエージェントユニット510と、1つ又は複数のバスコントローラユニット516と、1つ又は複数の統合型メモリコントローラユニット514と、統合型グラフィックス論理508、スチール及び/又はビデオカメラ機能を提供する画像プロセッサ924、ハードウェアオーディオ高速化を提供するオーディオプロセッサ926、及びビデオ符号化/復号化の高速化を提供するビデオプロセッサ928を含んでもよい1つ又は複数のメディアプロセッサ920と、スタティックランダムアクセスメモリ (SRAM) ユニット930と、直接メモリアクセス (Direct Memory Access: DMA) ユニット932と、1

10

20

30

40

50

つ又は複数の外部ディスプレイに結合するためのディスプレイユニット940と、に結合されている。

【0104】

図10は、一実施形態に従って少なくとも1つの命令を実行してもよい中央処理ユニット(Central Processing Unit: CPU)及びグラフィックス処理ユニット(Graphics Processing Unit: GPU)を収容するプロセッサを示している。一実施形態においては、少なくとも1つの実施形態に従って演算を実行するための命令をCPUによって実行することができよう。別の実施形態においては、命令をGPUによって実行することができよう。更に別の実施形態においては、命令は、GPU及びCPUによって実行される演算の組合せを通じて実行してもよい。例えば、一実施形態においては、GPU上における実行のために、一実施形態による命令を受信及び復号化してもよい。但し、復号化された命令内の1つ又は複数の演算をCPUによって実行してもよく、且つ、結果を命令の最終的なリタイアメントのためにGPUに返してもよい。逆に、いくつかの実施形態においては、CPUが主プロセッサとして機能してもよく、且つ、GPUがコプロセッサとして機能してもよい。

10

【0105】

いくつかの実施形態においては、高度な並列スループットプロセッサの利益を享受する命令をGPUによって実行してもよく、高深度のパイプライン型アーキテクチャの利益を享受するプロセッサの性能の利益を享受する命令を、CPUによって実行してもよい。例えば、グラフィックス、科学アプリケーション、財務アプリケーション、及びその他の並列ワークロードは、GPUの性能の利益を享受すると共に相応して実行してもよく、オペレーティングシステムカーネルやアプリケーションコードなどの更に逐次的なアプリケーションは、CPUに適していよう。

20

【0106】

図10においては、プロセッサ1000は、CPU1005と、GPU1010と、画像プロセッサ1015と、ビデオプロセッサ1020と、USBコントローラ1025と、UARTコントローラ1030と、SPI/SDIOコントローラ1035と、ディスプレイ装置1040と、HDMI(登録商標)(High-Definition Multimedia Interface)コントローラ1045と、MIPIコントローラ1050と、フラッシュメモリコントローラ1055と、DDR(Dual Data Rate)コントローラ1060と、セキュリティエンジン1065と、I²S/I²C(Integrated Interchip Sound/Inter-Integrated Circuit)インターフェイス1070と、を含む。更なるCPU又はGPU及びその他の周辺インターフェイスコントローラを含むその他の論理及び回路を図10のプロセッサ内に含んでもよい。

30

【0107】

少なくとも1つの実施形態の1つ又は複数の側面は、機械によって読み取られた際に本明細書に記述されている技法を実行するための論理を機械に製造させる、プロセッサ内の様々な論理を表す、機械可読媒体上に保存されたデータ表現によって実装してもよい。このような「IPコア」と呼ばれる表現は、有体の機械可読媒体(「テープ」)上に保存してもよく、且つ、論理又はプロセッサを実際に製造する製造機械に読み込まれるように、様々な顧客又は製造施設に供給してもよい。例えば、ARMホールディングス社によって開発されたCortex(商標)ファミリーのプロセッサ及びChinese Academy of SciencesのInstitute of Computing Technology(ICT)によって開発されたLoongson IPコアなどのIPコアは、Texas Instruments社、Qualcomm社、Apple社、又はSamsung社などの様々な顧客又はライセンシーにライセンス又は販売してもよく、且つ、これらの顧客又はライセンシーによって製造されるプロセッサ内において実装してもよい。

40

【0108】

50

図11は、一実施形態によるIPコアの開発を示すブロック図を示している。ストレージ1130は、シミュレーションソフトウェア1120及び/又はハードウェア又はソフトウェアモデル1110を含む。一実施形態においては、IPコア設計を表すデータは、メモリ1140（例えば、ハードディスク）、有線接続（例えば、インターネット）1150、又は無線接続1160を介して、ストレージ1130に供給することができる。次いで、シミュレーションツール及びモデルによって生成されたIPコア情報を製造施設に送信することが可能であり、製造施設において、少なくとも1つの実施形態に従って少なくとも1つの命令を実行するように、IPコアを第三者によって製造することができる。

【0109】

いくつかの実施形態においては、1つ又は複数の命令は、第1のタイプ又はアーキテクチャ（例えば、x86）に対応したものであってもよく、且つ、異なるタイプ又はアーキテクチャ（例えば、ARM）のプロセッサ上において変換又はエミュレートしてもよい。従って、一実施形態による命令は、ARM、x86、MIPS、GPU、又はその他のプロセッサタイプ又はアーキテクチャを含む任意のプロセッサ又はプロセッサタイプ上において実行してもよい。

【0110】

図12は、一実施形態による第1タイプの命令が異なるタイプのプロセッサによってエミュレートされる方式を示している。図12において、プログラム1205は、一実施形態による命令と同一の又は実質的に同一の機能を実行してもよいいくつかの命令を含んでいる。但し、プログラム1205の命令は、プロセッサ1215とは異なる又は互換性を有していないタイプ及び/又はフォーマットを有してもよく、これは、プログラム1205内のタイプの命令がプロセッサ1215によってネイティブで実行不能であってもよいことを意味している。但し、エミュレーション論理1210の支援により、プログラム1205の命令は、プロセッサ1215によって実行される能力をネイティブで有する命令に変換される。一実施形態においては、エミュレーション論理は、ハードウェアとして実施されている。別の実施形態においては、エミュレーション論理は、プログラム1205内のこのようなタイプの命令をプロセッサ1215によってネイティブで実行可能なタイプに変換するためのソフトウェアを含む有体の機械可読媒体内において実施されている。その他の実施形態においては、エミュレーション論理は、固定機能の又はプログラム可能なハードウェアと有体の機械可読媒体上に保存されたプログラムの組合せである。一実施形態においては、プロセッサは、エミュレーション論理を含み、その他の実施形態においては、エミュレーション論理は、プロセッサの外部に存在しており、且つ、第三者によって提供される。一実施形態においては、プロセッサは、プロセッサ内に収容された又はプロセッサと関連付けられたマイクロコード又はファームウェアを実行することにより、ソフトウェアを含む有体の機械可読媒体内において実施されたエミュレーション論理を読み込む能力を有する。

【0111】

図13は、本発明の実施形態によるソース命令セット内のバイナリ命令をターゲット命令セット内のバイナリ命令に変換するためのソフトウェア命令コンバータの使用を対比させたブロック図である。図示の実施形態においては、命令コンバータは、ソフトウェア命令コンバータであるが、この代わりに、命令コンバータは、ソフトウェア、ファームウェア、ハードウェア、又はこれらの様々な組合せにおいて実装してもよい。図13は、x86コンパイラ1304を使用することにより、高級言語1302のプログラムをコンパイルし、少なくとも1つのx86命令セットコア1316によってプロセッサによってネイティブで実行されてもよいx86バイナリコード1306を生成してもよいことを示している。少なくとも1つのx86命令セットコア1316を有するプロセッサは、少なくとも1つのx86命令セットコアを有するIntelプロセッサと実質的に同一の結果を実現するべく、(1)Intel x86命令セットコアの命令セットのかなりの部分、又は(2)少なくとも1つのx86命令セットコアを有するIntelプロセッサ上において稼働することを目的としたアプリケーション又はその他のソフトウェアのオブジェクト

10

20

30

40

50

コードバージョンを互換性を有する状態において実行又はその他の方法で処理することにより、少なくとも1つのx86命令セットコアを有するIntelプロセッサと実質的に同一の機能を実行することができる任意のプロセッサを表している。x86コンパイラ1304は、追加のリンケージ処理を伴って又は伴うことなしに、少なくとも1つのx86命令セットコア1316を有するプロセッサ上において実行可能なx86バイナリコード1306（例えば、オブジェクトコード）を生成するように動作可能なコンパイラを表している。同様に、図13は、代替命令セットコンパイラ1308を使用することにより、高級言語のプログラム1302をコンパイルし、少なくとも1つのx86命令セットコア1314を有していないプロセッサ（例えば、カリフォルニア州Sunnyvaleに所在するMIPS Technologies社のMIPS命令セットを実行する且つ/又はカリフォルニア州Sunnyvaleに所在するARMホールディングス社のARM命令セットを実行するコアを有するプロセッサ）によってネイティブで実行してもよい代替命令セットバイナリコード1310を生成してもよいことを示している。命令コンバータ1312は、x86バイナリコード1306を、x86命令セットコア1314を有していないプロセッサによってネイティブで実行されてもよいコードに変換するために使用される。この能力を有する命令コンバータは、製造が困難であるために、この変換されたコードは、代替命令セットバイナリコード1310と同一である可能性は低いが、変換されたコードは、全般的な演算を実現することになり、且つ、代替命令セットからの命令から構成されることになる。従って、命令コンバータ1312は、エミュレーション、シミュレーション、又は任意のその他のプロセスを通じて、x86命令セットプロセッサ又はコアを有していないプロセッサ又はその他の電子装置がx86バイナリコード1306を実行できるようにするソフトウェア、ファームウェア、ハードウェア、又はこれらの組合せを表している。

【0112】

図14は、複数の試験ソースに対するOR試験及びAND試験機能を提供するために命令をフュージングするためのプロセッサマイクロアーキテクチャの代替実施形態の要素を示している。プロセッサマイクロアーキテクチャ1400の実施形態は、複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令の実行のためのパイプライン400の一部分又はコア490の一部分（例えば、フロントエンドユニット430及び実行エンジンユニット450）であってもよい。装置1400の実施形態は、複数の試験ソースに対するOR試験及びAND試験機能を提供するために命令を復号化するための復号化ステージ（例えば、復号化406）又はデコーダ（例えば、復号化ユニット440）と結合してもよく、命令は、第1ソースデータオペランド、第2ソースデータオペランド、第3ソースデータオペランド、及び演算タイプを規定している。1つ又は複数の実行ユニット（例えば、実行ユニット462及び/又は実行ユニット464）は、復号化された命令に応答し、第1及び第2ソースデータオペランドからのデータの間において、規定の演算タイプに従って、第1論理演算（例えば、ビット論理OR又はビット論理AND）を実行し、且つ、第3ソースデータオペランドからのデータと第1論理演算の結果の間において第2論理演算（例えば、試験演算、又は非破壊的ビット論理AND）を実行し、条件フラグを設定する。

【0113】

例えば、プロセッサパイプライン1400の実施形態は、フェッチステージ1402、命令キューステージ1403、長さ復号化ステージ1404、マイクロ命令フュージョンステージ1405、復号化ステージ1406、micro-op保存ステージ1407、micro-opフュージョンステージ1411、スケジューリング（ディスパッチ又は発行とも呼ばれる）ステージ1412、実行ステージ1416、及び書戻し/メモリ書込みステージ1418を含む。又、プロセッサパイプライン1400の実施形態は、複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令のフュージングとの関係においてプロセッサパイプライン1400の機能的側面を理解するために不要なその他のパイプラインステージ（この図には示されていない）を含んでもよい。

【 0 1 1 4 】

プロセッサパイプライン 1 4 0 0 のいくつかの実施形態においては、復号化ステージ 1 4 0 6 が複数の試験ソースに対する O R 試験及び A N D 試験機能を提供するために第 1 命令を復号化する際に、第 1 命令を、マクロ命令フュージョンステージ 1 4 0 5 によって命令キューステージ 1 4 0 3 からの第 1 命令と共にフュージング可能であると識別された分岐命令とフュージングし、且つ、プロセッサパイプライン 1 4 0 0 における実行のために `micro-op` 保存ステージ 1 4 0 7 内に保存されるように、単一のフュージングされたマイクロ演算を生成している。

【 0 1 1 5 】

プロセッサパイプライン 1 4 0 0 のいくつかの代替実施形態においては、フェッチステージ 1 4 0 2 は、第 1 オペランドデスティネーションを規定する第 1 論理命令（例えば、ビット論理 O R 又はビット論理 A N D 命令）と、第 2 オペランドソースを規定する第 2 命令（例えば、ビット論理 A N D 又は非破壊的試験命令）と、分岐条件を規定する第 3 命令（例えば、J Z (J u m p - Z e r o)、J N Z (J u m p - N o t - Z e r o)、又はいくつかのその他の J C C (J u m p - C o n d i t i o n - C o d e) 命令）を含むいくつかの命令をフェッチしてもよい。フェッチステージ 1 4 0 2 は、命令キューステージ 1 4 0 3 に対して命令を保存し、長さ復号化ステージ 1 4 0 4 及びマクロ命令フュージョンステージ 1 4 0 5 の復号化前論理が、復号化ステージ 1 4 0 6 内における復号化のために、命令の境界を判定し、且つ、フュージング可能な命令を識別する。復号化ステージ 1 4 0 6 は、命令の一部分を単一のマイクロ演算としてフュージングしてもよく、この部分は、第 1 オペランドデスティネーションと前記第 2 オペランドソースが同一であり（例えば、シングルレジスタ）、且つ、前記分岐条件が第 2 命令に依存している場合には（例えば、第 2 命令が、分岐条件として第 3 命令によって使用される条件コードを変更する最後の命令である場合）、第 1 及び第 2 命令の両方を含む。いくつかの実施形態においては、フュージングされる部分は、（例えば、第 1 オペランドデスティネーションと前記第 2 オペランドソースが同一であり、第 2 及び第 3 命令が連続しており、且つ、第 2 命令が試験命令である場合）、第 1、第 2、及び第 3 命令を含んでもよい。いくつかの代替実施形態においては、フュージングされる部分は、第 1 及び第 2 命令のみを含んでもよく、復号化ステージ 1 4 0 6 は、プロセッサパイプライン 1 4 0 0 における実行のために `micro-op` 保存ステージ 1 4 0 7 内に保存されるように、第 1 及び第 2 命令を、複数の試験ソースに対する O R 試験及び A N D 試験機能を提供するための単一のマイクロ演算として、復号化している。

【 0 1 1 6 】

又、プロセッサパイプライン 1 4 0 0 のいくつかの代替実施形態においては、分岐条件が複数の試験ソースに対する O R 試験又は A N D 試験機能を提供するための単一の第 1 マイクロ演算に依存している場合には、`micro-op` 保存ステージ 1 4 0 7 を `micro-op` フュージョンステージ 1 4 1 1 と結合し、前記単一の第 1 マイクロ演算と前記第 3 命令の第 2 マイクロ演算をフュージングしてもよい。このような一実施形態においては、スケジューリングステージ 1 4 1 2 は、1 つのみのフュージングされたマイクロ演算を受け取り、複数の試験ソースに対する O R 試験又は A N D 試験機能と、結果に応じた分岐と、を提供するためのこの 1 つのフュージングされたマイクロ演算を、実行ステージ 1 4 1 6 に、且つ、次いで、書戻し/メモリ書込みステージ 1 4 1 8 に、発行してもよい。

【 0 1 1 7 】

いくつかの頻繁に実行される命令シーケンスのイディオムは、2 つのソースデータオペランドを合成するための論理演算（例えば、いずれかのソース内のビットが 1 であるかどうかを試験する際の O R、いずれかのソース内のビットがゼロであるかどうかを試験する際の A N D）、比較又は試験命令（例えば、マスクに照らして論理演算の結果を T E S T するためのもの）、及び条件付き分岐（例えば、ゼロフラグが T E S T によって設定されている/設定されていない場合に相対的に低速の一般的なコードシーケンスにジャンプするための J Z / J N Z）を含んでもよい。プロセッサ復号化論理は、比較命令又は試験命

10

20

30

40

50

令（例えば、TEST）を条件付き分岐命令（例えば、JZ）と合成する際に、マクロフュージョンを利用してよい。

【0118】

又、いくつかの実施形態は、1つの論理命令（例えば、OR）を従来技術の試験命令とフュージングすることにより、試験命令を動的に生成してもよい。その他の実施形態は、試験命令を、動的に、但し、JITコンパイラを通じて、生成している。又、いくつかの実施形態は、試験命令を後続の条件付き分岐命令（例えば、JZ）とフュージングしてもよく、且つ、条件フラグが設定された方式に従って分岐を実行してもよい。従って、頻繁に実行される命令シーケンスのイディオムは、実行時においてプロセッサ内に1つのマイクロ演算シーケンスを結果的にもたらず単一の命令としてフュージング可能であってもよい。

10

【0119】

複数の試験ソースに対するOR試験及び/又はAND試験機能を提供するためのフュージング可能な命令及び論理は、JITコンパイラによって生成される頻繁に実行されるタイプ試験命令イディオムと関連する性能の制限の問題、エネルギー消費の懸念、及びその他の実行時ボトルネックをほとんど除去することになることを理解されたい。

【0120】

図15Aは、複数の試験ソースに対するOR試験及びAND試験機能を提供するためのフュージングされた命令構造1501の一実施形態のフロー図を示している。第1オペランドデスティネーションを規定する第1論理命令（例えば、ビット論理OR又はビット論理AND命令）であるTEMPと、第2オペランドソースを規定する第2命令（例えば、非破壊的TEST命令）であるTESTが（例えば、フェッチステージ1402によって）命令キューIQ1510内に保存されている。又、分岐条件を規定する第3命令1514（例えば、JCC（Jump-Condition-Code）命令）も、（例えば、命令キューステージ1403において）IQ1510内に保存されている。（例えば、長さ復号化ステージ1404及びマイクロ命令フュージョンステージ1405の）復号化前論理が、（例えば、復号化ステージ1406における）復号化のために、命令境界を判定し、且つ、フュージング可能な命令1512を識別している。（例えば、復号化ステージ1406における）復号化の際に、命令（例えば、命令1512）の一部分を単一のマイクロ演算1522としてフュージングしてもよく、この部分は、第1オペランドデスティネーション及び前記第2オペランドソースが同一であり（例えば、単一のレジスタTEMP）、且つ、分岐条件コード（CC）が第2命令に依存している場合には（例えば、第2命令が、分岐条件として第3命令によって使用されるCCを変更する最後の命令である場合）、第1及び第2命令1512の両方を含む。いくつかの代替実施形態においては、フュージングされる部分は、第1及び第2命令1512のみを含んでもよく、プロセッサは、（例えば、プロセッサパイプライン1400における実行のために、micro-op保存ステージ1407において）micro-opストレージ1520内に保存されるように、第1及び第2命令1512を複数の試験ソースに対するOR試験又はAND試験機能を提供するための単一のマイクロ演算1522として復号化している。又、いくつかの実施形態においては、分岐条件が複数の試験ソースに対するOR試験又はAND試験機能を提供するための単一の第1マイクロ演算1522に依存している場合には、micro-opストレージ1520を、前記単一の第1マイクロ演算1522と前記第3命令1514の第2マイクロ演算1524を（例えば、micro-opフュージョンステージ1411内において）フュージングするためのmicro-opフュージョン論理と結合してもよい。このような一実施形態においては、複数の試験ソースに対するOR試験又はAND試験機能と、結果として得られるCCに応じた分岐と、を提供するための1つのフュージングされたマイクロ演算1526を（例えば、プロセッサパイプライン1400における実行のために）micro-opストレージ1520に保存してもよい

20

30

40

【0121】

いくつかの実施形態においては、フュージングされる部分は、（例えば、第1オペラン

50

ドデスティネーション及び前記第2オペランドソースが同一あり、第2及び第3命令が連続しており、且つ、第2又はミドル命令が試験命令である場合には)、第1、第2、及び第3命令を含んでもよい。

【0122】

図15Bは、複数の試験ソースに対するOR試験及びAND試験機能と、結果的に得られるCCに応じた分岐と、を提供するためのフュージングされた命令構造1502の代替実施形態のフロー図を示している。第1オペランドデスティネーションを規定する第1論理命令(例えば、ビット論理OR又はビット論理AND命令)であるTEMP、第2オペランドソースを規定する第2命令(例えば、非破壊的TEST命令)であるTEST、及び分岐条件を規定する第3命令(例えば、JCC(Jump-Condition-Code)命令)が(例えば、命令キューステージ1403においてフェッチステージ1402によって)IQ1510内に保存されている。(例えば、長さ復号化ステージ1404及びマイクロ命令フュージョンステージ1405の)復号化前論理が、(例えば、復号化ステージ1406における)復号化のために、命令の境界を判定し、且つ、フュージング可能な命令1516を識別している。(例えば、復号化ステージ1406における)復号化の際に、命令(例えば、命令1516)の一部を単一のマイクロ演算1526としてフュージングしてもよく、この部分は、第1オペランドデスティネーションと第2オペランドソースが同一であり(例えば、TEMP)、第2(例えば、TEST)及び第3(例えば、JCC)命令が連続しており、且つ、第2の又は中間の命令が、(例えば、プロセッサパイプライン1400内における)実行の結果として分岐CCを変更するTESTなどの命令である場合には、第1、第2、及び第3命令1516を含む。

【0123】

いくつかの実施形態においては、プロセッサ命令セットアーキテクチャ(ISA)は、複数の試験ソースに対するOR試験及びAND試験機能を提供するための代替タイプのマクロ命令TEST2を提供してもよい。このような一実施形態においては、単一のマイクロ演算としてフュージングすることができる命令の一部を識別するタスクが単純化されることになろう。

【0124】

図15Cは、複数の試験ソースに対するOR試験及びAND試験機能と、結果的に得られるCCによる分岐と、を提供するためのフュージングされた命令構造1503の別の代替実施形態のフロー図を示している。第1ソースデータオペランド、第2ソースデータオペランド、第3ソースデータオペランド、及び演算タイプ(例えば、OR試験又はAND試験命令)を規定する第1命令TEST2と、分岐条件を規定する第2分岐命令(例えば、JCC(Jump-Condition-Code)命令)が(例えば、命令キューステージ1403においてフェッチステージ1402によって)IQ1511内に保存され、復号化ステージ(復号化ステージ1406)が第1命令を復号化する際に、複数の試験ソースに対するOR試験又はAND試験機能と、結果的に得られるCCに応じた分岐と、を提供するための単一のフュージングされたマイクロ演算1526としての実行のために、第1命令を分岐命令とフュージングしており、単一のフュージングされたマイクロ演算1526は、(例えば、プロセッサパイプライン1400における実行のために、micro-op保存ステージ1407において)micro-opストレージ1520内に保存されている。

【0125】

従って、フュージング可能な命令及び論理の実施形態は、複数の試験ソースに対するOR試験及びAND試験機能を提供してもよい。いくつかの実施形態においては、プロセッサ復号化ステージ1406は、実行のために新しい試験命令(例えば、OR試験又はAND試験など)を復号化してもよく、命令は、第1、第2、及び第3ソースデータオペランドと、演算タイプと、を規定している(例えば、OR試験又はAND試験)。実行ユニット(例えば、実行ユニット462及び/又は実行ユニット464)は、復号化された試験命令に回答し、規定の演算タイプに従って、第1及び第2ソースデータオペランドからの

10

20

30

40

50

データの間において、1つの論理演算（例えば、OR）を実行してもよく、且つ、第3ソースデータオペランドからのデータと第1論理演算の結果の間において、第2論理演算（例えば、AND）を実行し、条件フラグを設定してもよい。いくつかの代替実施形態においては、複数の試験ソースに対するOR試験及びAND試験機能を提供するための処理は、専用のハードウェアによって実行してもよい。いくつかのその他の代替実施形態においては、このような処理は、汎用機械による又は特殊目的機械による又はなんらかの組合せによる実行が可能なソフトウェア又はファームウェア演算コードによって実行してもよい。

【0126】

図16Aは、複数の試験ソースに対するOR試験及びAND試験機能を提供するために命令をフュージングするためのプロセス1601の一実施形態のフロー図を示している。上述のように、本明細書に記述されているプロセス1601及びその他のプロセスは、汎用機械による、又は特殊目的機械による、又は両方の組合せによる実行が可能な専用ハードウェア又はソフトウェア又はファームウェア演算コードを有してもよい処理ブロックによって実行される。

10

【0127】

プロセス1601の処理ブロック1610において、第1オペランドデスティネーションを規定する第1命令がフェッチされる。処理ブロック1620において、第2オペランドソースを規定する第2命令がフェッチされる。処理ブロック1630において、第1及び第2オペランドが同一であるかどうか判定される。判定結果が否定的である場合には、処理は、処理ブロック1610に進む。判定結果が肯定的である場合には、処理は、処理ブロック1640に進み、ここで、第1及び第2命令が単一のmicro-op又はマイクロ演算としてフュージングされる。処理ブロック1650において、第2命令に依存する分岐条件を規定する第3命令がフェッチされる。次いで、処理ブロック1660において、第3命令も、単一のmicro-op又はマイクロ演算としてフュージングされる。

20

【0128】

図16Bは、複数の試験ソースに対するOR試験及びAND試験機能を提供するために命令をフュージングするためのプロセス1602の代替実施形態のフロー図を示している。プロセス1602の処理ブロック1610において、第1オペランドデスティネーションを規定する第1命令がフェッチされる。処理ブロック1620において、第2オペランドソースを規定する第2命令がフェッチされる。処理ブロック1650において、第2命令に依存した分岐条件を規定する第3命令がフェッチされる。次いで、処理ブロック1670において、フェッチされた命令の一部がフュージングされており、この部分は、前記第1オペランドデスティネーション及び前記第2オペランドソースが同一であり、且つ、分岐条件が第2命令に依存している場合には、第1及び第2命令の両方を含む。

30

【0129】

図16Cは、複数の試験ソースに対するOR試験及びAND試験機能を提供するために命令をフュージングするためのプロセスの別の代替実施形態1603のフロー図を示している。プロセス1603の処理ブロック1645において、2つのオペランドを試験するための第1命令であって、第1オペランドソース、第2オペランドソース、第3オペランドソース、及び演算タイプを規定する第1命令が、フェッチされる。処理ブロック1655においては、分岐条件フラグを規定する第2命令がフェッチされ、分岐条件フラグは、第1命令に依存している。次いで、処理ブロック1665において、フェッチされた命令の一部がフュージングされており、この部分は、例えば、分岐条件フラグが第1命令に依存している場合には、第1及び第2命令の両方を含む。

40

【0130】

図16Dは、複数の試験ソースに対するOR試験及びAND試験機能を提供するために命令をフュージングするためのプロセスの別の代替実施形態1604のフロー図を示している。プロセス1604の処理ブロック1647において、第1命令が実行のために復号

50

化されており、第1命令は、第1オペランドソースデータ、第2オペランドソースデータ、第3オペランドソースデータ、及び演算タイプを規定している。処理ブロック1649において、第1命令について、第1 `micro-op` 又はマイクロ演算が保存されている。処理ブロック1657において、第2命令が実行のために復号化されており、第2命令は、分岐条件を規定している。処理ブロック1659において、第2命令について、第2 `micro-op` 又はマイクロ演算が保存されている。処理ブロック1667において、分岐条件が第1命令に依存している場合に、第1及び第2 `micro-op` が単一の `micro-op` としてフュージングされる。

【0131】

いくつかの実施形態は、新規の試験命令（例えば、OR試験又はAND試験命令）を、動的に、但し、JITコンパイラを通じて、生成してもよい。又、これらの実施形態は、試験命令を後続の条件付き分岐命令（例えば、JZ）とフュージングしてもよく、且つ、条件フラグが設定された方式に従って分岐を実行してもよい。従って、頻繁に実行される命令シーケンスのイディオムを、実行時にプロセッサパイプライン内に単一のマイクロ演算を結果的にもたらず単一の命令としてフュージングしてもよい。

【0132】

図17は、複数の試験ソースに対するOR試験及びAND試験機能を提供するための命令を実行するためのプロセスの一実施形態1701のフロー図を示している。プロセス1701の処理ブロック1710において、第1命令が実行のために復号化されており、第1命令は、第1オペランドソースデータ、第2オペランドソースデータ、第3オペランドソースデータ、及び演算タイプを規定している。処理ブロック1720において、1つ又は複数の実行ユニットは、復号化された第1命令に应答し、規定の演算タイプに従って、第1及び第2オペランドソースからのデータの間において、第1論理演算を実行している。処理ブロック1730において、第3オペランドソースからのデータと第1論理演算の結果の間において、第2演算が実行され、条件フラグを条件付きで設定している。処理ブロック1740において、条件フラグが設定されている場合に、プログラムのフローが条件付きで分岐している。いくつかの実施形態においては、複数の試験ソースに対するOR試験及びAND試験機能と、結果的に得られるCCに応じた分岐と、を提供するための単一のフュージングされた `micro-op` 又はマイクロ演算に应答し、処理ブロック1720～1740を実行してもよい。

【0133】

頻繁に実行される命令シーケンスのイディオムは、しばしば、2つのソースデータオペランドを合成するための論理演算（例えば、いずれかのソース内のビットが1であるかどうかを試験する際のOR、いずれかのソース内のビットがゼロであるかどうかを試験する際のAND）、比較又は試験命令（例えば、マスクに照らして論理演算の結果をTESTするためのもの）、及び条件付き分岐（例えば、ゼロフラグがTESTによって設定されている/設定されていない場合に相対的に低速の一般的コードシーケンスにジャンプするためのJZ/JNZ）を含んでいる場合がある。プロセッサ復号化論理は、比較命令又は試験命令（例えば、TEST）を条件付き分岐命令（例えば、JZ）と合成する際に、マクロフュージョンを利用してもよい。

【0134】

複数の試験ソースに対するOR試験及び/又はAND試験機能を提供するためのフュージング可能な命令及び論理は、JITコンパイラによって生成される頻繁に実行されるタイプ試験命令イディオムと関連する性能の制限の問題、エネルギー消費の懸念、及びその他の実行時ボトルネックをほとんど除去することになることを理解されたい。

【0135】

本明細書に開示されているメカニズムの実施形態は、ハードウェア、ソフトウェア、ファームウェア、又はこのような実装方式の組合せによって実装してもよい。本発明の実施形態は、少なくとも1つのプロセッサと、ストレージシステム（揮発性及び不揮発性メモリ及び/又はストレージ要素を含む）と、少なくとも1つの入力装置と、少なくとも1つ

10

20

30

40

50

の出力装置と、を有するプログラム可能なシステム上において稼働するコンピュータプログラム又はプログラムコードとして実装してもよい。

【0136】

プログラムコードを入力命令に適用し、本明細書に記述されている機能を実行すると共に出力情報を生成してもよい。出力情報は、既知の方式により、1つ又は複数の出力装置に適用してもよい。このアプリケーションの目的のために、処理システムは、例えば、デジタル信号プロセッサ(DSP)、マイクロコントローラ、ASIC(Application Specific Integrated Circuit)、又はマイクロプロセッサなどのプロセッサを有する任意のシステムを含む。

【0137】

プログラムコードは、処理システムと通信するために、ハイレベルな手続型の又はオブジェクト指向のプログラム言語によって実装してもよい。又、プログラムコードは、必要に応じて、アセンブリ言語又は機械語によって実装してもよい。実際に、本明細書に記述されているメカニズムは、その範囲が任意の特定のプログラム言語に限定されるものではない。いずれの場合にも、言語は、コンパイラ型又はインタープリタ型の言語であってよい。

【0138】

少なくとも1つの実施形態の1つ又は複数の態様は、機械によって読み取られた際に本明細書に記述されている技法を実行するための論理を機械に製造させる、プロセッサ内における様々な論理を表す、機械可読媒体上に保存された命令表現によって実装してもよい。このような「IPコア」と呼ばれる表現は、有体の機械可読媒体上に保存してもよく、且つ、論理又はプロセッサを実際に製造する製造機械に読み込まれるように、様々な顧客又は製造施設に供給してもよい。

【0139】

このような機械可読ストレージ媒体は、限定を伴うことなしに、ハードディスク、フロッピー(登録商標)ディスク、光ディスク、CD-ROM(Compact Disk Read-Only Memory)、CD-RW(Compact Disk Rewritable)、及び磁気光ディスクを含む任意のその他のタイプのディスク、ROM(Read-Only Memory)、RAM(Random Access Memory)、DRAM(Dynamic Random Access Memory)、SRAM(Static Random Access Memory)、EPROM(Erasable Programmable Read-Only Memory)、フラッシュメモリ、EEPROM(Electrically Erasable Programmable Read-Only Memory)などの半導体装置、磁気又は光カード、或いは、電子的命令を保存するのに適した任意のその他のタイプの媒体などのストレージ媒体を含む機械又は装置によって製造又は形成された物品の一時的ではない有体の構成を含んでもよい。

【0140】

従って、本発明の実施形態は、命令を収容する又は本明細書に記述されている構造、回路、装置、プロセッサ、及び/又はシステム機能を定義するHDL(Hardware Description Language)などの設計データを収容する一時的ではない有体の機械可読媒体をも含む。又、このような実施形態は、プログラムプロダクトと呼ばれる場合がある。

【0141】

いくつかのケースにおいては、命令コンバータを使用し、命令をソース命令セットからターゲット命令セットに変換してもよい。例えば、命令コンバータは、命令を、コアによって処理される1つ又は複数のその他の命令に(静的バイナリ変換や動的コンパイルを含む動的なバイナリ変換を使用して)変換、変形、エミュレート、又はその他の方法で変換してもよい。命令コンバータは、ソフトウェア、ハードウェア、ファームウェア、又はこれらの組合せによって実装してもよい。命令コンバータは、オンプロセッサ、オフプロセ

10

20

30

40

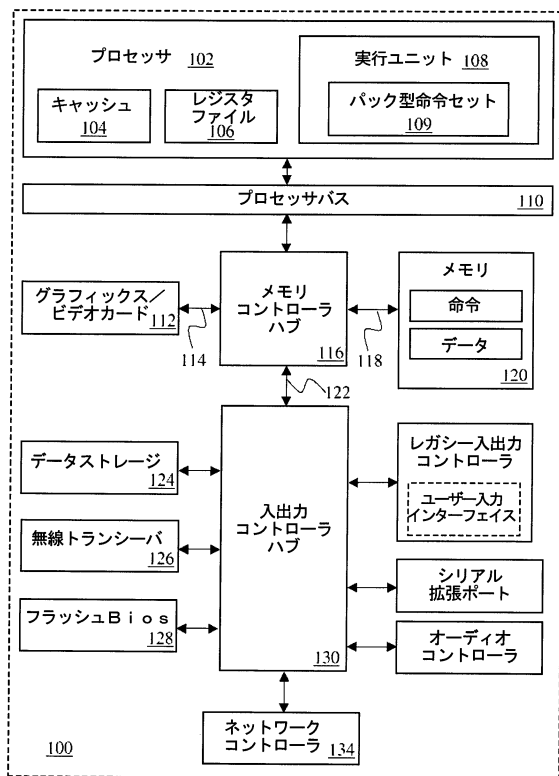
50

ッサ、或いは、パートオン及びパートオフプロセッサであってもよい。

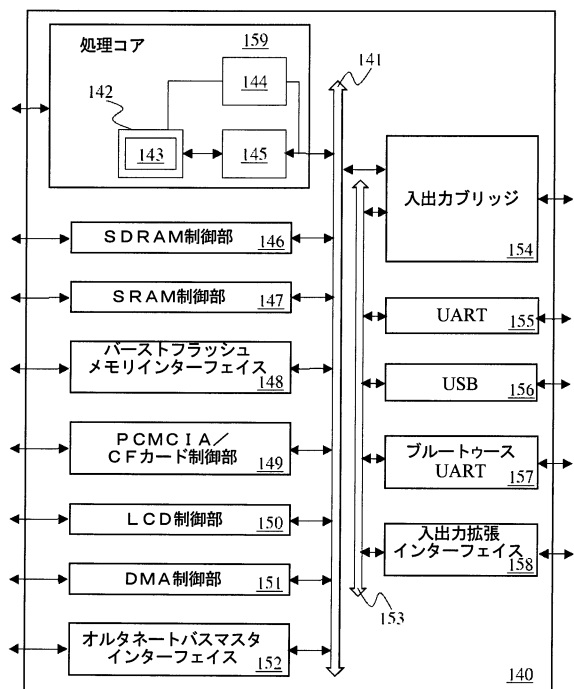
【0142】

従って、少なくとも1つの実施形態に従って1つ又は複数の命令を実行するための技法が開示されている。特定の例示用の実施形態について記述すると共に添付図面に図示したが、このような実施形態は、広範な本発明の限定ではなく、その例示を目的としたものに過ぎず、且つ、当業者には、本開示を参照した際に様々なその他の変更が想起されうることから、本発明は、図示及び記述されている特定の構造及び構成に限定されるものではないことを理解されたい。迅速に成長すると共に更なる進歩が容易には予想されない当技術分野などの技術分野においては、本開示の原理又は添付の請求項の範囲を逸脱することなしに、技術的な進歩の実現によって促進されるかたちで、開示されている実施形態の構成及び詳細の変更が容易であろう。

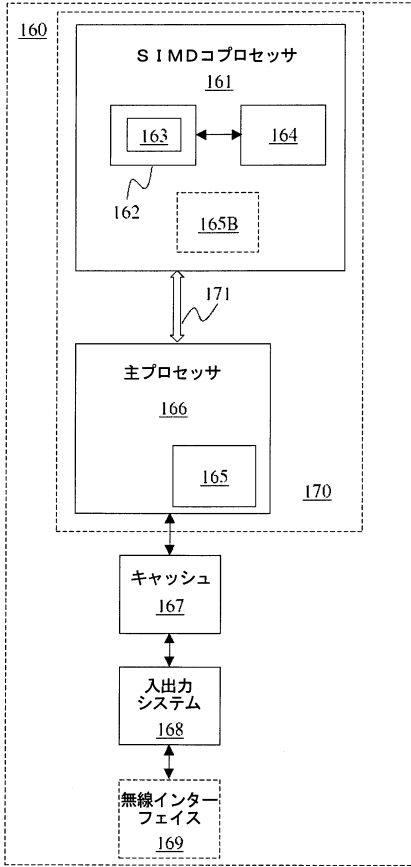
【図1A】



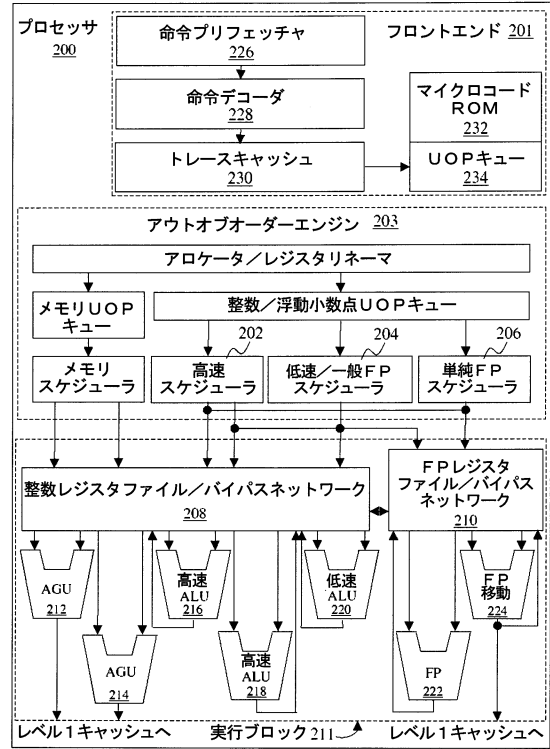
【図1B】



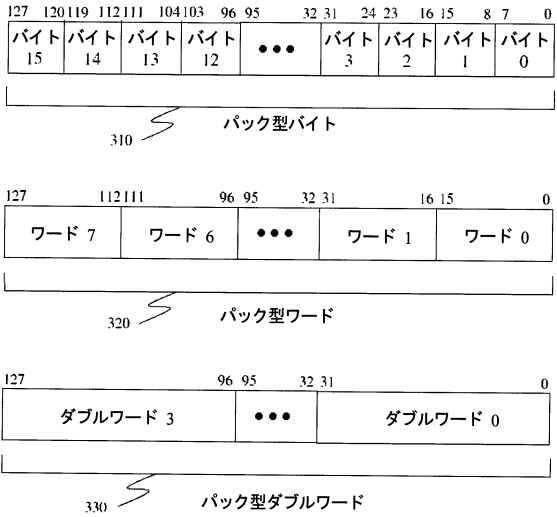
【図1C】



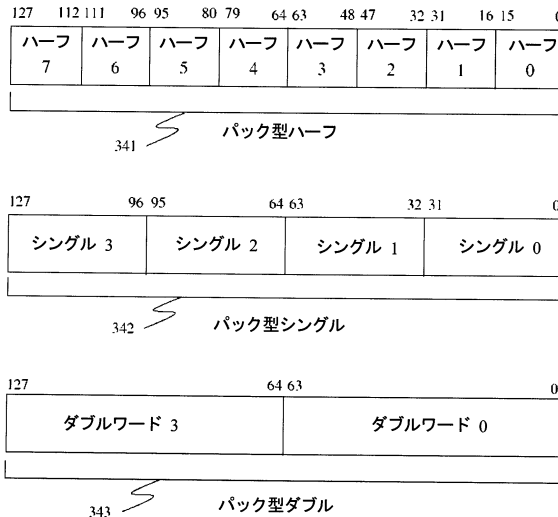
【図2】



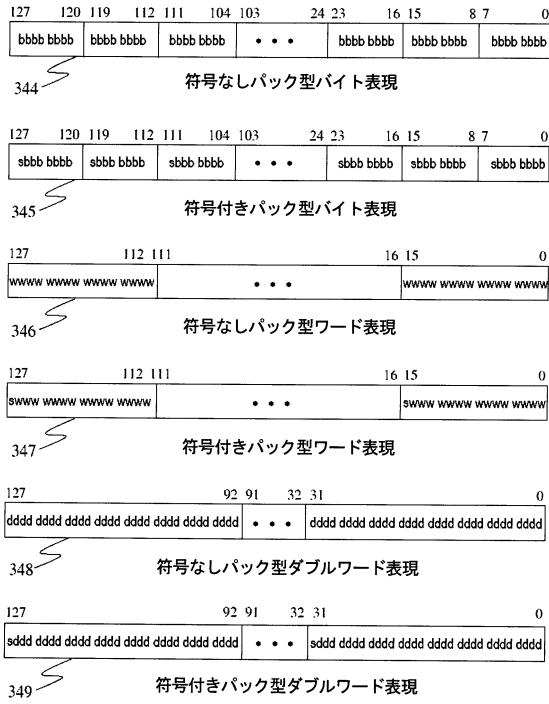
【図3A】



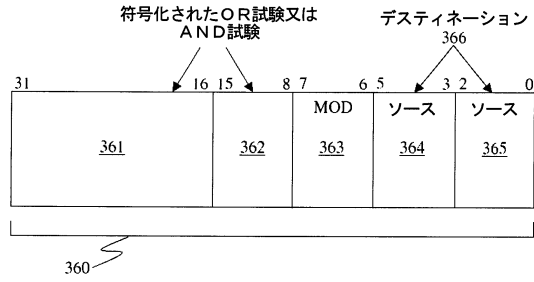
【図3B】



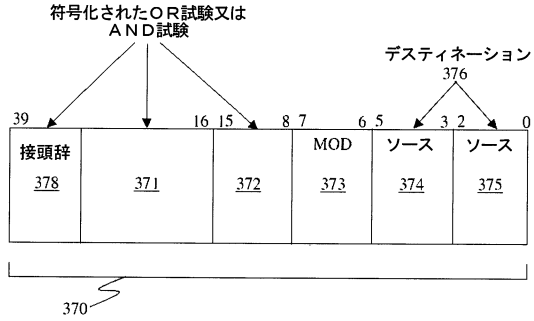
【図 3 C】



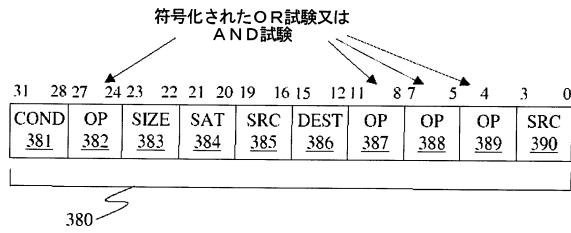
【図 3 D】



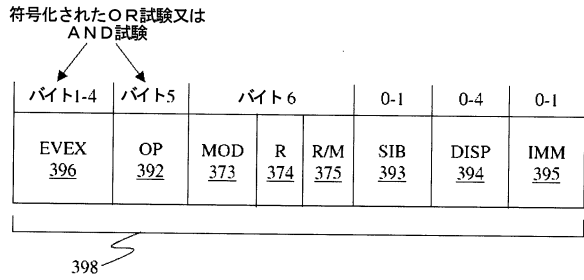
【図 3 E】



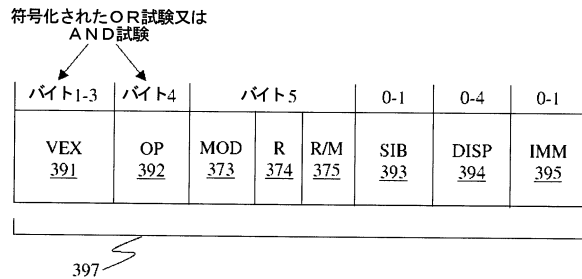
【図 3 F】



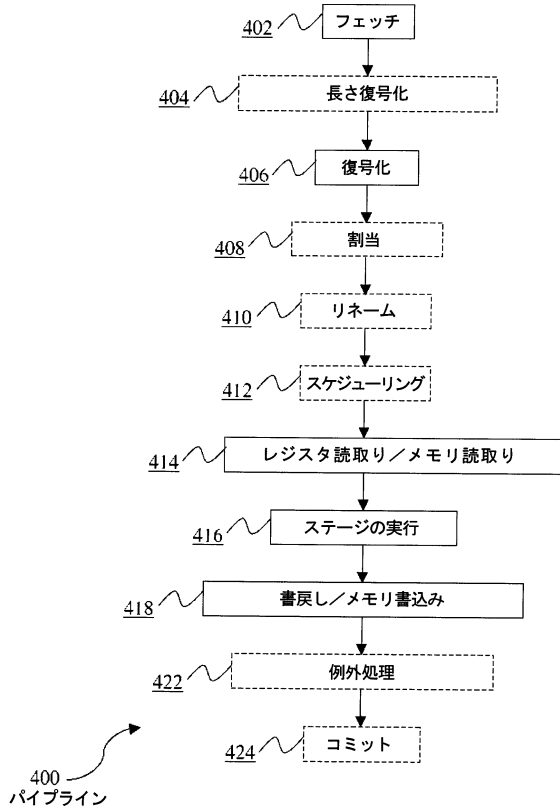
【図 3 H】



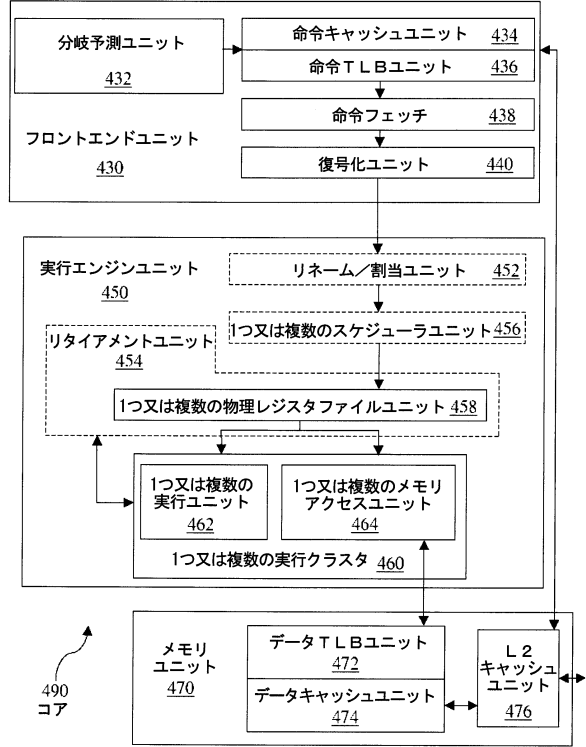
【図 3 G】



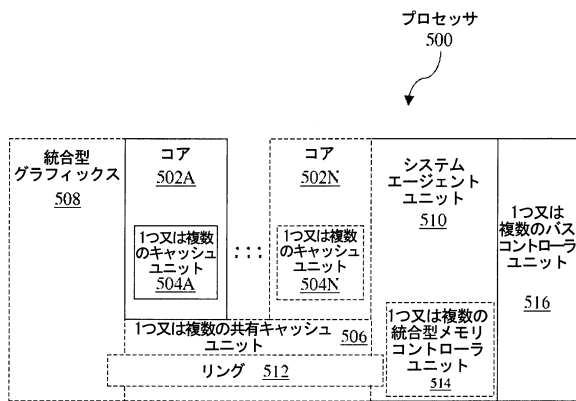
【図4A】



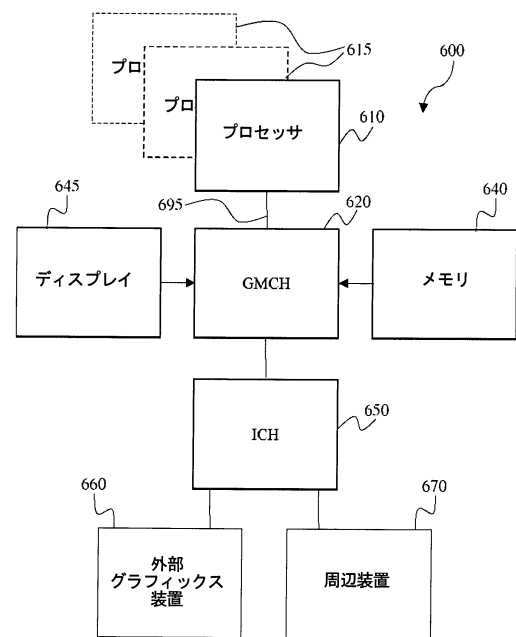
【図4B】



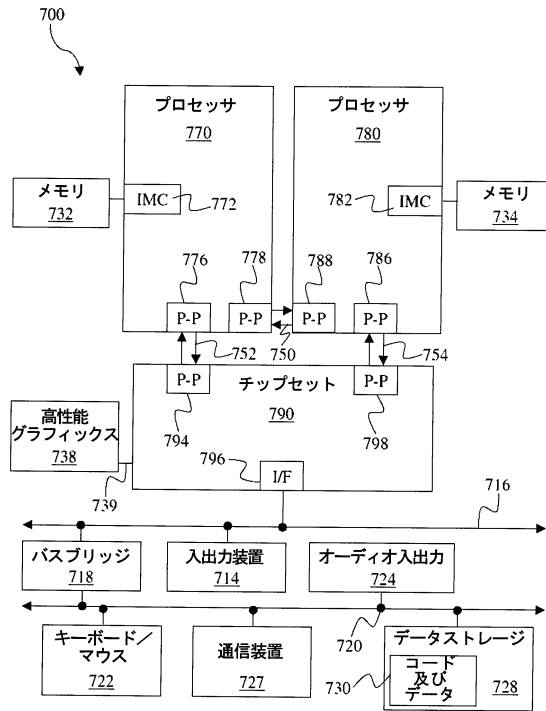
【図5】



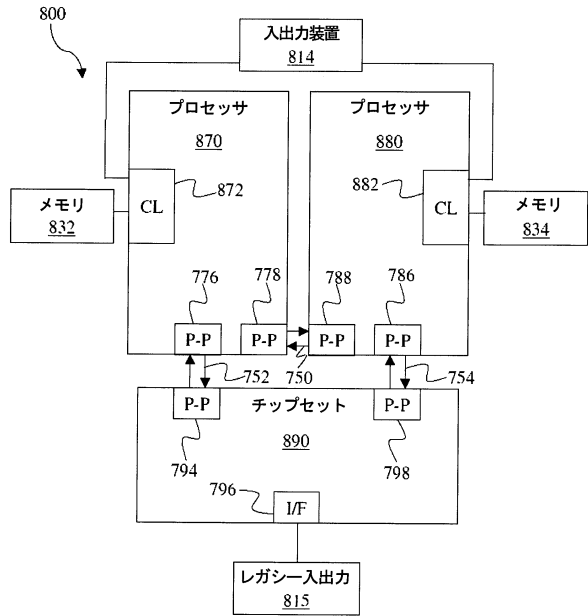
【図6】



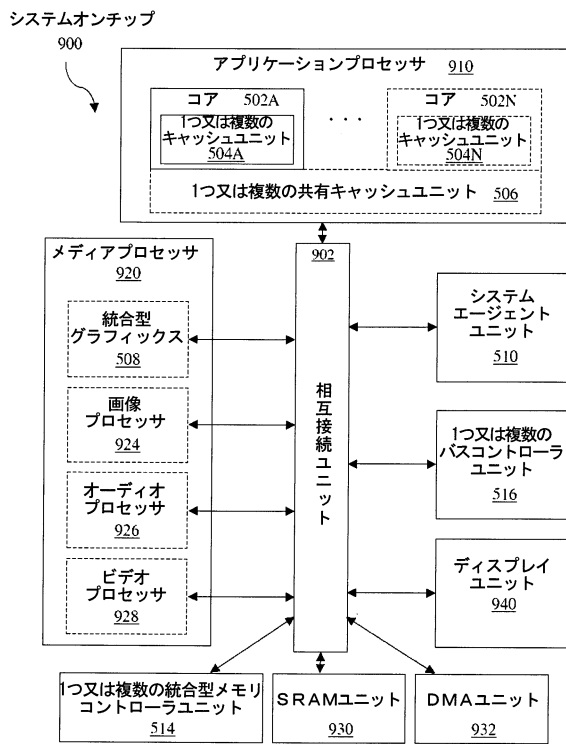
【図7】



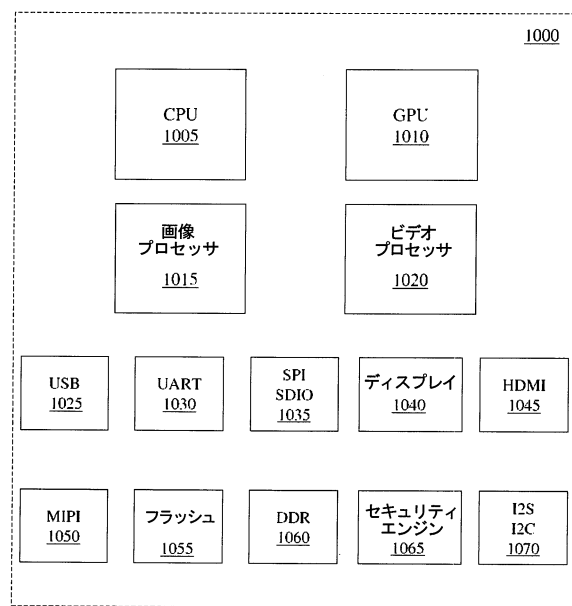
【図8】



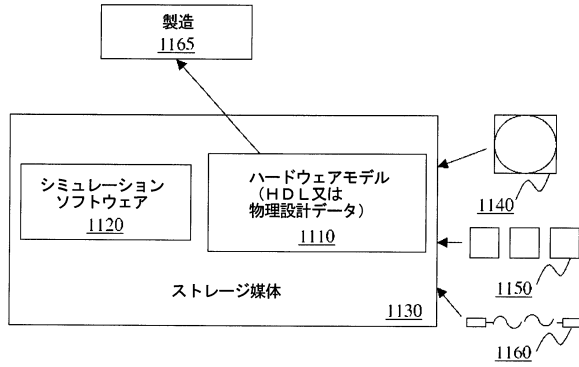
【図9】



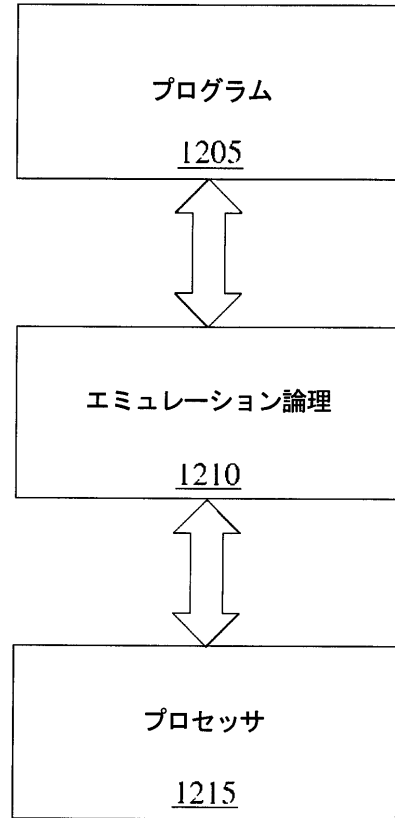
【図10】



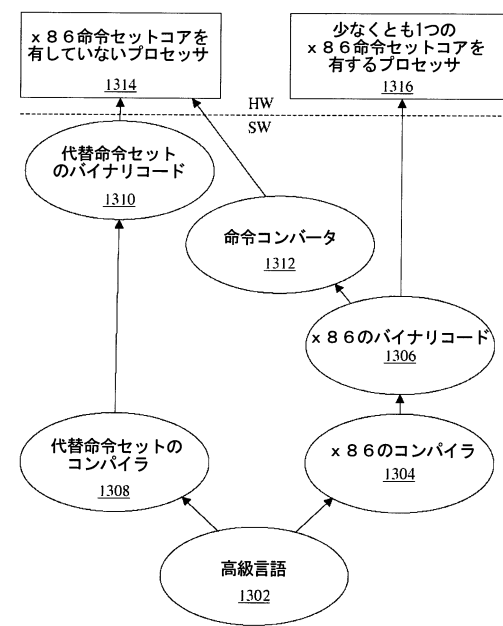
【図 1 1】



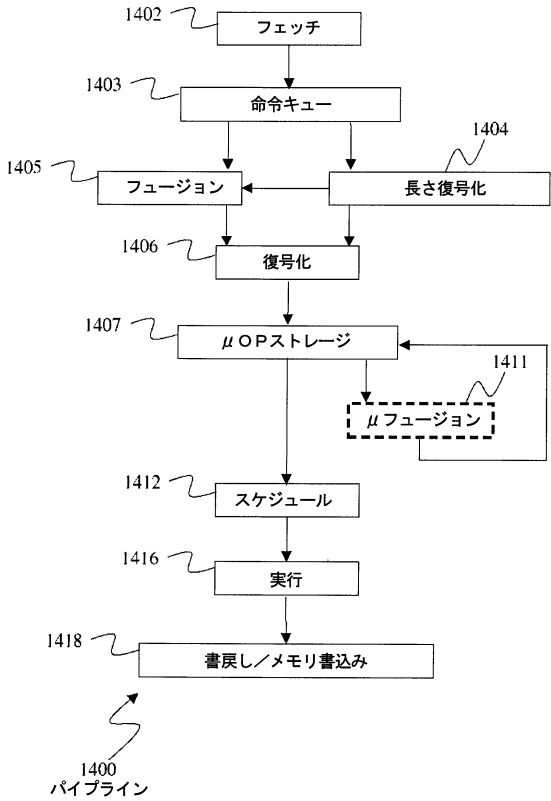
【図 1 2】



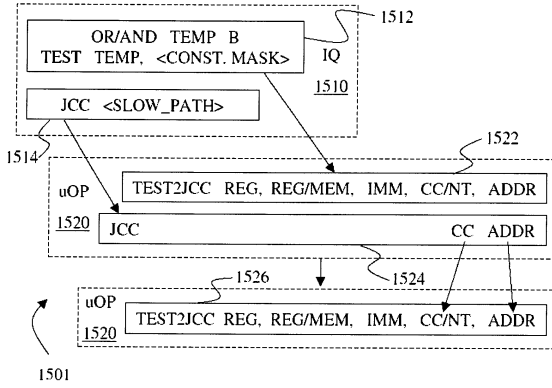
【図 1 3】



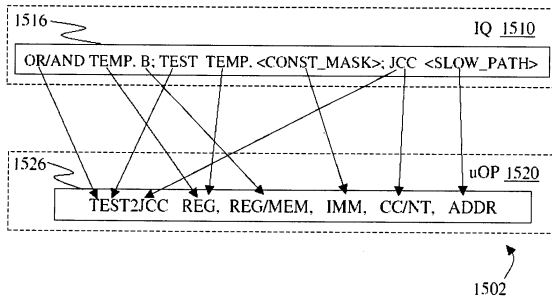
【図 1 4】



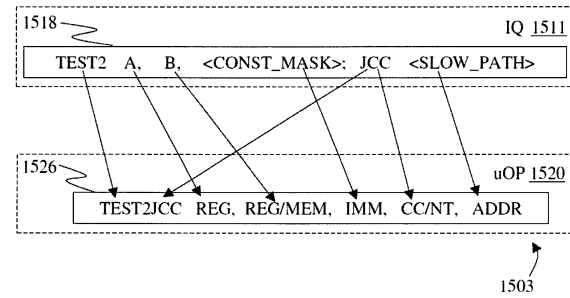
【図15A】



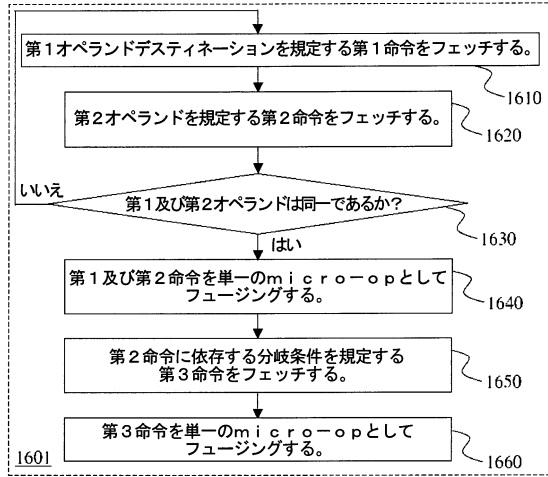
【図15B】



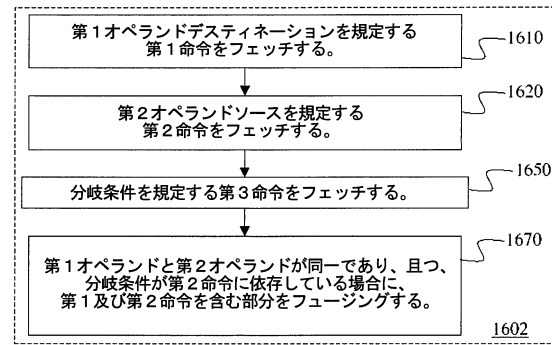
【図15C】



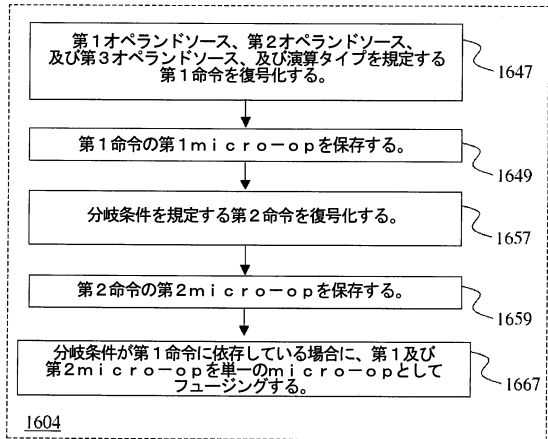
【図16A】



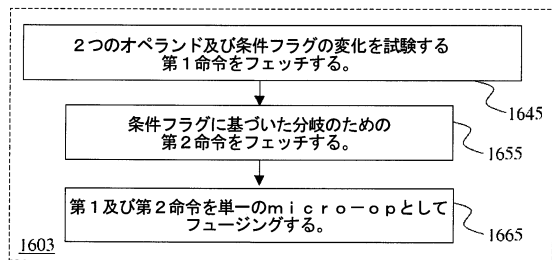
【図16B】



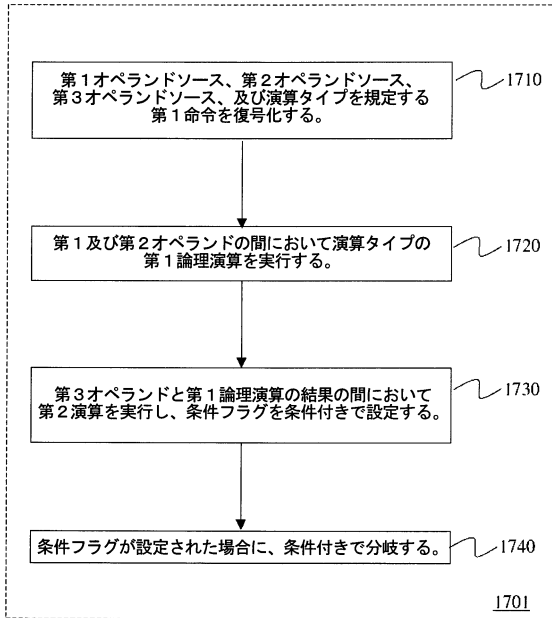
【図16D】



【図16C】



【図 17】



フロントページの続き

(51)Int.Cl. F I
G 0 6 F 9/22 (2006.01) G 0 6 F 9/38 3 3 0 X
 G 0 6 F 9/22 3 2 0 A

- (72)発明者 バレンタイン、ロバート
 アメリカ合衆国 9 5 0 5 4 カリフォルニア州・サンタクララ・ミッション カレッジ ブレー
 バード・2 2 0 0 インテル・コーポレーション内
- (72)発明者 ホーン、ジュリアン シー、
 アメリカ合衆国 9 5 0 5 4 カリフォルニア州・サンタクララ・ミッション カレッジ ブレー
 バード・2 2 0 0 インテル・コーポレーション内
- (72)発明者 チャーニー、マーク ジェイ、
 アメリカ合衆国 9 5 0 5 4 カリフォルニア州・サンタクララ・ミッション カレッジ ブレー
 バード・2 2 0 0 インテル・コーポレーション内

審査官 清木 泰

- (56)参考文献 米国特許出願公開第2007/0038844 (US, A1)
 特開平05-073309 (JP, A)
 特開2016-042382 (JP, A)
 特開2014-194753 (JP, A)
 特表2012-507794 (JP, A)
 特表2010-504595 (JP, A)
 特開平09-311786 (JP, A)
 特開平08-007083 (JP, A)
 特開平04-037926 (JP, A)
 米国特許出願公開第2011/0264897 (US, A1)
 米国特許出願公開第2011/0264891 (US, A1)
 米国特許出願公開第2011/0138156 (US, A1)
 米国特許出願公開第2004/0128483 (US, A1)
 米国特許出願公開第2002/0087955 (US, A1)
 米国特許第06961846 (US, B1)
 米国特許第06889318 (US, B1)
 Pentium Proファミリ ディベロッパーズマニュアル 中巻, インテルジャパン株式会社, 19
 96年, 初版, Pages:11-64~11-65, 11-377~11-378
 大貫 広幸, Try! PC 2003winter, CQ出版株式会社, 2003年 1月 1日, Pages:158~159
 , 175~180
 後藤弘茂, 2006年のCPU 第71回 インテルはCISCのアプローチをアーキテクチャに持ち込む, A
 SCII, 日本, 株式会社アスキー, 2006年 5月 1日, Vol.:30, No.:5, Pages:114~119
 大原雄介, Intel & AMD が語らない深部をベンチで暴く 最新CPU内部解析, 日経 Win PC, 日本
 , 日経BP社, 2009年 9月 1日, 第15巻, 第14号, Pages:108~121

- (58)調査した分野(Int.Cl., DB名)
 G 0 6 F 9 / 3 0 - 9 / 4 2
 G 0 6 F 9 / 2 2 - 9 / 2 8