

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3604697号
(P3604697)

(45) 発行日 平成16年12月22日(2004.12.22)

(24) 登録日 平成16年10月8日(2004.10.8)

(51) Int.C1.⁷

F 1

G06F 9/38

G06F 9/38 380C

G06F 11/28

G06F 11/28 315B

請求項の数 27 (全 14 頁)

(21) 出願番号	特願平8-536745
(86) (22) 出願日	平成8年5月31日(1996.5.31)
(65) 公表番号	特表2000-504444(P2000-504444A)
(43) 公表日	平成12年4月11日(2000.4.11)
(86) 国際出願番号	PCT/US1996/008256
(87) 国際公開番号	W01996/038789
(87) 国際公開日	平成8年12月5日(1996.12.5)
審査請求日	平成14年9月27日(2002.9.27)
(31) 優先権主張番号	08/456,746
(32) 優先日	平成7年6月1日(1995.6.1)
(33) 優先権主張国	米国(US)

(73) 特許権者	富士通株式会社 神奈川県川崎市中原区上小田中4丁目1番 1号
(74) 代理人	弁理士 石田 敏
(74) 代理人	弁理士 土屋 繁
(74) 代理人	弁理士 戸田 利雄
(74) 代理人	弁理士 西山 雅也

最終頁に続く

(54) 【発明の名称】プログラマブル命令トラップシステム及び方法

(57) 【特許請求の範囲】

【請求項 1】

複数のコンピュータ命令を受理すべく配置され、前記命令を検索する命令フェッチャユニットと、

前記フェッチャユニットに結合され、前記コンピュータ命令の各々と複数のプログラミング可能な値のうちの1つとを比較し、そして前記比較によって前記各コンピュータ命令につきその動作属性を含む実行すべき機能を示す前記各命令に対する命令トラップ制御信号を生成するプログラマブル命令トラップユニットと、を有し、前記プログラマブル命令トラップユニットは、

複数の第1のプログラマブルレジスタであって、その各々が、ユーザーがトラップすることを望む1つ又は複数のトラップ命令を表し、かつユーザーがプログラミングできる第1の値を自動的に受理して記憶し、

複数の第1の比較器であって、その各々が、前記コンピュータ命令の1つと前記プログラマブルレジスタ値の1つとを受理するように配置され、前記コンピュータ命令が前記プログラマブルレジスタ値と一致するか否かを判断し、

複数のプログラマブル制御レジスタであって、その各々が、前記第1のプログラマブルレジスタのうちの1つと関連し、前記コンピュータ命令が前記プログラマブルレジスタ値と一致すると前記第1の比較器が判断した場合に前記コンピュータ命令の動作属性を含む実行すべき機能を表す第2の値を記憶し、

複数の制御ユニットであって、その各々が、前記プログラマブル制御レジスタと結合され

10

20

、前記第1のプログラマブルレジスタの1つが前記コンピュータ命令の1つと一致する場合に、前記第2の値を表す複数の信号を生成する、
上記構成を含むことを特徴とするマイクロプロセッサ。

【請求項2】

前記第1の比較器の各々は、

前記コンピュータ命令を受理するように配置された第2の比較器と、

前記制御ユニットに結合され、前記第2の比較器の出力を受理するように配置された第3の比較器と、

を含む請求項1に記載のマイクロプロセッサ。

【請求項3】

10

前記第1のプログラマブルレジスタの各々は、

前記第2の比較器に結合され、前記トラップ命令のうちの1つ又は複数を表わすデータを記憶する値レジスタと、

前記第3の比較器に結合され、前記値レジスタと関連するマスク情報を記憶するマスクレジスタと、を含み、

前記マスク情報の各ビットは、マスクビットインジケータ及びドントマスクビットインジケータの1つを表わす請求項2に記載のマイクロプロセッサ。

【請求項4】

前記第2の比較器は、前記コンピュータ命令の各ビットと前記値レジスタの関連するビットとを比較し、第2の比較器の結果を表わす第1のビット値を生成する請求項3に記載のマイクロプロセッサ。

20

【請求項5】

前記第3の比較器は、各第1のビット値出力と前記マスクレジスタからの関連するビットとを比較し、第3の比較器の結果を表わす第2のビット値を生成する請求項4に記載のマイクロプロセッサ。

【請求項6】

前記第3の比較器は、前記第2の比較器により出力された前記第1のビット値の全てと前記マスクレジスタの全ての関連するビットとを比較し、全てのビット比較が一致した場合に一致信号を生成する請求項4に記載のマイクロプロセッサ。

【請求項7】

30

4つの命令は、前記第1のプログラマブルレジスタの全てと同時に比較される請求項1に記載のマイクロプロセッサ。

【請求項8】

複数の第1のレジスタ、複数の第2のレジスタ、及び複数の制御レジスタを含むマイクロプロセッサにおいて、

前記複数の第1のレジスタ、前記複数の第2のレジスタ、及び前記複数の制御レジスタの全てのビットをプログラミングすること、

複数のビットをもつ複数の命令を受理すること、

各命令の各ビットと前記各第1のレジスタの関連ビットとの第1の比較を実行すること、
前記第1の比較の各々に対して第1のビット値を生成すること、

前記第1のビット値の各々と前記第1のビット値と関連する前記第2のレジスタの関連ビットとの第2の比較を実行すること、

前記命令が前記第2の比較を満足した場合に、前記命令の動作属性を含む関連する制御レジスタの内容を表わす信号を生成すること、

から成ることを特徴とする、前記マイクロプロセッサにおけるコンピュータプログラム命令の識別方法。

40

【請求項9】

前記複数の第1のレジスタの各々は、第1のコンピュータプログラム命令表現を記憶する請求項8に記載の方法。

【請求項10】

50

前記複数の第2のレジスタの各々は、前記第2の比較と関連する前記命令のビットを識別するためのマスクビットを記憶する請求項8に記載の方法。

【請求項11】

前記複数の制御レジスタの各々は、前記命令が前記第2の比較と一致した場合に取るべきアクションに対応する代わりのコンピュータプログラムを記憶した、メモリ装置のロケーションを識別する請求項8に記載の方法。

【請求項12】

前記複数の第1のレジスタ、前記複数の第2のレジスタ、及び前記複数の制御レジスタの全てのビットをプログラミングする前記ステップは、前記第1のレジスタ、前記第2のレジスタ、及び前記制御レジスタの全ての値をスキャンインするステップを含む請求項8に記載の方法。

10

【請求項13】

前記スキャニングステップは、2進値をシーケンシャルに受理し、その値を前記第1のレジスタ、第2のレジスタ、及び制御レジスタに記憶するステップを含む請求項12に記載の方法。

【請求項14】

複数の第1のレジスタと、

複数の第2のレジスタと、

複数の制御レジスタと、

前記複数の第1のレジスタ、前記複数の第2のレジスタ、及び前記複数の制御レジスタにおける全てのビットをプログラミングするプログラミング手段と、

20

複数のビットを有する複数の命令を受理する命令受理手段と、

前記第1のレジスタ及び前記命令受理手段に結合され、各々の命令の各ビットを前記第1のレジスタの各々の関連ビットと第1の比較を行う第1の比較手段と、

前記第1の比較手段に結合され、前記第1の比較の各々に対する第1のビット値を生成する第1の生成手段と、

前記第1の生成手段に結合され、前記第1のビット値の各々と、前記第1のビット値と関連する前記第2のレジスタの関連ビットとの、第2の比較を行う第2の比較手段と、

前記第2の比較手段に結合され、前記命令が前記第2の比較を満足する場合に、前記命令の動作属性を含む関連する制御レジスタの内容を表す信号を生成する第2の生成手段と、

30

で構成することを特徴とするコンピュータプログラム命令を識別するマイクロプロセッサ。

【請求項15】

前記複数の第1のレジスタの各々は、第1のコンピュータプログラム命令表現を記憶する請求項14に記載のマイクロプロセッサ。

【請求項16】

前記複数の第2のレジスタの各々は、前記第2の比較と関連する前記命令のビットを識別するためのマスクビットを記憶する請求項14に記載のマイクロプロセッサ。

【請求項17】

前記複数の制御レジスタの各々は、前記命令が前記第2の比較と一致した場合に取るべきアクションに対応する代わりのコンピュータプログラムを記憶した、メモリ装置のロケーションを識別する請求項14に記載のマイクロプロセッサ。

40

【請求項18】

前記複数の第1のレジスタ、前記複数の第2のレジスタ、及び前記複数の制御レジスタの全てのビットをプログラミングする前記ステップは、前記第1のレジスタ、前記第2のレジスタ、及び前記制御レジスタの全ての値をスキャンインするステップを含む請求項14に記載のマイクロプロセッサ。

【請求項19】

前記スキャニングステップは、2進値をシーケンシャルに受理し、その値を前記第1のレジスタ、第2のレジスタ、及び制御レジスタに記憶するステップを含む請求項18に記載の

50

マイクロプロセッサ。

【請求項 20】

さらに、前記プログラマブル命令トラップユニットに結合され、前記命令トラップ制御信号を受
理しかつ非同期に前記機能を実行する発行ユニットを含む請求項 1 に記載のマイク
ロプロセッサ。

【請求項 21】

複数のコンピュータ命令を受理するように配置され、前記複数のコンピュータ命令を受
理する命令フェッチユニットと、

前記フェッチユニットに結合され、前記コンピュータ命令の各々と複数のプログラマブル
値の 1 つとを比較し、前記比較により、前記コンピュータ命令に代えて実行すべきエミュー
レーションルーチンを表す前記命令の各々に対する命令トラップ制御信号を生成するプロ
グラマブル命令トラップユニットと、を有し、

前記プログラマブル命令トラップユニットは、

複数の第 1 のプログラマブルレジスタであって、その各々が、ユーザーがトラップするこ
とを望む 1 つ又は複数のトラップ命令を表し、かつユーザーがプログラミングできる第 1
の値を記憶し、

複数の第 1 の比較器であって、その各々が、前記コンピュータ命令の 1 つと前記プログラ
マブルレジスタ値の 1 つとを受理するように配置され、前記コンピュータ命令が前記プログラ
マブルレジスタ値と一致するか否かを判断し、

複数のプログラマブル制御レジスタであって、その各々が、前記第 1 のプログラマブルレ
ジスタのうちの 1 つと関連し、前記コンピュータ命令が前記プログラマブルレジスタ値と
一致すると前記第 1 の比較器が判断した場合に実行すべきトラップ同期機能を表す第 2 の
値を記憶し、

複数の制御ユニットであって、その各々が、前記プログラマブル制御レジスタと結合され
、前記第 1 のプログラマブルレジスタの 1 つが前記コンピュータ命令の 1 つと一致する場
合に、前記第 2 の値を表す複数の信号を生成する、

上記構成を含むことを特徴とする、非同期にコンピュータ命令を実行可能なコンピュータ
処理システム。

【請求項 22】

前記第 1 の比較器の各々は、

前記コンピュータ命令を受理するように配置された第 2 の比較器と、

前記制御ユニットに結合され、前記第 2 の比較器の出力を受理するように配置された第 3
の比較器と、

を含む請求項 21 に記載のコンピュータ処理システム。

【請求項 23】

前記第 1 のプログラマブルレジスタの各々は、

前記第 2 の比較器に結合され、前記トラップ命令のうちの 1 つ又は複数を表わすデータを
記憶する値レジスタと、

前記第 3 の比較器に結合され、前記値レジスタと関連するマスク情報を記憶するマスクレ
ジスタと、を含み、

前記マスク情報の各ビットは、マスクビットインジケータ及びドントマスクビットインジ
ケータの 1 つを表わす請求項 21 に記載のコンピュータ処理システム。

【請求項 24】

前記第 2 の比較器は、前記コンピュータ命令の各ビットと前記値レジスタの関連するビッ
トとを比較し、第 2 の比較器の結果を表わす第 1 のビット値を生成する請求項 23 に記載の
コンピュータ処理システム。

【請求項 25】

前記第 3 の比較器は、各第 1 のビット値出力と前記マスクレジスタからの関連するビット
とを比較し、第 3 の比較器の結果を表わす第 2 のビット値を生成する請求項 24 に記載のコ
ンピュータ処理システム。

10

20

30

40

50

【請求項 26】

前記第3の比較器は、前記第2の比較器により出力された前記第1のビット値の全てと前記マスクレジスタの全ての関連するビットとを比較し、全てのビット比較が一致した場合に一致信号を生成する請求項25に記載のコンピュータ処理システム。

【請求項 27】

4つの命令は、前記第1のプログラマブルレジスタの全てと同時に比較される請求項26に記載のコンピュータ処理システム。

【発明の詳細な説明】**関連出願の相互参照**

本願発明の主題は、下記に掲げる出願の主題と関連している。

出願番号特願平8-536744、「並列データプロセッサにおけるアクティブ命令を回転させる方法および装置」の名称で、Sunil Savkar, Michael C. Shebanow, Gene W. ShenおよびFar nad Sajjadianによって1995年6月1日に出願、

出願番号08/388,602、「スーパースケーラ マイクロプロセッサ用命令フロー制御回路」の名称で、Takeshi Kitaharaによって1995年2月14日に出願、

出願番号08/388,389、「格納命令に関して負荷命令を順不同に実行するアドレス方法」の名称で、Michael A. SimoneおよびMichael C. Shebanowによって1995年2月14日に出願、

出願番号08/388,606、「名前を付け替えられたレジスタに結果を効率的に書き込む方法及び装置」の名称で、DeForest W. Tovey, Michael C. ShebanowおよびJohn Gmuenderによって1995年2月14日に出願、

出願番号08/388,364、「マイクロプロセッサにおける物理レジスタの利用を調整する方法および装置」の名称で、DeForest W. Tovey, Michael C. ShebanowおよびJohn Gmuenderによって1995年2月14日に出願、

出願番号特願平8-525085、「精密な状態を保持するため命令状態をトラッキングするプロセッサ構造および方法」の名称で、Gene W. Shen, John Szeto, Niteen A. PatkarおよびMichael C. Shebanowによって1995年2月14日に出願、

出願番号特願平8-526887、「アドレス変換の高速化のための並列アクセス マイクロ-TLB」の名称で、Chih-Wei David Chang, Kioumars Dawallu, Joel F. Boney, Ming-Ying LiおよびJen-Hong Charles Chenによって1995年3月3日に出願、

出願番号特願平8-526888、「コンピュータシステムにおけるアドレス変換用ルックアサイドバッファ」の名称で、Leon Kuo-Liang Peng, Yolin LinおよびChin-Wei David Changによって1995年3月3日に出願、

出願番号08/397,893、「データプロセッサにおけるプロセッサ資源の再生利用」の名称で、Michael C. Shebanow, Gene W. Shen, Ravi Swami, Niteen A. Patkarによって1995年3月3日に出願、

出願番号08/397,891、「実行準備ができたものから命令を選択する方法および装置」の名称で、Michael C. Shebanow, John Gmuender, Michael A. Simone, John R.F.S. Szeto, Takumi MaruyamaおよびDeForest W. Toveyによって1995年3月3日に出願、

出願番号08/397,911、「不履行命令の高速ソフトウェア エミュレーション用ハードウェアサポート」の名称で、Shalesh Thusoo, Farnad Sajjadian, Jaspal KohliおよびNiteen A. Patkarによって1995年3月3日に出願、

出願番号08/398,284、「制御転送リターンを加速する方法および装置」の名称で、Akira Katsuno, Sunil SavkarおよびMichael C. Shebanowによって1995年3月3日に出願、

出願番号08/398,066、「フェッチプログラムカウンタの更新方法」の名称で、Akira Kats uno, Niteen A. Patkar, Sunil SavkarおよびMichael C. Shebanowによって1995年3月3日に出願、

出願番号08/397,910、「コンピュータシステムにおけるエラーの優先化および処理方法および装置」の名称で、Chin-Wei David Chang, Joel Fredrick BoneyおよびJaspal Kohliによって1995年3月3日に出願、

出願番号08/398,151、「制御転送命令の迅速な実行方法および装置」の名称で、Sunil W. 50

Savkarによって1995年3月3日に出願、

出願番号08/397,800、「マイクロプロセッサにおけるゼロビット状態フラッグの生成方法および装置」の名称で、Michael Simoneによって1995年3月3日に出願、

出願番号08/397,912、「パイプライン化読取り - 修正 - 書込みアクセスを備えたECC保護メモリ編成」の名称で、Chien ChenおよびYuzhi Luによって1995年3月3日に出願および、

出願番号08/398,299、「精密な状態を保持するため命令状態をトラッキングするプロセッサ構造および方法」の名称で、Chien Chen, John R.F.S.Szeto, Niteen A.Patkar, Michael C.Shebanow, Hideki Osone, Takumi MaruyamaおよびMichael A.Simoneによって1995年3月3日に出願、

10

参考として、上記の出願の全てを本願発明の全体に亘って取り入れている。

1. 発明の分野

本発明は、一般にデータプロセッサに関する。特に本発明は、マイクロプロセッサ内でプログラミング可能なハードウェア命令トラップを提供するシステム及び方法に関する。

2. 背景技術の説明

データプロセッサを設計する場合には、テープアウト (tape-out) の前に設計を厳しくテストすることが重要である。一般に、このようなテストは、データプロセッサの機能特性中の誤りを識別しようとするものである。しかしながら、時間的制約のためにテープアウト前にデータプロセッサ設計の全ての機能特性を充分にテストできるとは限らない。テープアウトは、例えばデータプロセッサ等の部品を製造する物理データベース設計が完了した時点で生じる。

20

場合によっては、データプロセッサの設計における機能誤りが、テープアウト後まで発見されないこともある。このような状況では、設計を修正し、新しいデータプロセッサ部品を生産するのに多大な費用がかかる。機能誤りは、或る命令に影響を与える製造上の欠陥によっても生じる可能性がある。

関連する制限内容は、スーパースカラ中央処理ユニット (CPU) についても生じる。多くのソフトウェアアプリケーションプログラムは、スーパースカラ環境で使用されるようには開発されていない。スーパースカラCPUがそのようなアプリケーションプログラムにおける命令を実行した場合、いくつかの命令は設計通りに実行されない可能性がある。

必要なことは、CPU内に複数の命令をその実行前にトラップできるようにするプログラミング可能なハードウェアデバイスを与えるシステム及び方法である。トラップされるべき命令は、CPUのデバッギング中の融通性を提供し、CPUがさまざまなアプリケーションプログラムを適切に実行できるようにプログラミング可能である。このシステムは、またトラップされた命令をエミュレートし及び／又は順次実行することを可能にする手段を提供しなければならない。

30

発明の要約

本発明は、CPU内でプログラミング可能なハードウェアデバイスを提供するシステム及び方法に関する。プログラミング可能なハードウェアデバイスは、実行前に複数の命令をトラップ可能にする。トラップされるべき命令は、CPUデバッギングの間の融通性を提供し、又CPUがさまざまなアプリケーションプログラムを適切に実行し得るようにプログラミング可能である。システムは、またトラップされた命令をエミュレートし及び／又は順次実行させる手段も提供しなければならない。

40

【図面の簡単な説明】

図1は、本発明の好適な実施例が存在するコンピュータ環境の例示である。

図2は、本発明の好適な実施例のスーパースカラ中央処理ユニットのより詳細な例示である。

図3は、本発明の好適な実施例によるプログラミング可能な命令トラップユニットのより詳細な例示である。

図4は、本発明の好適な実施例によるプログラミング可能な命令トラップユニットのより詳細な例示である。

50

図5は、本発明の好適な実施例による命令トラップレジスタ値(ITRV)及び命令トラップレジスタマスク(ITRM)のさらに詳し例示である。

図6は、本発明の好適な実施例によるビットレジスタのより詳細な例示である。

図7は、本発明の好適な実施例による命令をトラップする方法を記述したフロー図である。

図8は、本発明の好適な実施例による命令をテストする方法を記述したフロー図である。
好適な実施例の詳細な説明

本発明の好適な実施例について図面を参照しながら説明する。ここで、同じ引用符号は同一の又は機能的に類似した構成要素を表わしている。また、図面中、各引用符号の最も左側の数字は、その引用符号が最初に用いられた図面と対応している。

図1は、本発明の好適な実施例が与えられたコンピュータ環境の例示である。従来のメモリーユニット104及び従来の入出力(I/O)デバイス106はデータプロセッサ102と通信する。好適な実施例において、プロセッサは、HaL Computer Systems, Campbell, Californiaにより開発されたHaL R1プロセッサである。HaL R1プロセッサは、SPARC International, Inc., Menlo Park, Californiaによって開発されたSPARC V9 64ビットの命令セットアーキテクチャを具備する。プロセッサ102は、スーパースカラCPU108、メモリー管理ユニット(MMU)、及び4つのキャッシュメモリユニット110, 112を含む。

キャッシュメモリユニット110, 112は、好適には2つの64キロバイト(KB)従来型データキャッシュと2つの64KB従来型命令キャッシュを内含している。データキャッシュ110は、スーパースカラCPU108及びMMU114にデータを与える。キャッシュメモリユニット110, 112の両セットは、仮想インデックスとタグ付けされている。各キャッシュチップ110, 112は、4セットとして構成された64KBのデータ記憶装置を含む。各64KBのキャッシュチップ110, 112は、スーパースカラCPU108からの独立した2つの要求をサービス可能である。スーパースカラCPU108とキャッシュ110, 112の間のインターフェイスはブロッキングされず、キャッシュラインがリファイル(refill)又はフラッシュ(flash)されると同時にスーパースカラCPU108はキャッシュ110, 112にアクセスできる。キャッシュメモリユニット110, 112に関し、さらなる詳細は、本文にその全体が参考として内含されるChen, Chien, Y. Lu, A. Wongの「 HaL Cacheのマイクロアーキテクチャ 」 Compcon議事録、1995年、の中に記載されている。

MMU114の機能は、プロセッサ102のために、メモリ管理及びデータの一貫性(coherence)を制御すること、メモリ104及びI/Oデバイス106とインターフェースすること、そしてエラー操作を含む。MMU114は、次の3つのレベルのアドレス空間を内含する：すなわち、(1)プロセッサ102のための仮想アドレス空間、(2)I/Oデバイス106及び診断プロセッサのための論理アドレス空間、そして(3)メモリ104のための物理アドレス空間である。これらの階層化されたアドレス空間は、大きなアドレス空間を管理するのに有効なメカニズムを提供する。好適な実施例において使用されるMMU114のより詳細な説明は、本文にその全体が参考として内含されるDavid Chang, D. Lyon, C. Chen, L. Peng, M. Massouni, M. Hakimi, S. Iyengar, E. Li, R. Remedios「 HaL のメモリ管理ユニットのマイクロアーキテクチャ 」、Compcon Proceedings 1995の中に記載されている。

図2は、本発明の好適な実施例のスーパースカラCPU108のさらに詳細な例示である。プロセッサ102は、固定小数点命令のための4段のパイプライン、及び「 LOAD 」命令のための6段のパイプラインを備える。スーパースカラCPU108は、命令フェッチユニット202、プログラマブル命令トラップユニット204、命令発行ユニット206、命令実行ユニット218、精細状態ユニット220、分岐履歴テーブル208、復帰予測テーブル210、プリフェッチバッファ212、記録済み命令キャッシュ214、そして分岐ユニット216を含む。

フェッチユニット202は、図2にデータライン202として示されるオフチップ(off - chip)64KB命令キャッシュ112、プリフェッチバッファ212、又は記録済み命令キャッシュ214のいずれかにリクエストしてそれから4つの命令を受理する。4つの命令は、次にどの命令をトラップすべきか決定するため、プログラマブル命令トラップユニット204に送られる。前記プログラマブル命令トラップユニット204については、以下図3 - 8を参照して

10

20

30

40

50

詳細に説明する。分岐履歴テーブル208は、実行分岐の方向予測のために用いられる複数のカウンタを含む。復帰予測テーブル210は、実行された命令の復帰アドレスを予測するのに用いられる。分岐ユニット216は、発行ユニット206、分岐履歴テーブル208、及び復帰予測テーブル210からの情報を使用して制御転送命令が現行の発行ウインドウ内にあるか否かを決定し、さらに制御転送命令が用いられた場合にはプログラムカウンタを修正する。発行ユニット206は、利用可能なコンピュータ資源及び例えば1つの命令を同期化すべき等の、発行制約条件を決定する。全ての命令は、従来のシリアルプロセッサ内で発行される順序で発行ユニット206により発行され、そして命令実行ユニット218へ送られる。命令実行ユニット218は命令を実行するが、必ずしも前記発行された順序である必要はない。命令実行ユニット218は、命令の実行結果をデータバス222上に出力する。好適な実施例のスーパースカラCPU108についてさらに詳細な説明は、Niteen Patkar,A.Katsuno,S.Li,T.Maruyama,S.Savkar,M.Simone,G.Shen,R.Swami,D.Tovey,HaL PM1 CPUのマイクロアーキテクチャ、Compcon議事録1995、に記述されている。本発明は、ここで述べるプログラマブル命令トラップユニット204の中に存在する。

図3は、本発明の好適な実施例によるプログラマブル命令トラップユニット204のより詳細な例示である。プログラマブル命令トラップユニット204は、好適にはフェッチユニット202から4つの命令を受信する。命令は、図3にINST00,INST01,INST10,INST11として示されている。各命令は、命令トラップ論理ユニット(ITLU) 302へ送られる。命令トラップ論理ユニットは、また好適には4つの命令トラップレジスタ(ITRV) 304、4つの命令トラップレジスタマスク(ITRM) 306、及び4つの制御レジスタ(CR) 308からの信号を受信する。命令トラップ論理ユニット302、ITRM304、ITRM306、及びCR308については、以下図4-8を参照して詳細に説明する。

図4は、本発明の好適な実施例によるプログラマブル命令トラップユニットのより詳細な例示である。命令トラップ論理ユニット302は、好適には4つの排他的NOR論理デバイス404A-Dを含み、ここで各排他的NORはそれぞれのITRV304と結合する。各排他的NORデバイス404は、命令の各反転ビットと、関連するITRV、例えばITRV0410、からの対応するビットとを比較し、各々のビットについて、一致するか否かを表わす値を出力する。命令INST00の各ビットをITRV304中の各ビットと比較するのに他の技術も使用可能と考えられる。この比較を行なう方法については、以下図7-8を参照してより詳細に説明する。

命令トラップ論理ユニット302は、好適には関連する排他的NORにより出力された各ビットと、ITRM306のビット値とを比較する4つのデバイス406A-Dを含む。各ITRM306は、「ケア(care)」又は「ドントケア(don't care)」値を含む複数のビットを有する。4つのデバイス406A-Dの各々は、関連する排他的NORデバイスによって出力されたビットと、それと関連するITRM306中のビットとを比較する。各デバイス406は、命令ビットの全てが関連するITRV-ITRMレジスタ対と一致するか否かを表わす1ビットの値を出力する。好適には、各デバイス406は、A 032×2デバイスである。このデバイスは、2セットの32ビット入力を有し、1つのセットは、関連する排他的NOR404出力の各ビットラインに結合され、もう1つのセットは関連するITRM306に結合される。A 032×2は、32の論理ANDゲートを含み、それらの出力は論理ORゲートに入力される。この比較を実施する方法については、図7-8により以下さらに詳細に説明する。

CR308は、一つのITRV-ITRM対と関連する。例えば、CROはITRV0及びITRM0と関連する。各CRは、命令がその関連するITRV-ITRMレジスタ対と一致した場合にとるべきアクションを表わすデータを含む。各制御レジスタは、任意の数のビットを含むことができるが、好適な実施例では各CR308は2つのビットを含む。1つのビットは同期動作を表わし、もう1つのビットは命令トラップを表わす。すなわち、制御レジスタのビットが「00」であり、命令が関連するITRV-ITRMレジスタ対304,306と一致する場合には、特別なアクションは全くとられない。制御レジスタ内のビットが「01」であり、命令が関連するITRV-ITRMレジスタ対304,306と一致する場合、プログラマブル命令トラップユニットは、一致命令を同期的に、すなわち、一致命令より前に発行した全ての命令が実行を完了し、マシンは既知の状態、すなわち一致命令を実行する前の状態、となるよう、実行させる信号を出力す

10

20

30

40

50

る。さらに、同期命令が発行され、実行され、完了され、そして次の命令が発行される前の状態となる。制御レジスタのビットが「10」であり、命令が関連するITRV - ITRMレジスタ対304,306と一致する場合、プログラマブル命令トラップユニット204は、非同期命令発行トラップを生成する。すなわち、前記、命令は命令トラップのセマンティクスを引き継ぐ。制御レジスタのビットが「11」であり、命令が、付随するITRV - ITRMレジスタ対304,306と一致する場合、プログラマブル命令トラップユニット204は同期命令発行トラップを生成する。すなわち命令はトラップされ同期して実行されることになる。

命令トラップ論理ユニット302は、例えばINST00といった受信した命令と一致するITRV - ITRMレジスタ対304,306と関連するCR308に記憶された値を生成するためのデバイスを有する。この値を生成するためには、さまざまな技術を利用することができる。好適には、4つの論理ORゲート（各々はそれぞれのCR308と関連する）の2つのセットが、CR308から入力を受理する。第1のORゲートセットの各ORゲートは、（1）例えばCR0といったCR308の第1のビット及び（2）前記CR308と関連するA_{032×2}からの1ビットの出力、という2つの入力を受理する。同様に、第2のORゲートセットの各ORゲートは、（1）例えばCR0といったCR308の第2ビット及び（2）前記CR308と関連するA_{032×2}からの1ビットの出力という2つの入力を受理する。各々のORゲートセットの全てのORゲート出力は、論理ANDゲート408A - Bに与えられる。命令トラップ論理ユニット302の出力を生成する方法は、以下図7 - 8を参照して詳細に説明する。

ITRV304, ITRM306及びCR308の値はプログラミング可能である。これらの値は、上述の通り、ユーザーが命令をトラップし、或る種の命令を同期的に実行させるようにプログラミングできるものである。本発明のプログラミング可能であるという特長は、命令をトラップし同期化する有効な技術を提供し、例えばITRV304, ITRM306に記憶された命令などのある命令を同期的に実行することによって、プロセッサ命令の機能誤りを正し、アプリケーションプログラムの適切なオペレーションを確保する。ITRV304, ITRM306、及びCR308（まとめてプログラマブルレジスタと呼ぶ）をプログラミングするには、さまざまな技術を使用することができる。例えば、プログラマブルレジスタはデータバスに結合され、そこからデータを受理できるようにし得る。好適な実施例では、コンピュータブートアップオペレーション中に、データは「スキャンイン」される。プログラマブルレジスタは、シフトレジスタと同様な技術を用いてプログラミングされる。

図5は、本発明の好適な実施例によるITRV304及びITRM306のより詳細な例示である。各プログラマブルレジスタ410は、複数のビットレジスタ502を有する。好適には、各ITRV304及び各ITRM306は27のビットレジスタ502を含み、一方各CR306は2つのビットレジスタ502を含む。図5は、ITRV304又はITRM306を示している。CR308は、その各々が好適には上述の通り27ビットのレジスタ502ではなく2ビットのレジスタ502を含む点を除いて、同じ仕方で動作する。各プログラマブルレジスタ410は、スキャンクロック(SC)信号、スキャン - イネーブル(SE)信号及びスキャン - 入力(SI)信号を受信し、スキャン - 出力(SO)信号及びQ出力信号を生成する。

図6は、本発明の好適な実施例によるビットレジスタ502のより詳細な例示である。ビット値を含む信号はSIライン上で受理される。SC及びSE信号が高レベルの場合、すなわち信号がバイナリの「1」値を表わす場合、論理ANDゲート602は高レベル信号を生成する。この信号は、トランジスタ604のゲートに結合される。その結果、トランジスタ604は導通し、それによってSIライン上の信号がトランジスタ604を通過し、その論理値は2つのインバータ604によって保持される。SE信号が高レベルの間にSCが低レベルになると、トランジスタ604はオフされ、論理ANDゲート608は高レベル信号を生成する。前記論理ANDゲート608は、トランジスタ612のゲートに結合される。その結果として、トランジスタ612は導通し、反転SI信号は再びインバータ610の1つにより反転され、SOライン上に出力される。前記信号は、またビットレジスタがITRV304の一部である場合にはデバイス404Aに、ビットレジスタがITRM306の一部である場合にはインバータ及びデバイス406Aに、又はビットレジスタがCR308の一部である場合にはインバータ及び論理ORゲート、のいずれかに結合されるQライン上に出力される。SOラインは次のビットレジスタSIラインに結合される。

。ITRV304の最後のビットレジスタのS0ラインは、第1のITRM306の第1のビットレジスタに結合される。ITRM306の最後のビットレジスタは、第1のCR308の第1のビットレジスタに結合される。コンピュータ初期化プロセス中、プログラマブルレジスタ値を表わすデータは第1のSIライン412内に体系的に入力され、各々のプログラマブルレジスタが所望の値を得るまで全てのビットレジスタを通じてシフトされる、すなわちビットレジスタ値はスキヤンインされる。

ここで、本発明の方法について詳細に説明する。図7は、本発明の好適な実施例による命令をトラップする方法を示したフロー図である。まず最初に、ITRV304、ITRM306、及びCR308がプログラミングされる。上述の通り、これらのプログラマブルレジスタは、データバス及び制御信号を用いて個別にプログラミングできる(704, 706, 708)。好適には、プログラマブルレジスタはスキヤンイン、すなわち図5-6を参照して上述したビットシフティング技術を用いて、コンピュータ初期化の間にプログラミングされる。プログラマブルレジスタがプログラミングされた後(702)、命令がテストされる(710)。命令テスト方法は、図8を用いて以下で述べられる。

図8は、本発明の好適な実施例による命令のテスト方法を示したフロー図である。好適には、図2に示されているように、プログラマブル命令トラップユニット204によりフェッチャユニット202から4つの命令が受理される(802)。各命令は、個々の命令トラップ論理ユニット302に送られる。例えば、命令INST00は、命令トラップ論理ユニット302Aに送られる。命令トラップ論理ユニット302Aは、インバータ402を用いて命令内の各ビットを反転させる。反転された命令ビットは、図4で説明したように4つの排他的NORデバイス404A-Dにより受理される。各排他的NORデバイス404は、各反転された命令ビットを、先にプログラミングされたITRV304の1つの対応ビットと比較する(804)。例えば、排他的NORデバイス404Aは、図4-6に示すように命令ビット0とITRV0のビットレジスタ0とを比較する。反転された命令ビットがITRV304の1つの関連するビットと一致する場合には、排他的NORは、論理「1」を表わす信号を生成する。そうでなければ、排他のNORは、論理「0」を表わす信号を生成する。排他のNORデバイスは、関連するITRVビットレジスタ502と命令の反転ビットとを比較するため、排他のNORが論理「0」を表わす信号を生成した場合、ITRVビットは命令ビットと一致する。命令の各反転ビット、INST00、は、各ITRV304のそれぞれ関連するビットと比較される。好適には、これらの比較結果は、デバイス404A-Dの排他のNOR比較の結果を表わす4つの27ビット信号である。

好適には、27ビット信号の各セットにおける各ビットは、A 032×2デバイス406A-Dを用いて関連するITRM306のビットレジスタ502と比較される(806)。各A 032×2デバイス406A-Dは、少なくとも27の論理ANDゲートで構成され、その出力は論理ORゲートに入力される。例えば、デバイス404Aから出力される最下位ビット又はゼロビットは、ITRM0の反転ゼロビットレジスタに伴う論理ANDゲートに入力される。ITRM306はマスクである。時として、各命令を関連する命令パターンセットと比較することが望ましい。例えば、ビット5-9の或る組合せからなる全ての命令と一致させることが望ましい場合もある。この状況下では、残りのビット例えばビット0-4及び10-26は、その値がこの特定の一一致に対して不適切であることから、マスキングされるべきである。この一致は、適切なITRMビットレジスタ内に「ドントケア」値を配置することによって達成される。本発明においては、「ドントケア」値は論理「1」である。図4に示すように、ITRM値は、A 032×2によって受理される前に反転される。従って、命令ビットが関連するITRVビットレジスタと一致したことからデバイス404Aからのビットがゼロであるならば、A 032×2 406内の関連する論理ANDゲートの出力はゼロとなる。同様にして、デバイス404Aからのビットが論理1の場合、AND出力は、ITRM0内の関連するビットレジスタがマスクされたすなわちその反転値がゼロに等しい場合にのみ、ゼロとなる。

上述したように、全てのANDゲート出力は、ORゲートに結合される。いずれかのANDゲート出力が論理的1であるなら、ORゲートの出力は論理1である。ORゲートの出力は、A 032×2 406の出力である。従って、A 032×2 406は、命令INST00からのビットの全てが関連するITRVビットと一致するか又はマスキングされた場合のみ、すなわち命令INST00がITRV

10

20

30

40

50

- ITRM304,306レジスタ対と一致する場合にのみ、論理ゼロを表わす命令一致値(IMV)信号を生成する。

プログラマブル命令トラップユニット204は、何らかの一致が発生したか否かを決定し(8 10)、発生した場合には関連するCR308の値を生成する(812)。これを達成するプロセスについて述べる。各A 032×2デバイス406からの出力は、図4に示されるように2つの論理ORゲート、例えばORゲート418,420、によって受理される。第1のORゲート418に対する第2の入力は、関連するCR308の第1のビットレジスタの値である。第2のORゲート420に対する第2の入力は、関連するCR308の第2のビットレジスタの値である。例えば、命令INST00がレジスタ対ITRV0 - ITRM0のみと一致する場合、A 032×2 406Aはその出力で論理ゼロを表わす信号を生成し、残りのA 032×2 406B - Dは、その出力で論理1を表わす信号を生成する。A 032×2 406B - Dと関連する3つのORゲートは、その入力の1でバイナリ1を受理する。従って、これら3つのORゲートは、バイナリ1を表わす信号を出力する。NANDゲート408Aは、ORゲート418の出力が論理1を生成した場合にのみ論理ゼロ出力を生成する。デバイス406Aの出力は論理的ゼロであるため、ORゲート418の出力は、CR0の第1のビットレジスタが論理ゼロを含む場合にのみ論理1である。4つのORゲート及び1つのNANDゲート408の各セットは、上述のとおり、(1) NANDゲート408A - Bにより出力されるべき一致レジスタ対と関連するCR308の値、又は(2) 命令INST00がいかなるレジスタ対とも一致しない場合、NANDゲート408A - Bはいかなる特別なアクションもとるべきでないということを表わす「00」値、のいずれかを出力する。

受理した4つの命令、すなわちINST00,INST01,INST10、及びINST11、は全て、それぞれの4つのITRVレジスタ及びITRMレジスタと同様に比較される。 20

複数のレジスタ対が一つの命令と一致することも可能である。例えば、レジスタ対ITRV1 - ITRM1及びレジスタ対ITRV2 - ITRM2の両方が特定の命令、例えばINST00と一致することも可能である。このような状況下では、1つのCR、例えばCR2、のCRビットレジスタ内のビット値「1」により、関連するNANDゲートが論理1を表わす信号を生成する。

4つの命令すべてが一緒にテストされた後もこの方法は続行される(814)。より多くの命令が受理された場合、前記プロセスは反復する(712)。全ての命令がテストされた後、上述のとおりデータバスを使用することにより、又は好適にはコンピュータをリブートし、新しいプログラマブルレジスタ値におけるスキャンによって、ITRV304,ITRM306、及びCR308の新しい値をプログラミングすることができる(714)。その後、プロセス702 - 7 30 14が繰り返される。

発明は、好適な実施例及びそのいくつかの別の実施例を参考にして特定的に図示され記述されてきたが、当業者であれば、本発明の精神及び範囲から逸脱することなく形状及び細部のさまざまな変更をそれに加えることが可能である。

【図1】

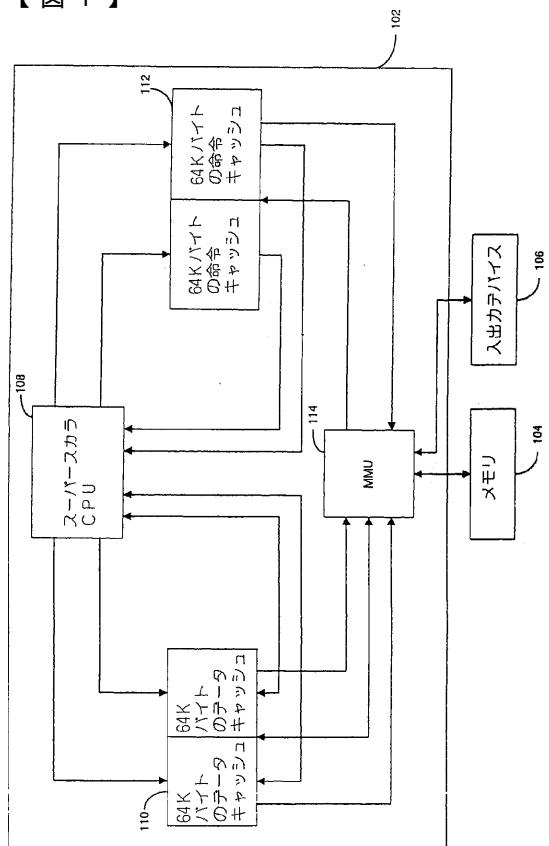


Figure 1

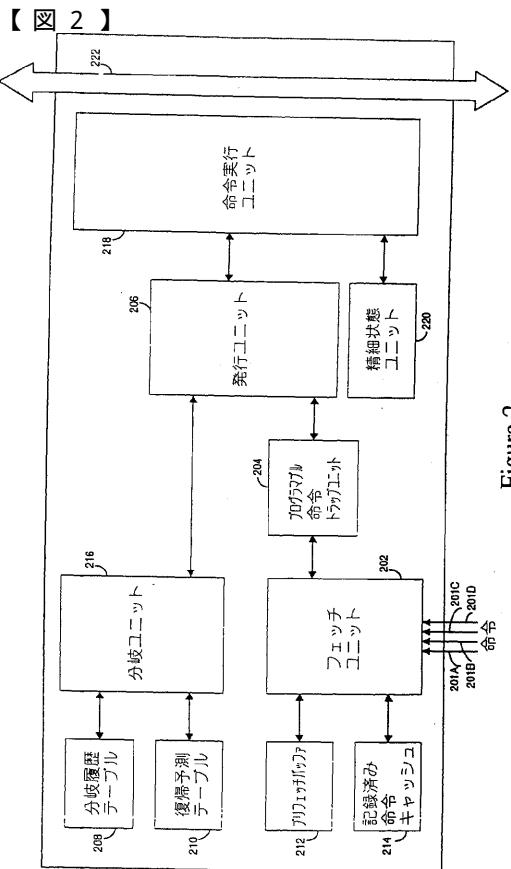


Figure 2

【図3】

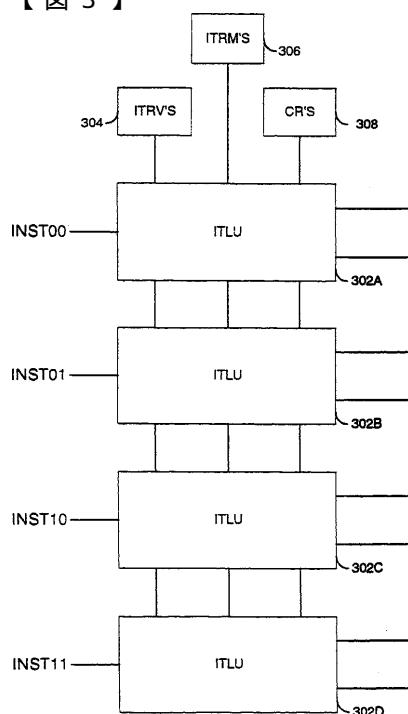


Figure 3

【図4】

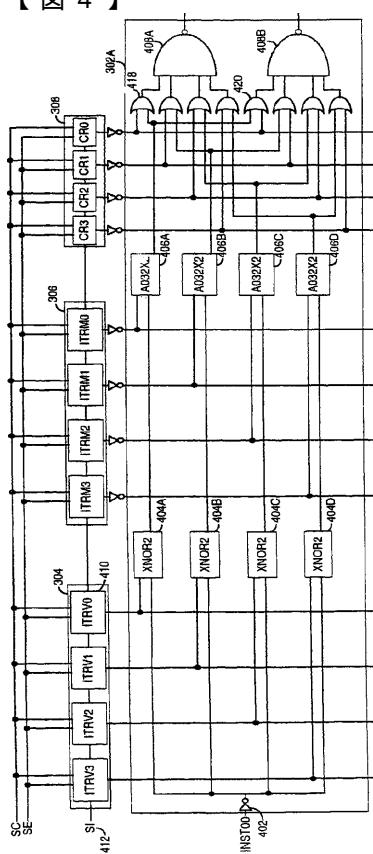


Figure 4

【図5】

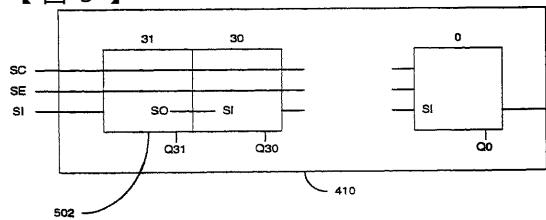


Figure 5

【図6】

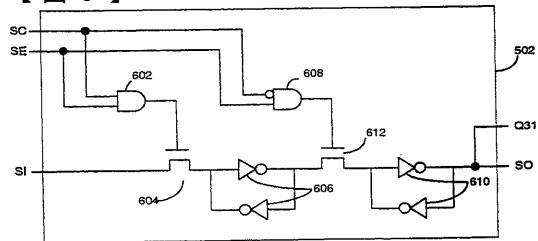


Figure 6

【図7】

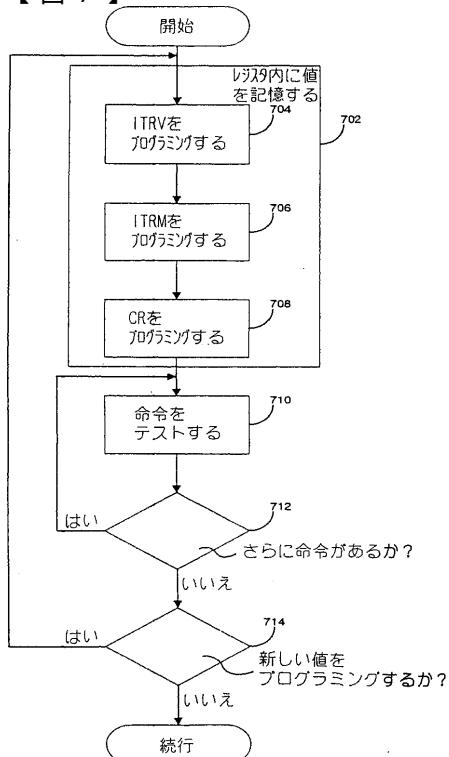


Figure 7

【図8】

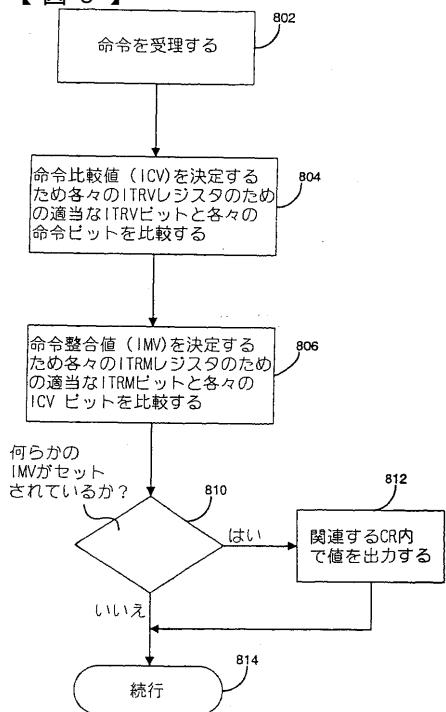


Figure 8

フロントページの続き

(72)発明者 サブカー , スニル

アメリカ合衆国 , ニューヨーク 10027 , ニューヨーク , 411 ウエスト ワンシックステ
ィーンス ストリート , キャンバス エイチ 103エー イースト

(72)発明者 シエン , ジェネ

アメリカ合衆国 , カリフォルニア 94043 , マウンテン ビュー , セントラル アベニュー 1
81エー

(72)発明者 サッジヤディアン , ファーナッド

アメリカ合衆国 , カリフォルニア 94087 , サニーベール , サウス カスケード テラス 5
24

(72)発明者 シエバナウ ,マイク

アメリカ合衆国 , テキサス 75075 , プラノ , グレンウィック ドライブ 1920

審査官 後藤 彰

(56)参考文献 特開平7 - 13807 (JP, A)

特開平4 - 367902 (JP, A)

特開昭63 - 240634 (JP, A)

(58)調査した分野(Int.Cl.⁷ , DB名)

G06F 9/38

G06F 11/28