

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2014-515532
(P2014-515532A)

(43) 公表日 平成26年6月30日(2014.6.30)

(51) Int. Cl.	F I	テーマコード (参考)
G06F 9/45 (2006.01)	G06F 9/44 322D	5B013
G06F 9/54 (2006.01)	G06F 9/06 640D	5B033
G06F 9/30 (2006.01)	G06F 9/30 350D	5B081
G06F 9/38 (2006.01)	G06F 9/38 370C	5B376

審査請求 有 予備審査請求 有 (全 39 頁)

(21) 出願番号 特願2014-513554 (P2014-513554)
 (86) (22) 出願日 平成24年5月21日 (2012.5.21)
 (85) 翻訳文提出日 平成25年11月27日 (2013.11.27)
 (86) 国際出願番号 PCT/US2012/038797
 (87) 国際公開番号 W02012/166410
 (87) 国際公開日 平成24年12月6日 (2012.12.6)
 (31) 優先権主張番号 61/491,075
 (32) 優先日 平成23年5月27日 (2011.5.27)
 (33) 優先権主張国 米国 (US)
 (31) 優先権主張番号 13/181,712
 (32) 優先日 平成23年7月13日 (2011.7.13)
 (33) 優先権主張国 米国 (US)

(71) 出願人 507364838
 クアルコム、インコーポレイテッド
 アメリカ合衆国 カリフォルニア 921
 21 サン ディエゴ モアハウス ドラ
 イヴ 5775
 (74) 代理人 100108453
 弁理士 村山 靖彦
 (74) 代理人 100163522
 弁理士 黒田 晋平
 (72) 発明者 ルイス・セゼ
 アメリカ合衆国・カリフォルニア・921
 21・サン・ディエゴ・モアハウス・ドラ
 イヴ・5775

最終頁に続く

(54) 【発明の名称】 動的言語のハッシュテーブルのためのハードウェアサポート

(57) 【要約】

態様は、コンピューティングデバイスが、従来ソフトウェアベースのJavaScript(登録商標)動作をハードウェアで実行することを可能にする。各JavaScript(登録商標)オブジェクトは、ソフトウェアに記憶され得るマスターハッシュテーブル内にハッシュされる。ソフトウェアハッシュテーブルの一部分は、ハッシュテーブル処理に専用の特殊な命令セットレジスタを使用して、ハードウェアハッシュテーブルにプッシュされ得る。ソフトウェアプロセスがハッシュテーブル動作(たとえば、ルックアップ)を要求するたびに、ハードウェアハッシュテーブルは、値がハードウェア内に存在するかどうかを判断するために調べられる。要求された値がハードウェアハッシュテーブル内にある場合、要求された値は単一の動作ステップでアクセスされる。要求された値がハードウェアハッシュテーブル内にない場合、要求された値は、ソフトウェア内のマスターハッシュテーブルから取り出され、取り出された値を含有するマスターハッシュテーブルの一部分が、特殊な命令セットレジスタを使用してハードウェアにプッシュされる。

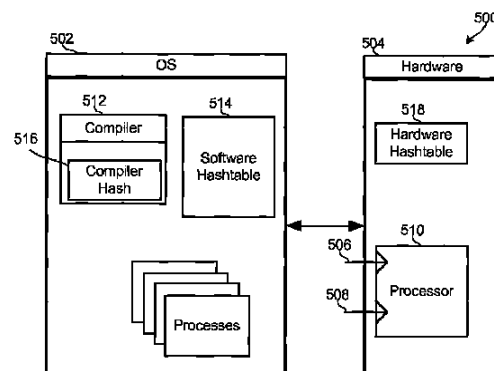


FIG. 5A

【特許請求の範囲】**【請求項 1】**

動的ソフトウェア言語コードを実行するための方法であって、
ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップと、
前記符号化されたハッシュテーブル整数キーを、ソフトウェアによってアクセス可能なメモリに記憶するステップと、
前記符号化されたハッシュテーブル整数キーの一部を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップと
を含む、方法。

10

【請求項 2】

前記第1の専用のハードウェア命令が、前記専用のハードウェアハッシュテーブルインプリメンテーションに値をプッシュするSET命令である、請求項1に記載の方法。

【請求項 3】

ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、ハードウェアプロセッサ上で専用のハードウェア命令を実行することによって、ハッシュテーブル値が前記ハードウェアプロセッサによって取り出され得るかどうかを判断するステップと、

前記キーが前記ハードウェアプロセッサに記憶されている場合、対応する前記ハッシュテーブル整数キーを返すためにハードウェアベースのハッシュテーブルルックアップを実行するステップと

20

をさらに含む、請求項2に記載の方法。

【請求項 4】

前記キーが前記ハードウェアプロセッサ上にない場合、ソフトウェアアクセス可能なメモリに記憶されている、要求された前記ハッシュテーブル値と前記符号化されたハッシュテーブル整数キーの一部とを取り出すために、ソフトウェアベースのハッシュテーブルルックアップを実行するステップと、

符号化されたハッシュテーブル整数キーの前記取り出された部分を、第2の専用のハードウェア命令を介して前記ハードウェアプロセッサの専用メモリにロードすることによって、前記専用のハードウェアハッシュテーブルインプリメンテーションを更新するステップと

30

をさらに含む、請求項3に記載の方法。

【請求項 5】

前記第2の専用のハードウェア命令が、前記ハードウェアプロセッサの前記専用メモリから値を取り出すGET命令である、請求項4に記載の方法。

【請求項 6】

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップがコンパイル時に厳密に一度だけ遂行され、前記符号化されたハッシュテーブル整数キーの一部をロードするステップが、動的ソフトウェア言語動作の要求ごとに実行される、請求項5に記載の方法。

40

【請求項 7】

前記動的ソフトウェア言語コードがJavaScript(登録商標)コードであり、
ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップが、JavaScriptハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するようにJavaScriptコードをコンパイルするステップを含む、請求項5に記載の方法。

【請求項 8】

JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップがコンパイル時に厳密に一度だけ遂行され、前記符号化されたハ

50

ッシュテーブル整数キーの一部をロードするステップが、JavaScript(登録商標)動作の要求ごとに行われる、請求項7に記載の方法。

【請求項9】

前記第1の専用のハードウェア命令が、メモリマップされたロケーションに値を記憶することによって、前記専用のハードウェアハッシュテーブルインプリメンテーションに前記値をプッシュするSTORE命令である、請求項1に記載の方法。

【請求項10】

前記符号化されたハッシュテーブル整数キーの一部を専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップが、前記符号化されたハッシュテーブル整数キーの一部を、1次プロセッサに結合されたコプロセッサ上の専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップを含む、請求項1に記載の方法。

10

【請求項11】

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップがコンピューティングデバイスの第1のプロセッサによって実施され、前記符号化されたハッシュテーブル整数キーの一部を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップが、前記コンピューティングデバイスの第2のプロセッサによって実施される、請求項1に記載の方法。

【請求項12】

前記コンピューティングデバイスの前記第2のプロセッサが専門のハッシュテーブルコプロセッサである、請求項11に記載の方法。

20

【請求項13】

ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、前記専門のハッシュテーブルコプロセッサ上で第2の専用のハードウェア命令を実行することによって、前記ハッシュテーブル値が取り出され得るかどうかを判断するステップと、

前記ハッシュテーブル値が取り出され得るものと判断される場合、前記コンピューティングデバイスの前記第1のプロセッサ上で対応する前記ハッシュテーブル整数キーを取り出すために、前記専門のハッシュテーブルコプロセッサ上でハードウェアベースのハッシュテーブルルックアップを実行するステップと

30

をさらに含む、請求項12に記載の方法。

【請求項14】

メモリと、

前記メモリに結合されたプロセッサであって、

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように動的ソフトウェア言語コードをコンパイルするステップと、

前記符号化されたハッシュテーブル整数キーがソフトウェアによってアクセス可能となるように、前記符号化されたハッシュテーブル整数キーを前記メモリに記憶するステップと、

前記符号化されたハッシュテーブル整数キーの一部を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップと

40

を含む動作を実施するためのプロセッサ実行可能命令で構成される、プロセッサとを備える、コンピューティングデバイス。

【請求項15】

前記メモリが前記プロセッサ内のハッシュテーブルメモリである、請求項14に記載のコンピューティングデバイス。

【請求項16】

前記メモリが前記プロセッサの外部にある、請求項14に記載のコンピューティングデバイス。

50

【請求項 17】

前記プロセッサが、前記第1の専用のハードウェア命令が前記専用のハードウェアハッシュテーブルインプリメンテーションに値をプッシュするSET命令であるような動作を実施するためのプロセッサ実行可能命令で構成される、請求項14に記載のコンピューティングデバイス。

【請求項 18】

前記プロセッサが、

ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、ハードウェアプロセッサ上で専用のハードウェア命令を実行することによって、ハッシュテーブル値が前記ハードウェアプロセッサによって取り出され得るかどうかを判断するステップと、

前記キーが前記ハードウェアプロセッサに記憶されている場合、対応する前記ハッシュテーブル整数キーを返すためにハードウェアベースのハッシュテーブルルックアップを実行するステップと

をさらに含む動作を実施するためのプロセッサ実行可能命令で構成される、請求項17に記載のコンピューティングデバイス。

【請求項 19】

前記プロセッサが、

前記キーが前記ハードウェアプロセッサ上にない場合、ソフトウェアアクセス可能なメモリに記憶されている、要求された前記ハッシュテーブル値と前記符号化されたハッシュテーブル整数キーの一部とを取り出すためにソフトウェアベースのハッシュテーブルルックアップを実行するステップと、

符号化されたハッシュテーブル整数キーの前記取り出された部分を、第2の専用のハードウェア命令を介して前記ハードウェアプロセッサの専用メモリにロードすることによって、前記専用のハードウェアハッシュテーブルインプリメンテーションを更新するステップと

をさらに含む動作を実施するためのプロセッサ実行可能命令で構成される、請求項18に記載のコンピューティングデバイス。

【請求項 20】

前記プロセッサが、前記第2の専用のハードウェア命令が前記ハードウェアプロセッサの前記専用メモリから値を取り出すGET命令であるような動作を実施するためのプロセッサ実行可能命令で構成される、請求項19に記載のコンピューティングデバイス。

【請求項 21】

前記プロセッサが、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップがコンパイル時に厳密に一度だけ遂行され、前記符号化されたハッシュテーブル整数キーの一部をロードするステップが、動的ソフトウェア言語動作の要求ごとに実行されるような動作を実施するためのプロセッサ実行可能命令で構成される、請求項20に記載のコンピューティングデバイス。

【請求項 22】

前記プロセッサが、

前記動的ソフトウェア言語コードがJavaScript(登録商標)コードであり、

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップが、JavaScriptハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するようにJavaScriptコードをコンパイルするステップを含む

ような動作を実施するためのプロセッサ実行可能命令で構成される、請求項20に記載のコンピューティングデバイス。

【請求項 23】

前記プロセッサが、JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップがコンパイル時に厳密に一度だけ遂行され、

10

20

30

40

50

前記符号化されたハッシュテーブル整数キーの一部をロードするステップが、JavaScript(登録商標)動作の要求ごとに行われるような動作を実施するためのプロセッサ実行可能命令で構成される、請求項22に記載のコンピューティングデバイス。

【請求項24】

前記プロセッサが、前記第1の専用のハードウェア命令がメモリマップされたロケーションに値を記憶することによって前記専用のハードウェアハッシュテーブルインプリメンテーションに前記値をプッシュするSTORE命令であるような動作を実施するためのプロセッサ実行可能命令で構成される、請求項14に記載のコンピューティングデバイス。

【請求項25】

前記プロセッサが、前記符号化されたハッシュテーブル整数キーの一部を専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップが、前記符号化されたハッシュテーブル整数キーの一部を、1次プロセッサに結合されたコプロセッサ上の専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップを含むような動作を実施するためのプロセッサ実行可能命令で構成される、請求項14に記載のコンピューティングデバイス。

10

【請求項26】

前記プロセッサが第1のプロセッサと第2のプロセッサとを備え、前記プロセッサが、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップが前記第1のプロセッサによって実施され、前記符号化されたハッシュテーブル整数キーの一部を第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップが前記コンピューティングデバイスの前記第2のプロセッサによって実施されるような動作を実施するためのプロセッサ実行可能命令で構成される、請求項14に記載のコンピューティングデバイス。

20

【請求項27】

前記第2のプロセッサが専用のハッシュテーブルコプロセッサである、請求項26に記載のコンピューティングデバイス。

【請求項28】

前記プロセッサが、

ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、前記専用のハッシュテーブルコプロセッサ上で第2の専用のハードウェア命令を実行することによって、前記ハッシュテーブル値が取り出され得るかどうかを判断するステップと、

30

前記ハッシュテーブル値が取り出され得るものと判断される場合、前記コンピューティングデバイスの前記第1のプロセッサ上で対応する前記ハッシュテーブル整数キーを受信するために、前記専用のハッシュテーブルコプロセッサ上でハードウェアベースのハッシュテーブルルックアップを実行するステップと

をさらに含む動作を実施するためのプロセッサ実行可能命令で構成される、請求項27に記載のコンピューティングデバイス。

【請求項29】

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように動的ソフトウェア言語コードをコンパイルするための手段と、

40

前記符号化されたハッシュテーブル整数キーをソフトウェアによってアクセス可能なメモリに記憶するための手段と、

前記符号化されたハッシュテーブル整数キーの一部を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段と

を含む、コンピューティングデバイス。

【請求項30】

前記メモリが前記コンピューティングデバイスのプロセッサ内のハッシュテーブルメモリである、請求項29に記載のコンピューティングデバイス。

50

【請求項 3 1】

前記メモリが前記コンピューティングデバイスのプロセッサの外部にある、請求項29に記載のコンピューティングデバイス。

【請求項 3 2】

前記符号化されたハッシュテーブル整数キーの一部を第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段が、前記符号化されたハッシュテーブル整数キーの一部を、専用のハードウェアハッシュテーブルインプリメンテーションに値をプッシュするSET命令を介して前記専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段を含む、請求項29に記載のコンピューティングデバイス。

10

【請求項 3 3】

ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、ハードウェアプロセッサ上で専用のハードウェア命令を実行することによって、ハッシュテーブル値が前記ハードウェアプロセッサによって取り出され得るかどうかを判断するための手段と、

前記キーが前記ハードウェアプロセッサに記憶されている場合、対応する前記ハッシュテーブル整数キーを返すためにハードウェアベースのハッシュテーブルルックアップを実行するための手段と

をさらに含む、請求項32に記載のコンピューティングデバイス。

【請求項 3 4】

20

前記キーが前記ハードウェアプロセッサ上にない場合、ソフトウェアアクセス可能なメモリに記憶されている、要求された前記ハッシュテーブル値と前記符号化されたハッシュテーブル整数キーの一部とを取り出すために前記ソフトウェアベースのハッシュテーブルルックアップを実行するための手段と、

符号化されたハッシュテーブル整数キーの前記取り出された部分を、第2の専用のハードウェア命令を介して前記ハードウェアプロセッサの専用メモリにロードすることによって、前記専用のハードウェアハッシュテーブルインプリメンテーションを更新するための手段と

をさらに含む、請求項33に記載のコンピューティングデバイス。

【請求項 3 5】

30

符号化されたハッシュテーブル整数キーの前記取り出された部分を第2の専用のハードウェア命令を介して前記ハードウェアプロセッサの専用メモリにロードすることによって前記専用のハードウェアハッシュテーブルインプリメンテーションを更新するための手段が、符号化されたハッシュテーブル整数キーの前記取り出された部分を前記ハードウェアプロセッサの前記専用メモリから値を取り出すGET命令を介して前記ハードウェアプロセッサの専用メモリにロードすることによって前記専用のハードウェアハッシュテーブルインプリメンテーションを更新するための手段を含む、請求項34に記載のコンピューティングデバイス。

【請求項 3 6】

40

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするための手段が、コンパイル時に厳密に一度だけハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするための手段を含み、

前記符号化されたハッシュテーブル整数キーの一部を第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段が、前記符号化されたハッシュテーブル整数キーの一部を、動的ソフトウェア言語動作の要求ごとに第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段を含む、請求項35に記載のコンピューティングデバイス。

【請求項 3 7】

50

前記動的ソフトウェア言語コードがJavaScript(登録商標)コードであり、

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするための手段が、JavaScriptハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するようにJavaScriptコードをコンパイルするための手段を含む、
請求項35に記載のコンピューティングデバイス。

【請求項38】

JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するための手段が、コンパイル時に厳密に一度だけJavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するための手段を含む、

10

前記符号化されたハッシュテーブル整数キーの一部を第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段が、JavaScript(登録商標)動作の要求ごとに、前記符号化されたハッシュテーブル整数キーの一部を第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段を含む、請求項37に記載のコンピューティングデバイス。

【請求項39】

前記第1の専用のハードウェア命令が、メモリマップされたロケーションに値を記憶することによって、前記専用のハードウェアハッシュテーブルインプリメンテーションに前記値をプッシュするSTORE命令である、請求項29に記載のコンピューティングデバイス。

20

【請求項40】

前記符号化されたハッシュテーブル整数キーの一部を専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段が、前記符号化されたハッシュテーブル整数キーの一部を、1次プロセッサに結合されたコプロセッサ上の専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段を含む、請求項29に記載のコンピューティングデバイス。

【請求項41】

第1のプロセッサと、

第2のプロセッサとを備え、

30

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするための手段が、前記第1のプロセッサによってハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするための手段を含む、

前記符号化されたハッシュテーブル整数キーの一部を第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段が、前記符号化されたハッシュテーブル整数キーの一部を、前記第2のプロセッサによって第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段を含む、
請求項29に記載のコンピューティングデバイス。

40

【請求項42】

前記第2のプロセッサが専門のハッシュテーブルコプロセッサである、請求項41に記載のコンピューティングデバイス。

【請求項43】

ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、前記専門のハッシュテーブルコプロセッサ上で第2の専用のハードウェア命令を実行することによって、前記ハッシュテーブル値が取り出され得るかどうかを判断するための手段と、

前記ハッシュテーブル値が取り出され得るものと判断される場合、前記コンピューティングデバイスの前記第1のプロセッサ上で対応する前記ハッシュテーブル整数キーを受信するために、前記専門のハッシュテーブルコプロセッサ上でハードウェアベースのハッシ

50

ユテーブルルックアップを実行するための手段と
をさらに含む、請求項42に記載のコンピューティングデバイス。

【請求項 4 4】

プロセッサ実行可能命令を記憶した非一時的プロセッサ可読媒体であって、前記プロセッサ実行可能命令がプロセッサに、

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように動的ソフトウェア言語コードをコンパイルするステップと、

前記符号化されたハッシュテーブル整数キーをソフトウェアによってアクセス可能なメモリに記憶するステップと、

前記符号化されたハッシュテーブル整数キーの一部を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップと

を含む動作を実行させるように構成される、非一時的プロセッサ可読媒体。

【請求項 4 5】

前記記憶されたプロセッサ実行可能命令がプロセッサ上で機能するように構成され、前記メモリが前記プロセッサ内のハッシュテーブルメモリである、請求項44に記載の非一時的プロセッサ可読媒体。

【請求項 4 6】

前記記憶されたプロセッサ実行可能命令がプロセッサ上で機能するように構成され、前記メモリが前記プロセッサの外部にある、請求項44に記載の非一時的プロセッサ可読媒体

【請求項 4 7】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、前記第1の専用のハードウェア命令が前記専用のハードウェアハッシュテーブルインプリメンテーションに値をプッシュするSET命令であるような動作を実施させるように構成される、請求項44に記載の非一時的プロセッサ可読媒体。

【請求項 4 8】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、

ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、ハードウェアプロセッサ上で専用のハードウェア命令を実行することによって、ハッシュテーブル値が前記ハードウェアプロセッサによって取り出され得るかどうかを判断するステップと、

前記キーが前記ハードウェアプロセッサに記憶されている場合、対応する前記ハッシュテーブル整数キーを返すためにハードウェアベースのハッシュテーブルルックアップを実行するステップと

をさらに含む動作を実施させるように構成される、請求項47に記載の非一時的プロセッサ可読媒体。

【請求項 4 9】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、

前記キーが前記ハードウェアプロセッサ上にない場合、ソフトウェアアクセス可能なメモリに記憶されている、要求された前記ハッシュテーブル値と前記符号化されたハッシュテーブル整数キーの一部とを取り出すためにソフトウェアベースのハッシュテーブルルックアップを実行するステップと、

符号化されたハッシュテーブル整数キーの前記取り出された部分を、第2の専用のハードウェア命令を介して前記ハードウェアプロセッサの専用メモリにロードすることによって、前記専用のハードウェアハッシュテーブルインプリメンテーションを更新するステップと

をさらに含む動作を実施させるように構成される、請求項48に記載の非一時的プロセッサ可読媒体。

【請求項 5 0】

10

20

30

40

50

前記記憶されたプロセッサ実行可能命令が、プロセッサに、前記第2の専用のハードウェア命令が前記ハードウェアプロセッサの前記専用メモリから値を取り出すGET命令であるような動作を実施させるように構成される、請求項49に記載の非一時的プロセッサ可読媒体。

【請求項51】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップがコンパイル時に厳密に一度だけ遂行され、前記符号化されたハッシュテーブル整数キーの一部をロードするステップが動的ソフトウェア言語動作の要求ごとに行われるような動作を実施させるように構成される、請求項50に記載の非一時的プロセッサ可読媒体。

10

【請求項52】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、前記動的ソフトウェア言語コードがJavaScript(登録商標)コードであり、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップが、JavaScriptハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するようにJavaScriptコードをコンパイルするステップを含むような動作を実施させるように構成される、請求項50に記載の非一時的プロセッサ可読媒体。

20

【請求項53】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップがコンパイル時に厳密に一度だけ遂行され、前記符号化されたハッシュテーブル整数キーの一部をロードするステップがJavaScript(登録商標)動作の要求ごとに行われるような動作を実施させるように構成される、請求項52に記載の非一時的プロセッサ可読媒体。

【請求項54】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、前記第1の専用のハードウェア命令が、メモリマップされたロケーションに値を記憶することによって前記専用のハードウェアハッシュテーブルインプリメンテーションに前記値をプッシュするSTORE命令であるような動作を実施させるように構成される、請求項44に記載の非一時的プロセッサ可読媒体。

30

【請求項55】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、前記符号化されたハッシュテーブル整数キーの一部を専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップが、前記符号化されたハッシュテーブル整数キーの一部を、1次プロセッサに結合されたコプロセッサ上の専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップを含むような動作を実施させるように構成される、請求項44に記載の非一時的プロセッサ可読媒体。

【請求項56】

前記記憶されたプロセッサ実行可能命令が第1のプロセッサと第2のプロセッサとを備えるプロセッサによって実行可能であるように構成され、前記記憶されたプロセッサ実行可能命令が、前記プロセッサに、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップが前記第1のプロセッサによって実施され、前記符号化されたハッシュテーブル整数キーの一部を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップが前記コンピューティングデバイスの前記第2のプロセッサによって実施されるような動作を実施させるように構成される、請求項44に記載の非一時的プロセッサ可読媒体。

40

【請求項57】

前記第2のプロセッサが専門のハッシュテーブルコプロセッサである、請求項56に記載

50

の非一時的プロセッサ可読媒体。

【請求項58】

前記記憶されたプロセッサ実行可能命令が、前記プロセッサに、ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、前記専門のハッシュテーブルコプロセッサ上で第2の専用のハードウェア命令を実行することによって、前記ハッシュテーブル値が取り出され得るかどうかを判断するステップと、

前記ハッシュテーブル値が取り出され得るものと判断される場合、前記コンピューティングデバイスの前記第1のプロセッサ上で対応する前記ハッシュテーブル整数キーを受信するために、前記専門のハッシュテーブルコプロセッサ上でハードウェアベースのハッシュテーブルルックアップを実行するステップと

をさらに含む動作を実施させるように構成される、請求項57に記載の非一時的プロセッサ可読媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本出願は、一般に、コンピュータのソフトウェアおよびハードウェアに関し、より詳細には、ハッシュテーブル動作を実施するエネルギーコストを低減すると同時に、ハッシュテーブル動作の実行を加速するための方法、システムおよびデバイスに関する。

【0002】

関連出願

本出願は、すべての目的のためにその内容全体が参照によって本明細書に組み込まれる、2011年5月27日に提出した「Hardware Support for Hashtables in Dynamic Languages」という名称の米国仮特許出願第61/491,075号の優先権の利益を主張する。

【背景技術】

【0003】

動的スクリプト言語は、コンピュータプログラミングおよびソフトウェアの開発において好適な開発プラットフォームである。特に、JavaScript(登録商標)は、ウェブページおよびウェブアプリケーションのための、普及している開発言語である。ますます多くの特徴が、ウェブページおよびウェブアプリケーションのユーザによって要求されているので、動的スクリプト言語の複雑さおよび量が増加し続けている。その結果、ますます多くの時間量が、コンピューティングデバイスがスクリプトを構文解析し、解釈し、分析し、実行することによって費やされ、ウェブページのロード時間が遅くなり、ユーザエクスペリエンスの劣化を招いている。

【0004】

JavaScript(登録商標)など、いくつかの動的ソフトウェア言語は、ハッシュテーブル機能をそれらのコアに実装しているものと考えられ得る。コンピュータサイエンスでは、ハッシュテーブルまたはハッシュマップは、キーとして知られる識別値(たとえば、人名)をそれらの関連値(たとえば、それらの電話番号)にマップするために、ハッシュ関数を使用するデータ構造である。したがって、ハッシュテーブルは、連想配列を実装する。ハッシュ関数は、キーを、対応する値が求められるアレイ要素(またはメモリロケーション)のインデックス(ハッシュ)に変換するために使用される。

【発明の概要】

【課題を解決するための手段】

【0005】

JavaScript(登録商標)のようなソフトウェア言語の動作を、基本的なハッシュテーブル動作となるように特徴づけることによって、専門のハードウェアハッシュテーブルプロセッサが、コンピューティングデバイスの処理、効率改良、速度、および電力消費を支援するために使用され得る。本明細書で開示する様々な態様は、効率的なハッシュテーブルの特徴を実装するJavaScript(登録商標)ランタイムエンジンのための方法、システムおよびデバイスを提供する。一態様では、JavaScript(登録商標)コードは、JavaScript(登録商

10

20

30

40

50

標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するようにコンパイルし、符号化されたハッシュテーブル整数キーは、ソフトウェアによってアクセス可能なメモリに記憶され得る。符号化されたハッシュテーブル整数キーの一部は、第1の専用のハードウェア命令を介してハードウェアプロセッサの専用メモリにロードされ得る。ハッシュテーブルルックアップを必要とするJavaScript(登録商標)動作において、プロセッサ上の専用のハードウェア命令は、ハッシュテーブル値がハードウェアプロセッサによって取り出され得るかどうかを判断するために実行され得る。キーがハードウェアプロセッサ内に記憶されている場合、ハードウェアベースのハッシュテーブルルックアップが、対応するハッシュテーブル整数キーを返すために実行され得る。キーがハードウェアプロセッサ内に記憶されていない場合、ソフトウェアベースのハッシュテーブルルックアップが、要求されたハッシュテーブル値を取り出すために実行され得、符号化されたハッシュテーブル整数キーの一部が、ソフトウェアアクセス可能なメモリに記憶され得る。一態様では、ハードウェアプロセッサの専用メモリが、符号化されたハッシュテーブル整数キーの取り出された一部分を、第2の専用のハードウェア命令を介してハードウェアプロセッサの専用メモリにロードするステップによって更新され得る。

10

20

30

40

50

【0006】

一態様は、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように動的ソフトウェア言語コードをコンパイルするステップと、符号化されたハッシュテーブル整数キーをソフトウェアによってアクセス可能なメモリに記憶するステップと、符号化されたハッシュテーブル整数キーの一部を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップとの動作を含む、動的ソフトウェア言語コードを実行するための方法を含む。一態様の方法では、第1の専用のハードウェア命令は、専用のハードウェアハッシュテーブルインプリメンテーションに値をプッシュするSET命令であってよい。一態様の方法は、ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとにハードウェアプロセッサ上で専用のハードウェア命令を実行することと、キーがハードウェアプロセッサ内に記憶されている場合、対応するハッシュテーブル整数キーを返すためにハードウェアベースのハッシュテーブルルックアップを実行することとによって、ハッシュテーブル値がハードウェアプロセッサによって取り出され得るかどうか判断するステップをさらに含むことができる。一態様の方法は、キーがハードウェアプロセッサ上にない場合、ソフトウェアアクセス可能なメモリに記憶されている、要求されたハッシュテーブル値と符号化されたハッシュテーブル整数キーの一部とを取り出すためにソフトウェアベースのハッシュテーブルルックアップを実行するステップと、符号化されたハッシュテーブル整数キーの取り出された一部分を、第2の専用のハードウェア命令を介してハードウェアプロセッサの専用メモリにロードすることによって、専用のハードウェアハッシュテーブルインプリメンテーションを更新するステップとをさらに含むことができる。一態様の方法では、第2の専用のハードウェア命令は、ハードウェアプロセッサの専用メモリから値を取り出すGET命令であってよい。一態様の方法では、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化することは、コンパイル時に厳密に一度だけ遂行され得、符号化されたハッシュテーブル整数キーの一部をロードするステップは、動的ソフトウェア言語動作の要求ごとに実行される。一態様の方法では、動的ソフトウェア言語コードはJavaScript(登録商標)コードであってよく、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように動的ソフトウェア言語コードをコンパイルするステップは、JavaScriptハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するようにJavaScriptコードをコンパイルするステップを含むことができる。一態様の方法では、JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップは、コンパイル時に厳密に一度だけ遂行され得、符号化されたハッシュテーブル整数キーの一部をロードするステップは、JavaScript(登録商標)動作の要求ごとに実行され得る。一態様の方法では、第1の専用のハードウェア命令は、メモリマップされたロケーションに値を記憶することによって、専用の

ハードウェアハッシュテーブルインプリメンテーションに値をプッシュするSTORE命令であってよい。一態様の方法では、符号化されたハッシュテーブル整数キーの一部を専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップは、符号化されたハッシュテーブル整数キーの一部を、1次プロセッサに結合されたコプロセッサ上の専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップを含むことができる。一態様の方法では、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように動的ソフトウェア言語コードをコンパイルするステップは、コンピューティングデバイスの第1のプロセッサによって実施され得、符号化されたハッシュテーブル整数キーの一部を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップは、コンピューティングデバイスの第2のプロセッサによって実施される。一態様の方法では、コンピューティングデバイスの第2のプロセッサは専門のハッシュテーブルコプロセッサであってよく、方法は、ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、専門のハッシュテーブルコプロセッサ上で第2の専用のハードウェア命令を実行することによって、ハッシュテーブル値が取り出され得るかどうかを判断するステップと、ハッシュテーブル値が取り出され得るものと判断される場合、コンピューティングデバイスの第1のプロセッサ上で対応するハッシュテーブル整数キーを受け取るために、専門のハッシュテーブルコプロセッサ上でハードウェアベースのハッシュテーブルルックアップを実行するステップとをさらに含むことができる。

10

20

【0007】

別の態様は、メモリに結合されたプロセッサを含むコンピューティングデバイスを含み、プロセッサは、上記で説明した様々な態様の方法の動作を実施するためのプロセッサ実行可能命令で構成される。別の態様は、上記で説明した様々な態様の方法の機能と動作とを実施するための手段を含むコンピューティングデバイスを含む。別の態様は、プロセッサに、上記で説明した様々な態様の方法の動作を実施させるように構成されたプロセッサ実行可能命令を記憶した、非一時的プロセッサ可読媒体を含む。

【0008】

本明細書に組み込まれ、本明細書の一部をなす添付の図面は、本発明の例示的な態様を示し、上記の全般的な説明および下記の発明を実施するための形態とともに、本発明の特徴を説明するのに役立つ。

30

【図面の簡単な説明】

【0009】

【図1】例示的なアレイの特徴を示すデータ構造図である。

【図2】連想配列の例示的なインプリメンテーションの特徴を示すデータ構造図である。

【図3】連想配列のハッシュテーブルのインプリメンテーションの特徴を示すデータ構造図である。

【図4】例示的なページテーブル変換システムを示すプロセッサの機能構成要素図である。

【図5A】様々な態様による、ハッシュテーブル動作をハードウェア内で実行するためのコンピューティングデバイスアーキテクチャの機能構成要素図である。

40

【図5B】様々な態様による、ハッシュテーブル動作をハードウェア内で実行するためのメモリおよびプロセッサ構成要素を含む、代替コンピューティングデバイスアーキテクチャの回路ブロック図である。

【図5C】様々な態様による、ハッシュテーブル動作をハードウェア内で実行するためのメモリおよびプロセッサ構成要素を含む、代替コンピューティングデバイスアーキテクチャの回路ブロック図である。

【図5D】様々な態様による、ハッシュテーブル動作をハードウェア内で実行するためのメモリおよびプロセッサ構成要素を含む、代替コンピューティングデバイスアーキテクチャの回路ブロック図である。

【図6】様々な態様とともに使用するのに適した例示的な方法のプロセスフロー図である

50

。

【図7】様々な態様とともに使用するのに適したモバイルコンピューティングデバイスの破断斜視図である。

【図8】様々な態様とともに使用するのに適した例示的なコンピュータの図である。

【発明を実施するための形態】

【0010】

様々な態様が、添付の図面を参照して詳細に説明される。可能な場合はいつでも、同じまたは同様の部分を指すのに、図面全体にわたって同じ参照番号を使用する。特定の例およびインプリメンテーションに行われる参照は、説明を目的とし、本発明の範囲または本特許請求の範囲を限定することは意図されない。

10

【0011】

「例示的な」という言葉は、「一例、実例または例として」を意味するために本明細書で使用される。本明細書に「例示的」と記載されるいかなるインプリメンテーションも、必ずしも他のインプリメンテーションよりも好ましいまたは有利であると解釈されるべきではない。

【0012】

「電子デバイス」、「モバイルデバイス」および「コンピューティングデバイス」という用語は、パーソナルコンピュータ、セルラー電話、携帯情報端末(PDA)、パームトップコンピュータ、タブレットコンピュータ、ワイヤレス電子メール受信機(たとえば、ブラックベリー(登録商標)デバイス)、マルチメディアインターネット対応セルラー電話、およびプログラマブルプロセッサを含む類似の携帯電子デバイスのうちの1つまたは全部を指すために、本明細書では互換的に使用される。様々な態様は、限られた処理電力しか有し得ないセルラー電話など、モバイルデバイスにおいて特に有用であるが、態様は、全体として、動的言語、スクリプト言語および/またはマークアップ言語で書かれたスクリプトおよび/またはアプリケーションを実行する任意のコンピューティングデバイスにおいて有用である。

20

【0013】

「動的言語」および「スクリプト言語」という用語は、本出願では、任意の動的言語もしくはスクリプト言語、またはランタイムに解釈され、および/またはコンパイルされるプログラムを(本明細書では「スクリプト」として)書くために使用される任意のインタープリタ型言語を指すために、総称的および互換的に使用される。これらの用語はまた、管理されたランタイムに作動する任意の言語を指すこともできる。したがって、本出願では、「動的言語」および「スクリプト言語」という用語は、ソースコードもしくはバイトコードから解釈される言語、または従来ネイティブマシンコードにコンパイルされるプログラムとともに実行する言語に限定されるべきではない。本出願の範囲内の動的言語およびスクリプト言語の例は、たとえば、JavaScript(登録商標)、Perl、PythonおよびRuby、ならびに今後開発される可能性があるJava(登録商標)および他の言語を含む。

30

【0014】

参照しやすいように、様々な態様を、JavaScript(登録商標)用語を使用して説明する。しかしながら、この用語の使用およびJavaScript(登録商標)の例の使用は、特許請求の範囲に明確に記載されないならば、特許請求の範囲を特定のソフトウェア言語またはソフトウェア技術に限定するものと解釈されるべきではない。

40

【0015】

図1~図3は、様々な既知のデータ構造のデータ構造要素および基本動作を示す。これらの構造はよく知られているので、多くの詳細(たとえば、異なる型のデータ、メモリサイズなどの記憶)は、重要な特徴の説明に焦点を当てるために省略されている。

【0016】

図1は、従業員の識別子(ID)の値を記憶している例示的なアレイ100の図である。アレイは、一般に、各値がメモリ内のその相対位置による参照符号(すなわち、インデックス)であり得るように、いくつかの値を連続的にメモリ内に記憶する。たとえば、図1は、例示

50

的なアレイ100が、従業員のID値を、インデックス(たとえば、1、2、3)に関する各値(たとえば、ID)によって連続的に記憶することを示している。アレイの連続的な性質は、インデックスに基づいてアレイから値を取り出すための命令(たとえば、`getItem[2]`)は、一般に、システムがアレイ内の第1の要素のメモリロケーション(たとえば、メモリロケーション「1005」)にアクセスし、インデックス値(たとえば、「2」)だけメモリロケーションをインクリメントして所望の項目(たとえば、メモリロケーション「1007」)における値「489954」)を探し出すことを必要とすることを意味する。アレイ内の特定の値のロケーションは、一般に、アレイに対して知られていない。特定の値を発見するために、アレイ内の各項目は、値が発見されてその値に関連するインデックスが同定されるまで調査され得る。

10

【0017】

図2は、連想配列200の例示的なインプリメンテーションの図である。連想配列は、一意キーの集合および値の集合からなる抽象データ型であり、ここで各キーは1つの値または値の1セットに関連付けられる。図2は、連想配列200が、所望の値(たとえば、従業員ID)が記憶される可能性がある他のメモリロケーション(たとえば、メモリロケーション20、105、155)を特定する一意キー値(たとえば、整数値20、105、155)をいかに含有しているかを示している。図2はまた、各従業員ID202が、従業員名204、電話番号206、住所208および職位210など、他の関連情報に関連して記憶され得ることを示している。このようにして、所望の値(たとえば、従業員ID)が探し出されると、所望の値に関連する他の情報もまた、所望の値に関連するメモリ内のそれらの位置に基づいて探し出され得る。たとえば、そのインデックスに基づく値を取り出すための命令(たとえば、`getItem[2]`)は、アレイ内の第1の要素のメモリロケーション(たとえば、メモリロケーション1005)にアクセスし、所望のキー値(たとえば、メモリロケーション1007におけるキー155)を探し出すためにメモリロケーションをインデックス値(たとえば、2)だけインクリメントし、キー値に関連する第2のメモリロケーション(たとえば、メモリロケーション155)にアクセスし、第2のメモリに記憶されている値(または値へのポインタ)(たとえば、メモリロケーション155における値489954)を返すように、プロセッサに命令することができる。したがって、所望の値に関連する情報(所望の値に関連して記憶されている)は、所望の値と同時に取り出され得る。

20

【0018】

図2を参照して上記で説明した例示的なインプリメンテーションは、複数の項目が、単一のアレイ要素(すなわち、キー)に関連付けられることを可能にするが、アレイ要素にアクセスすることは、依然として順次プロセスである(たとえば、インデックス値は依然としてオフセットである)。この型のデータ構造の型では、アレイをトラバース、探索、および/またはソートすることなく所与の値のロケーション(たとえば、インデックス)を決定するための効率的な方法は存在しない。アレイをソートおよび/または探索するために多くの技法が存在するが、アレイ動作は、依然として比較的リソースを消費するものであり遅い。

30

【0019】

図3は、連想配列の例示的なハッシュテーブルインプリメンテーションを示す。ハッシュテーブルはハッシュ関数を使用して、キー値を、対応する値が求められるアレイ要素のインデックス(たとえば、メモリロケーション)に変換することによって、キーをそれらの関連付けられた値にマップする。図3は、アレイのメモリロケーション(たとえば、メモリロケーション1005、1006、1007)が、キー値として従業員IDを含有することができる一例を示す。ハッシュ関数は、キー値(たとえば、従業員ID)上で実施されて、対応する値のロケーション(たとえば、メモリロケーション739、846、954)を特定するハッシュ値(たとえば、846、739、954)を生成することができる。ハッシュ関数は、キー値をハッシュ値に変換するために、多数の利用可能な数式のうちのいずれか1つを使用することができる。図3に示す例では、ハッシュ関数は、ハッシュ値としてキー値の最後の3桁を使用し、ハッシュによって特定されたロケーションに対する関連情報(たとえば、名前)を記憶する。この

40

50

ようにして、ハッシュテーブルから値を取り出すための命令(たとえば、getItem[489954])は、単に、キー値(たとえば、489954)をハッシュすることと、所望の項目(たとえば、メモリロケーション954におけるJohn Doe)を探し出すためにハッシュ値(954)のメモリロケーションにアクセスすることとを必要とするだけである。

【0020】

ハッシュテーブルは非常に高速で効率的であるので、基本的なハッシュテーブル機能を実装することで、効率、速度、エネルギー消費、ならびに多くの他の性能要因が、大きく改善される。しかしながら、ハッシュテーブルは、全体的にソフトウェア構成体であり、当今のハードウェアメーカーは、固有のハッシュテーブル機能を提供するプロセッサを供給していない。

10

【0021】

様々な態様は、アプリケーションが、ハッシュテーブル動作(たとえば、ルックアップ)をハードウェアで実行することを可能にする方法、システムおよびデバイスを提供する。ハッシュテーブル動作の実行は、メモリマップされた入力/出力(MMIO)アクセラレータデバイスを介してマイクロプロセッサ上の特殊な命令レジスタによって、および/または1つまたは複数のコプロセッサを使用することによって、支援または制御され得る。様々な態様はまた、ハッシュテーブル機能を局所性の原理と組み合わせて、JavaScript(登録商標)など、基本的なハッシュテーブル機能を実装するソフトウェア言語の性能を大きく改善する。

20

【0022】

「局所性の原理(principle of locality)」は、典型的に、仮想メモリおよびキャッシュを設計することに関連する原理である。局所性の原理は、概して、プロセス要求がメモリ内のある項目にアクセスする場合、近い将来にいつか、そのメモリに近い別の項目へのアクセスを要求する可能性が非常に高いことを示す。その原理によれば、(要求された項目だけを取り出すのではなく)その項目が位置するメモリブロック全体をフェッチし、フェッチされたブロックを、今後のアクセスのためにより速くより査定可能なメモリに移すことが、全体的に有益である。

【0023】

局所性の原理を利用する技法を実装することが、ハードウェアシステムにとって有利である。今日のマイクロプロセッサは、当今のメモリよりもずっと高速に動作する。したがって、処理においてハードウェアの主要なボトルネックは、外部メモリにアクセスするためにかかる時間である。たとえば、メモリが比較的低速であり、それゆえメモリアクセスを遂行するのに時間がかかるため、マイクロプロセッサは、メモリが次の動作に対してレディになるのに十分な時間を取るために、メモリ読出または書込み動作を伴うプロセッサ動作の間、待ち状態に入る(または、異なるスレッドのための処理に取り組む)ことがある。プロセッサのこの待ち時間を低減するために、近年のコンピュータアーキテクチャは、メモリアクセスの速度をプロセッサの動作速度に整合させるキャッシュメモリを使用する。キャッシュメモリは、プロセッサが、コンピューティングデバイス内の他のメモリ(たとえば、ハードディスク、キャパシタベースメモリ、ランダムアクセスメモリ、読出し専用メモリなど)よりずっと高速にアクセス可能な小型、高価で高速のメモリである。所与のメモリアドレスにおけるバイトが要求されるときはいつでも、プロセッサは最初に、より速いキャッシュメモリからデータを取得することを試みる。しかしながら、キャッシュが要求されたデータを含んでいない場合、プロセッサは、要求されたデータが他のより遅いシステムメモリのうちの1つからロードされる間、待ち状態に入る。したがって、キャッシュメモリの有効性(たとえば、処理速度の総合的改善)は、プロセッサによって要求されるデータを、キャッシュメモリがどれほどひんばんに含有しているかに直接関係する。

30

40

【0024】

性能を極大化するために、プロセッサまたはキャッシュメモリコントローラは、プロセッサが次のまたは今後の動作で必要とするデータを予測して、プロセッサがそのデータを

50

要求する前にそのデータをキャッシュにロードすることができる。所与のメモリロケーションを探し出すことは、メモリからデータを読み出すよりずっと遅いので、当今のキャッシュメモリは、局所性の原理(すなわち、プロセッサは所与の時点においてアドレス空間の小部分だけにアクセスする傾向があること)を利用して、参照されたメモリの近くに記憶されているデータもまた、近い将来にプロセッサによって参照されそうであるとの予測のもとに、データのブロック全体をロードする。

【 0 0 2 5 】

総合的なプロセッサ性能を改善するために局所性の原理を利用することに加えて、様々なオペレーティングシステムの当今のページテーブル変換システムは、局所性の原理を使用して、ページテーブルを仮想メモリから物理メモリに変換する。その結果、オペレーティングシステムは、本来物理メモリに適合するより多くのプロセスを、並行して実行することができる。

10

【 0 0 2 6 】

図4は、例示的なページテーブル変換システム400を実装するコンピューティングデバイスの機能構成要素を示す。図4に示すように、コンピューティングデバイス上で実行しているオペレーティングシステム402は、多数のプロセス410の並行実行をサポートするために、物理メモリ406と仮想メモリ408との組合せを使用することができる。物理メモリ406は、コンピュータに実際にインストールされるメモリモジュールのサイズおよび数によって制限されるアドレス空間(メモリ1~n)を提供する。対照的に、仮想メモリ408は、コンピューティングデバイスが、物理メモリ406に実際に適合するより多くのプロセスを並行して実行することを可能にする、ずっと大きいアドレス空間(メモリ1~m)を提供する。すなわち、仮想メモリ408の存在によって、コンピューティングデバイスが、主としてより速いが小さなメモリ(たとえば、物理メモリ406)によって稼働すると同時に、より遅いがずっと大きい大容量メモリ(たとえば、仮想メモリ408)に記憶されている情報へのアクセスを有することが可能になる。物理メモリ406内にはない情報へのアクセスをプロセスが要求するときはいつでも、コンピューティングデバイスは、仮想メモリ408から情報をフェッチして、要求された情報(または、情報を含むブロック)を物理メモリ406に移すことができる。ページテーブル412が、物理メモリ406を仮想メモリ408にマップするために使用され得る。ページテーブル412について、以下でより詳細に説明する。

20

【 0 0 2 7 】

図4は、オペレーティングシステム402が、仮想メモリ408と物理メモリ406との間でアドレス変換方式を実施するために使用されるマッピング情報(すなわち、アドレスマッピング)を記憶するページテーブル412を含むことができることを示す。同様に、ハードウェア404は、ページテーブル412から最近使用されたマッピングを含む変換ルックアップバッファ414を含むことができる。仮想から物理へのアドレス変換が必要とされるときはいつでも、システムは、変換ルックアップバッファ414が必要なマッピング情報を含むかどうかを判断するために、バッファ414を最初に調べることができる。変換ルックアップバッファ414が必要なマッピング情報を含む(たとえば、マッチが発見される)場合、物理メモリアドレスが、要求しているプロセスに提供され、データアクセスが進行する。しかしながら、マッチが発見されない場合、オペレーティングシステム402は、ページテーブル412が、要求されているマッピング情報を含むかどうかを判断するために、ページテーブル412を調べる。マッピング情報がページテーブル412内で発見された場合、情報(または、その情報を含むブロック)が、ハードウェア404の変換バッファ414に書き込まれ得、その後、データアクセスが進行する。

30

40

【 0 0 2 8 】

上記で説明したように、ページテーブル412は、物理メモリ406を仮想メモリ408にマップし得、変換ルックアップバッファ414は、最近使用および/または予測された物理メモリのマッピング情報のブロックを含むことができる。マッピングの性質に従って、変換ルックアップバッファ414は、ハードウェアハッシュテーブルの1次のハードウェア実装として見なされ得る。しかしながら、この観点から見る場合でも、現在の解決策(たとえば

50

、ページテーブル変換システム)は、オペレーティングシステムプロセスだけに対する、1タイプのアプリケーション(たとえば、ページテーブルルックアップ)のためだけの「スーパーユーザモード」においてのみアクセス可能であるように、プロセッサの高度に保護された層において、このマッピングを実施する。

【0029】

様々な態様は、アプリケーションが利用可能なハードウェアベースのハッシュ動作を実装する専用のプロセッサ回路を含むシステム、方法およびデバイスを提供する。様々な態様は、ハードウェアハッシュ動作に対してユーザレベル(たとえば、ユーザモード)の許可/アクセスを認める。様々な態様は、ソフトウェアとハードウェアの両方のハッシュテーブル論理で解釈および/または実行され得るコードを生成するコンパイラを提供する。様々な態様は、アプリケーション開発者が、ソフトウェアプログラムおよびプロセスの効率および実行/解釈時間を改善するために、JavaScript(登録商標)など、様々なソフトウェア言語の基本構造を利用することを可能にする。

10

【0030】

上述のように、JavaScript(登録商標)のようないくつかのソフトウェア言語は、それらのコアにハッシュテーブル機能を実装しているものと見なされ得る。JavaScript(登録商標)は、最も簡単な構成が「オブジェクト」データ型である、オブジェクト指向のスクリプト言語である。JavaScript(登録商標)では、オブジェクトは、名前付きのプロパティの集合として実装され、その集合は、単に、別のJavaScript(登録商標)オブジェクト(すなわち、それらは別のオブジェクトに対して局所的なオブジェクトである)によって所有されるJavaScript(登録商標)オブジェクトにすぎない。プロパティは、ドット演算子(たとえば、「object.property」)または括弧(たとえば、「object[property]」)のいずれかを使用してアクセスされ得る。ハッシュテーブルが、類似のシンタックスおよび論理を使用して要素を探し出す/要素にアクセスすることを考慮すると、各JavaScript(登録商標)オブジェクトは、ハッシュテーブル実装であるものと見なされ得る。たとえば、JavaScript(登録商標)では、オブジェクト「o」のプロパティ「x」に書き込む(たとえば、o.x=10)ことは、オブジェクト「o」へのハッシュ挿入(たとえば、o[x]=10)と等価であるものと見なされ得、オブジェクト「o」のプロパティ「x」を読み出す(たとえば、z=o.x)ことは、オブジェクト「o」内のハッシュ読出し(たとえば、z=o[x])と等価であるものと見なされ得る。この観点から見ると、あらゆるJavaScript(登録商標)動作は、1つのストリング値(キー)から別のストリングまたは数値(値)を通して進む、特殊なタイプのハッシュ動作であるものとして特徴づけられてよく、あらゆるJavaScript(登録商標)オブジェクトは、連想配列のハッシュテーブル実装と見なされてよい。

20

30

【0031】

各JavaScript(登録商標)動作を基本的にハッシュテーブル動作であるものと特徴づけることによって、専用のハッシュテーブルプロセッサが、JavaScript(登録商標)オブジェクトを処理するために使用され得る。専用のハッシュテーブルプロセッサは、ハッシュテーブル動作を実装するために確保された(マイクロプロセッサの中に埋め込まれた)回路を含むことができる。プロセッサは、アプリケーションが、JavaScript(登録商標)機能を実行するためにハードウェアハッシュテーブル論理にアクセスすることを可能にする特殊な命令レジスタで構成され得る。特殊な命令レジスタは、ソフトウェアアプリケーションが、低減された数の命令でハードウェアベースのハッシュ動作にアクセスすることを可能にする、修正された命令セットアーキテクチャ(ISA)を使用してアクセスされ得る。たとえば、JavaScript(登録商標)コンパイラおよびアプリケーションが単一の命令(たとえば、GETまたはSET)を使用してハードウェア内でより容易にハッシュテーブルルックアップを実行し得るように、「get」機械語命令および「set」機械語命令が、専用のハッシュテーブルプロセッサに付加され得る。ハッシュテーブルのハードウェア実装はまた、通常のISAと特殊なレジスタもしくはメモリアドレスとの組合せを介してアクセスされ得る。

40

【0032】

図5Aは、ハードウェアとソフトウェアとの組合せでハッシュテーブル動作(たとえば、J

50

avaScript)を実行するための例示的なコンピューティングデバイスアーキテクチャ500を示す。図5Aは、コンピューティングデバイスアーキテクチャが、専門のプロセッサ510と1つまたは複数のハードウェアハッシュテーブル518とを有するハードウェア504と通信しているオペレーティングシステム502を含むことができることを示す。専門のプロセッサ510は、ハードウェアハッシュテーブル518に対する値を読み出すことおよび書き込むことなど、ハッシュテーブル動作を実行することができる専門の回路を含むことができる。プロセッサ510は、さらに、専門の回路およびハードウェアハッシュテーブル518へのアクセスをともに提供する、「get」機械語命令レジスタ506と「set」機械語命令レジスタ508とを含むように構成され得る。

【0033】

図5Bは、ハードウェアとソフトウェアとの組合せでハッシュテーブル動作を実行するためのハードウェア504の例示的な実施形態のハードウェアアーキテクチャ530を示す。この実施形態のハードウェアアーキテクチャ530は、算術演算および論理演算を実施するための算術論理ユニット(ALU)522と、浮動小数点演算(たとえば、加算、減算、乗算、除算、平方根など)を実施するための浮動小数点ユニット(FPU)524と、ハードウェアハッシュテーブルインプリメンテーションユニット528とを含む中央処理装置520を含むことができる。ハードウェアハッシュテーブルインプリメンテーションユニット528は、上記で説明したハードウェアハッシュテーブル518に対応するハッシュテーブル値を記憶するためのレジスタ/メモリを含むことができる。これらのレジスタおよび/またはメモリは、本明細書では時々、ハッシュテーブルメモリと呼ばれる。

【0034】

図5Cは、ハードウェアとソフトウェアとの組合せでハッシュテーブル動作を実行するためのハードウェア504の、別の例示的な実施形態のハードウェアアーキテクチャ560を示す。この実施形態のハードウェアアーキテクチャ560は、上記で説明したアーキテクチャ530と同様に、算術論理ユニット(ALU)522と浮動小数点ユニット(FPU)524とを含む中央処理装置(CPU)520を含むことができる。ハードウェアアーキテクチャ560はまた、CPU 520と分離されているハードウェアハッシュテーブルインプリメンテーションユニット528を含むことができる。たとえば、ハードウェアハッシュテーブルインプリメンテーションユニット528は、ハッシュ動作を実施し、その結果を並列もしくは直列の入力/出力ポートを介してCPU 520に伝達する専門のコプロセッサ(図示せず)に内在することができる。

【0035】

図5Dは、ハードウェアとソフトウェアとの組合せでハッシュテーブル動作を実行するための、別の例示的な実施形態のハードウェアアーキテクチャ590を示す。この実施形態のハードウェアアーキテクチャ590は、上記で説明したアーキテクチャ530,560と同様に、算術論理ユニット(ALU)522と浮動小数点ユニット(FPU)524とを含む中央処理装置(CPU)520を含むことができる。ハードウェアアーキテクチャ560はまた、CPU 520と分離されているハードウェアハッシュテーブルインプリメンテーションユニット528とメモリマップされた入力/出力(MMIO)アクセラレータデバイス592とを含むことができる。ハードウェアアーキテクチャ590は、CPU 520の外側でハッシュ動作を実施し、その結果をCPU 520にプロセッサバスを介して伝達するために、MMIOアクセラレータデバイス592を使用することができる。

【0036】

図5Aに戻ると、オペレーティングシステム502は、コンパイル時にJavaScript(登録商標)ストリングを整数値にマップする(たとえば、ハッシュする)コンパイラハッシュモジュール516を有するコンパイラ512を含むことができる。一態様では、このマッピングは一度だけ行われる。一態様では、マッピングは、JavaScript(登録商標)オブジェクトのソフトウェアハッシュテーブル514の表示を生成するために使用され得る。ハッシュテーブルを処理するための伝統的な方法は、JavaScript(登録商標)オブジェクトをハードウェアで処理するために使用することはできないので、このマッピングは重要である。上記で説明したように、JavaScript(登録商標)オブジェクトは、ストリング値(キー)を別のストリング

10

20

30

40

50

値または数値(値)にリンクする特殊なタイプのハッシュテーブルであるものと見なされ得る。文字列値を(整数値ではなく)キーとして使用することは、ハッシュテーブルから値をハッシュし、記憶し、取り出すプロセスを複雑にする。たとえば、文字列は、任意の文字数(すなわち、文字は任意の長さであってよい)を含有し得る一方で、整数キーは、既知の数のビットまたはバイトを含有する。文字列をキーとして使用することは、図3を参照しながら上記で説明した、キーの最後の数桁を使用することなど、ハッシュ値を生成するために単純な数学的アルゴリズムを使用することを不可能にする。

【0037】

これらおよび他の理由から、文字列値をキーとして有するハッシュテーブル上でハッシュテーブル動作を実施することで、数千ものソフトウェア命令の実行が必要となる。このことは、これらのハッシュ動作がJavaScript(登録商標)オブジェクト上で実施される場合は特にそうであり、そこでは、入れ子形のハッシュが存在する(Javascript(登録商標)の1つの特徴)ため、単一のハッシュテーブルルックアップ要求が数百ものハッシュテーブルルックアップ要求に変わり、ハッシュテーブルルックアップ要求のそれぞれが、数千ものソフトウェア命令の実行を必要とすることがある。

10

【0038】

文字列をハッシュテーブルキーとして使用するための現在の解決策は、ハードウェア内に効率的に実装され得ない、長くて複雑なソフトウェアベースの処理を必要とする。これらの制約を克服するために、コンパイラハッシュモジュール516は、コンパイル時に(すなわち、ランタイムに反して)、ハッシュテーブル動作(たとえば、JavaScript(登録商標)コード)をハードウェア構成要素で実行することを可能にする方式で、JavaScript(登録商標)文字列を整数値にマップすることができる。コンパイラハッシュモジュール516は、ソフトウェアハッシュテーブル514を生成し得、ソフトウェアハッシュテーブル514は、ハッシュテーブル(たとえば、JavaScript(登録商標)オブジェクト)のハッシュテーブル(たとえば、文字列から整数へのマッピング)である。この二重ハッシング(または、入れ子式ハッシュテーブル)は、記憶された情報が単一の読み出し動作または単一の書き込み動作を使用してアクセスされ得るように、ソフトウェアハッシュテーブル514の一部をハードウェアハッシュテーブル518内に記憶することを可能にする。このことで、各ハッシュ動作を実行するために必要とされる命令の数が、一般的に必要とされる数千から、単一の命令(たとえば、読み出しまたは書き込み)に低減される。

20

30

【0039】

上述のように、あらゆるJavaScript(登録商標)オブジェクトは、ハッシュテーブルであるものと見なされ得、一般化されたハッシュテーブルプロセッサ(ハードウェア内に実装される)が、JavaScript(登録商標)コードを処理するために使用され得る。たとえば、各JavaScript(登録商標)オブジェクトは、現在の解決策より著しく少ない命令を用いて、プロセッサ510の一般化されたハッシュテーブル回路内で処理され得る。この低減された数の必要な命令によって、JavaScript(登録商標)アプリケーションの全体的な速度および効率が大幅に改善される。

【0040】

図5に戻ると、コンパイラハッシュモジュール516は、文字列から整数へのマッピングのプロセスを一度だけ実行し得、そのことで、後続のハッシュテーブルの更新およびルックアップをハードウェアによって単一の命令で実行することが可能になる。このマッピングによって、同様に、従来ジャストインタイムコンパイラコード内に埋め込まれるハッシュテーブルサポートのいくつかを、GET命令およびSET命令など、命令セットアーキテクチャ(ISA)拡張に移すことが可能になる。

40

【0041】

一態様では、プロセッサは、3つのレジスタパラメータで実行され得るGETおよびSET命令を処理するために3つのレジスタを有することができる。最初の2つのパラメータ(たとえば、r1、r2)がキーを計算するために使用され得、第3のパラメータが、GET命令に対して得られた値をコピーするため、およびSET命令から生じる値を記憶するために使用され

50

得る。以下の疑似コードは、3つのパラメータの使用法を示す：

```
get r1,r2,r3 // r3=HASH[key(r1,r2)]
set r1,r2,r3 // HASH[key(r1,r2)]=r3
```

【0042】

上記の疑似コードでは、最初の2つのレジスタ(パラメータr1およびr2に対応する)はソースレジスタであり、第3のレジスタ(パラメータr3に対応する)はディスティネーションレジスタである。2つのレジスタ(r1、r2)をキーとして使用することは、オブジェクト間のエントリの分離を容易にし、入れ子式ハッシュの処理を支援し、したがってJavaScript(登録商標)などの言語に好適となる。たとえば、JavaScript(登録商標)の場合、キーの一方の構成要素(たとえば、r1)がJavaScript(登録商標)オブジェクトにマップされ得、キーの他方の構成要素(たとえば、r2)がJavaScript(登録商標)オブジェクトのプロパティにマップされ得る。「キー」は、ハッシュテーブルキーをr1およびr2によって構成する数学的変換を表す。「HASH」は、計算されたキーに基づいて値ルックアップを実施する実際のハッシュテーブルインプリメンテーションである。

10

【0043】

一態様では、命令レジスタ506、508は、(ページテーブル変換に必要とされるスーパーユーザモードに反して)ユーザモードにおいてオペレーティングシステムにアクセス可能にされ得る。このことは、ユーザモードで動作している間に、オペレーティングシステム上で実行している任意のプロセスまたはアプリケーションが、上述のGETおよびSET命令を出すことを可能にする。一態様では、以下でより詳細に説明するように、ハードウェアハッシュテーブル516は、ハードウェアベースのハッシング機能の組合せが各プロセスに対してユーザモードにおいて利用可能であるように、ハッシュテーブル値のブロックをキャッシュすることができる。

20

【0044】

ソフトウェアプロセスがハッシュ動作(たとえば、ルックアップ)を要求するときはいつでも、コンピューティングデバイスプロセッサは、最初に、GET命令を介して要求されたハッシュ値の存在に対してハードウェア(すなわち、キャッシュメモリ)を調べることができる。値がハードウェア(すなわち、キャッシュメモリ)内にある場合、命令レジスタは、単に、要求された値を返すことができ、ハッシュ動作を1命令で計算することが可能になる。要求された項目がハードウェアハッシュテーブル518内にないことをGET命令が示した場合、ソフトウェアハッシュテーブル514ルックアッププロセスに対するコールがなされ得る。ルックアッププロセスは、ソフトウェア内(たとえば、ソフトウェアハッシュテーブル514上)で要求されたハッシュ動作を実施し、要求された値と今後アクセスされそうな周囲の値の両方を返すことができる。これらの値は、ハードウェアキャッシュ504にプッシュされ得、今後の処理のためにプロセッサ510に対して容易に利用できるように、ハードウェアハッシュテーブル518に(たとえば、SET命令を介して)記憶され得る。このようにして、様々な態様が、生成されたハッシュ情報のブロックをSET命令(たとえば、「set」レジスタ508上で出される)を介してプロセッサ510のキャッシュメモリにプッシュすることによって局所性の原理を利用する。この動作は、今後アクセスされそうな値が、後続の動作における(たとえば、単一の命令を介する)高速プロセッサアクセスのために、ハードウェアハッシュテーブル518のキャッシュ内に記憶されることを可能にする。

30

40

【0045】

ソフトウェアハッシュテーブル514を生成するために、コンパイラ512は、コードをコンパイルする間に、あらゆるストリングのリテラル値に一意IDを割り当てるグローバルストリングテーブルを作成することができる。キャッシュを有するのではなく、動的オブジェクトに対して、コンパイラは、JavaScript(登録商標)プロパティの値を保持するアレイを使用することができる。プロパティにアクセスするために、コンパイラは、プロパティ名のストリングIDをルックアップし、オブジェクトIDとプロパティストリングIDとの組合せをマップし、その組合せを、プロパティ値を表すオブジェクト内のフィールドアレイインデックスを表す整数にマップする、別のハッシュテーブルを使用することができる。以下

50

の疑似コードが、このプロセスを示す：

```
// z=o.x
ID1=StringtoIDHash["x"]; // first hash lookup
ID2=o.ID
ID3=IdToIndexHash[ID1,ID2] // second hash lookup
// Using the GET instruction

Z=o.Fields[ID3]
```

【 0 0 4 6 】

上記の疑似コードでは、第1のハッシュルックアップは、コンパイル時に一度だけ発生する。一方、第2のハッシュルックアップは、コードが実行されるごとに発生することができる。しかしながら、第2のハッシュルックアップはGET命令を使用するので、要求されたハッシュ情報がハードウェアハッシュテーブル内にあるとすれば、唯一の命令が実行される。

10

【 0 0 4 7 】

上述のように、ハッシュキーは、SET命令を介してハードウェアにプッシュされ得、それにより、今後アクセスされる可能性が高い値がハードウェア(すなわち、キャッシュ)内に記憶され得る。以下の疑似コードは、SET命令の使用法を示す：

```
// o.x=z
ID1=StringtoIDHash["x"];
ID2=o.ID
ID3=IdToIndexHash[ID1,ID2] // Using the GET instruction
IF(ID3==-1){
    ID3=o.Fields.Legth
    IdToIndexHash[ID1,ID2]=ID3; // Using the SET instruction
}
o.Fields[ID3]=z
```

20

【 0 0 4 8 】

図6は、ハードウェアとソフトウェアとの組合せでハッシュテーブル動作(たとえば、JavaScript(登録商標)動作)を実行するための一態様の方法600を示す。ブロック602の方法600では、コンパイラが、コンパイル時に、ソフトウェア内に記憶されるマスターハッシュテーブルを生成するために、あらゆるストリングリテラル値(たとえば、JavaScript(登録商標)オブジェクト)に一意IDを割り当てることによって、JavaScript(登録商標)ストリングキーを整数にマップすることができる。ブロック604で、ハッシュテーブル回路を有するハードウェアプロセッサの専門の命令レジスタ(ISA拡張)が、ユーザモードのソフトウェアに対して利用できるように初期化され得る。判定ブロック606で、プロセッサが、ハッシュテーブル動作(たとえば、ルックアップ)を要求するために、1つまたは複数のソフトウェアプロセスを待つことができる。ソフトウェアプロセスがハッシュ動作を要求する(すなわち、判定ブロック608=「Yes」)場合、ブロック608で、GET命令が、専門の命令レジスタ上のプロセッサに出され得る。ブロック610で、要求されたハッシュテーブル動作に対応する値がハードウェアハッシュテーブル内にあるかどうかを判断するために、ハードウェアハッシュテーブルがハッシュテーブル回路を使用してアクセスされ得る。判定ブロック612で、プロセッサは、ハードウェアハッシュテーブルが要求された値を含有するかどうかを判断することができる。要求された値がハードウェアハッシュテーブル内にある(すなわち、判定ブロック612=「Yes」)場合、ブロック614で、専門の命令レジスタは、ハッシュテーブル動作がハードウェアで実行されるように、要求された値を返すことができる。この動作は、唯一の命令を使用して実行され得、その結果がソフトウェアプロセスに提供され得る。要求された値がハードウェアハッシュテーブル内にない(すなわち、判定ブロック612=「No」)場合、ハードウェアは、その値がブロック616において利用できないことをソフトウェアプロセスに通知することができる。ブロック616で、ソフトウェアプロセスは、ハッシュテーブル動作がソフトウェアで実施されるように、マスターハッシ

30

40

50

ュテーブルから値を要求することができる。ブロック618で、ソフトウェアプロセスは、要求された値を含有するハッシュテーブル値のブロックをハードウェアハッシュテーブルにプッシュするために、専門の命令レジスタ上のSET命令を出すことができる。ブロック614または618で要求された値が返されると、プロセッサは、ハッシュテーブル動作(たとえば、ルックアップ)を要求するための1つまたは複数のソフトウェアプロセスを待つために、判定ブロック606に戻ることができる。

【0049】

図7は、態様のいずれかとともに使用するのに好適なコンピューティングデバイスのシステムブロック図である。典型的なコンピューティングデバイス700は、内部大容量メモリ702(たとえば、FLASHメモリ)と、ディスプレイ703と、スピーカ764とに結合されたプロセッサ701を含むことができる。加えて、コンピューティングデバイス700はまた、プロセッサ701に結合されたワイヤレスデータリンクおよび/またはセルラー電話トランシーバ705に接続され得る、電磁放射を送信および受信するためのアンテナ704を含むことができる。ハッシュテーブル動作をハードウェアで実行するための様々な態様の方法は、プロセッサ701およびメモリ702の部分によって実施され得る。代替として、コンピューティングデバイス700内部の、またはそれに結合された専用モジュールが、態様の方法を実施することができる。

10

【0050】

上記で説明した態様は、メモリマップされた入力/出力(MMIO)アクセラレータデバイスとコプロセッサとを使用する構成を含む多様なハードウェア構成内に実装され得る。たとえば、MMIOデバイスは、上記で説明したGETおよびSET命令に対応するLOADおよびSTORE動作を実施することができる。様々な態様では、ハードウェア支援の仮想マシナーキテクチャが、様々な特徴を実装するために使用され得る。

20

【0051】

上記で説明した態様はまた、図8に示すラップトップコンピュータ800など、多様なコンピューティングデバイス内に実装され得る。ラップトップコンピュータ800は通常、揮発性メモリ802と、ディスクドライブ803のような大容量の不揮発性メモリとに結合されたプロセッサ801を含む。コンピュータ800はまた、プロセッサ801に結合された、フロッピー(登録商標)ディスクドライブ804およびコンパクトディスク(CD)ドライブ805も含み得る。コンピュータデバイス800はまた、USBもしくはFireWire(登録商標)コネクタソケット、または、プロセッサ801をネットワークに結合するための他のネットワーク接続回路806のような、データ接続を確立するための、または外部メモリデバイスを受け入れるための、プロセッサ801に結合されたいくつかのコンネクタポートを含み得る。ノートブック構成において、コンピュータハウジングは、すべてがプロセッサ801に結合される、タッチパッド807、キーボード808、およびディスプレイ809を含む。コンピューティングデバイスの他の構成は、周知であるように、(たとえば、USB入力を介して)プロセッサに結合されたコンピュータマウスまたはトラックボールを含み得る。

30

【0052】

プロセッサ701、801は、以下で説明する様々な態様の機能を含む、様々な機能を実施するためのソフトウェア命令(アプリケーション)によって構成され得る、任意のプログラマブルマイクロプロセッサ、マイクロコンピュータ、または1つもしくは複数の多重プロセッサチップであり得る。いくつかのモバイル受信機デバイスでは、1つのプロセッサがワイヤレス通信機能に専用され、1つのプロセッサが他のアプリケーションの実行に専用されるなど、複数のプロセッサ701が設けられてよい。一般的には、ソフトウェアアプリケーションは、それらがアクセスされ、プロセッサ701、801にロードされる前に、内部メモリ702、802、803内に記憶され得る。様々な態様では、プロセッサ701、801はまた、アプリケーションソフトウェア命令を記憶するのに十分な内部メモリ、ならびにハッシュテーブルまたはハッシュテーブル値のブロックを記憶するためのキャッシュメモリを含む。

40

【0053】

上記の方法の説明およびプロセス流れ図は、単に説明のための例として提供したもので

50

あり、様々な態様のステップを提示された順序で実施しなければならないことを要求または暗示するものではない。当業者が諒解するように、上記の態様におけるステップの順序は、どのような順序で実施されてもよい。「その後」、「次いで」、「次に」などの単語は、ステップの順序を限定するものではなく、これらの単語は、単に、方法の説明において読者を案内するために使用される。さらに、たとえば、冠詞「1つの(a, an)」または「その(the)」などの使用による単数形での請求要素への言及は、その要素を単数に限定するものとして解釈されるべきではない。

【0054】

本明細書で開示された態様に関して説明された様々な例示的な論理ブロック、モジュール、回路、およびアルゴリズムステップは、電子ハードウェア、コンピュータソフトウェア、または両方の組合せとして実装され得る。ハードウェアとソフトウェアのこの互換性を明確に示すために、様々な例示的な構成要素、ブロック、モジュール、回路、およびステップを、上記では概してそれらの機能に関して説明した。そのような機能をハードウェアとして実装するか、ソフトウェアとして実装するかは、特定の適用例および全体的なシステムに課される設計制約に依存する。当業者なら、説明した機能を特定の適用例ごとに様々な方法で実装し得るが、そのような実装の決定は、本特許請求の範囲からの逸脱と解釈すべきではない。

10

【0055】

本明細書に開示された態様に関して記載された様々な例示的な論理、論理ブロック、モジュール、および回路を実装するために使用されるハードウェアは、汎用プロセッサ、デジタル信号プロセッサ(DSP)、マルチメディア放送受信機チップ内のDSP、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)もしくは他のプログラマブル論理デバイス、個別ゲートもしくはトランジスタ論理、個別のハードウェアコンポーネント、または、本明細書に記載された機能を実施するように設計されたそれらの任意の組合せで、実装または実施され得る。汎用プロセッサはマイクロプロセッサであり得るが、代替として、プロセッサは任意の従来プロセッサ、コントローラ、マイクロコントローラ、または状態機械であり得る。プロセッサはまた、コンピューティングデバイスの組合せ、たとえば、DSPとマイクロプロセッサとの組合せ、複数のマイクロプロセッサ、DSPコアと連携する1つまたは複数のマイクロプロセッサ、あるいは任意の他のそのような構成として実装され得る。代替的に、いくつかのステップまたは方法が、所与の機能に固有の回路によって実施され得る。

20

30

【0056】

1つまたは複数の例示的な態様では、説明した機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの組合せに実装することができる。ソフトウェアに実装される場合、機能は、1つまたは複数の命令もしくはコードとしてコンピュータ可読媒体上に記憶され、または、コンピュータ可読媒体を介して送信され得る。本明細書で開示する方法またはアルゴリズムのステップは、非一時的コンピュータ可読媒体またはプロセッサ可読媒体上に常駐し得る、実行されるプロセッサ実行可能なソフトウェアモジュールで実施され得る。非一時的コンピュータ可読媒体は、コンピューティングデバイスのコンピュータまたはプロセッサによってアクセスされ得る任意の物理的(すなわち、非一時的)記憶媒体を含む。限定ではなく例として、そのような非一時的コンピュータ可読媒体は、RAM、ROM、EEPROM、CD-ROMもしくは他の光ディスク記憶装置、磁気ディスク記憶装置もしくは他の磁気記憶デバイス、または、命令もしくはデータ構造の形式で所望のプログラムコードを搬送もしくは記憶するために使用され得るとともに、コンピューティングデバイスのコンピュータまたはプロセッサによってアクセスされ得る任意の他の媒体を含み得る。本明細書で使用する場合、ディスク(disk)およびディスク(disc)は、コンパクトディスク(CD)、レーザーディスク(登録商標)、光ディスク、デジタル多用途ディスク(DVD)、フロッピー(登録商標)ディスク、およびブルーレイディスクを含み、ディスク(disk)は、通常、磁氣的にデータを再生し、ディスク(disc)は、レーザーで光学的にデータを再生する。上記の組合せも非一時的コンピュータ可読媒体の範囲内に含めるべきである。加えて、方

40

50

法またはアルゴリズムの動作は、コンピュータプログラム製品に組み込まれ得る、非一時的なプロセッサ可読媒体および/またはコンピュータ可読媒体上に記憶されたコードおよび/または命令の、1つまたは任意の組合せ、またはそのセットとして常駐し得る。

【0057】

開示された態様の上記の説明は、当業者が本発明を製作または使用できるように提供されたものである。これらの態様への様々な修正が当業者には容易に明らかであり、本明細書で定義される一般的な原理は、本発明の趣旨または範囲を逸脱することなく他の態様に適用され得る。したがって、本発明は、本明細書で示された態様に限定されるものではなく、以下の特許請求の範囲ならびに本明細書で開示された原理および新規の特徴に合致する、最も広い範囲を与えられるべきである。

10

【符号の説明】

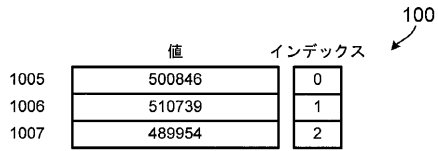
【0058】

- 500 コンピューティングデバイスアーキテクチャ
- 502 オペレーティングシステム
- 504 ハードウェア
- 506 「get」機械語命令レジスタ
- 508 「set」機械語命令レジスタ
- 510 専門のプロセッサ
- 512 コンパイラ
- 514 ソフトウェアハッシュテーブル
- 516 コンパイラハッシュモジュール
- 518 ハードウェアハッシュテーブル
- 520 CPU
- 522 算術論理ユニット(ALU)
- 524 浮動小数点ユニット(FPU)
- 528 ハードウェアハッシュテーブルインプリメンテーションユニット
- 530 ハードウェアアーキテクチャ
- 560 ハードウェアアーキテクチャ
- 590 ハードウェアアーキテクチャ
- 592 入力/出力(MMIO)アクセラレータデバイス

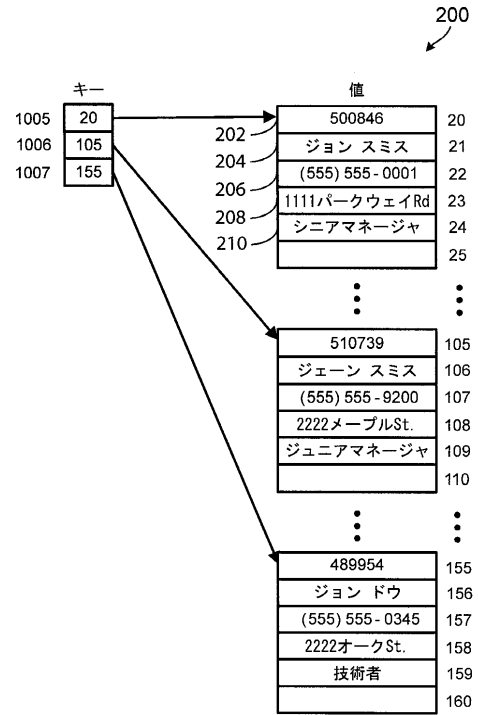
20

30

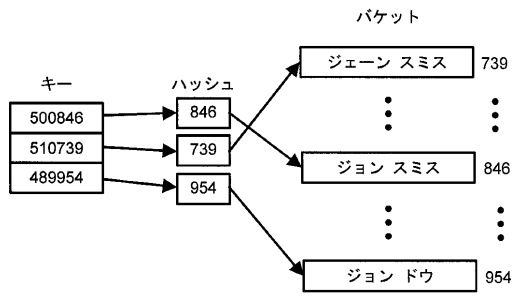
【 図 1 】



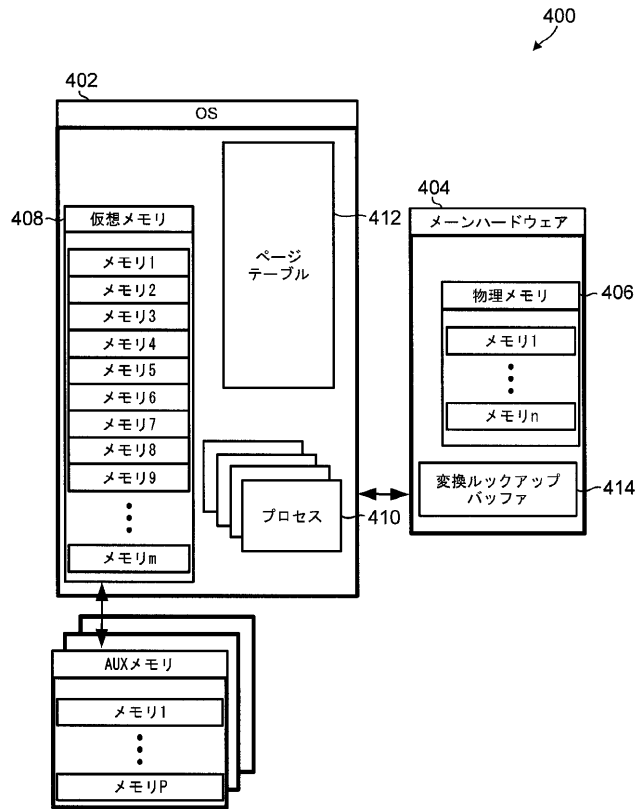
【 図 2 】



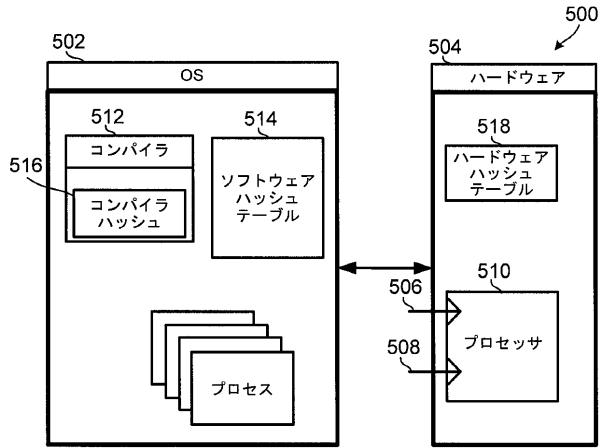
【 図 3 】



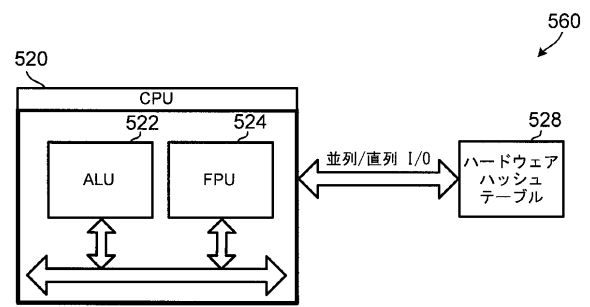
【 図 4 】



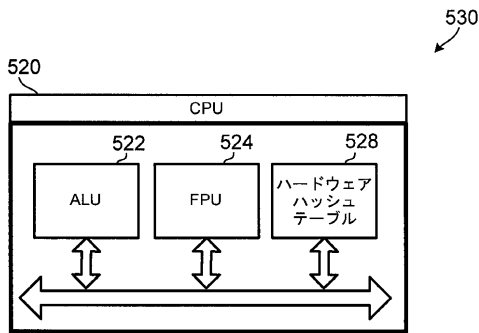
【図 5 A】



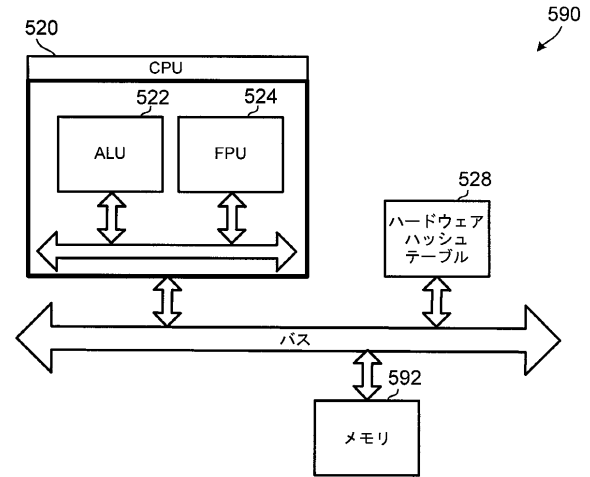
【図 5 C】



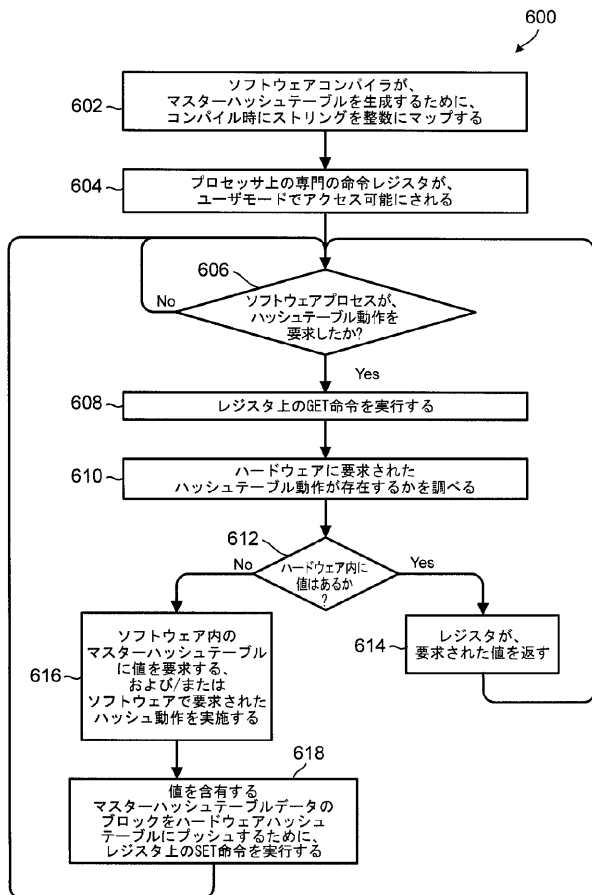
【図 5 B】



【図 5 D】



【図 6】



【図 7】

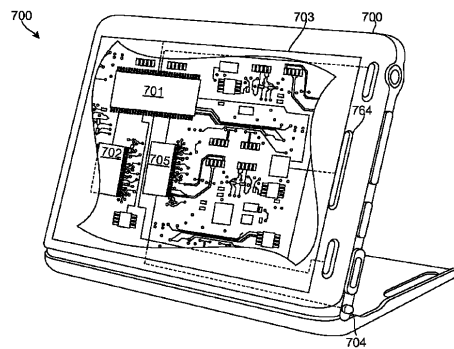


FIG. 7

【 図 8 】

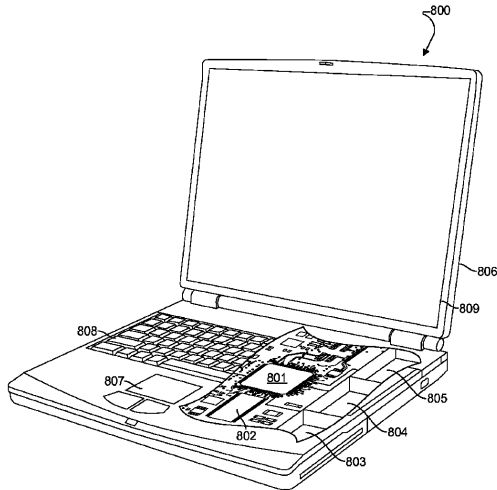


FIG. 8

【 手続補正書 】

【 提出日 】平成25年11月27日(2013.11.27)

【 手続補正 1 】

【 補正対象書類名 】特許請求の範囲

【 補正対象項目名 】全文

【 補正方法 】変更

【 補正の内容 】

【 特許請求の範囲 】

【 請求項 1 】

動的ソフトウェア言語コードを実行するための方法であって、
 ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように
 前記動的ソフトウェア言語コードをコンパイルするステップと、
 前記符号化されたハッシュテーブル整数キーにマップされた値を、メモリ内に保持され
るソフトウェアハッシュテーブルに記憶するステップと、
 前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部を、第1の
 専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテー
 ションにロードするステップと
 を含む、

動的ソフトウェア言語が、マネージドランタイムで実行され、ランタイムにおいて解釈
 され、またはランタイムにおいてコンパイルされるプログラムを書くために使用される任
 意の言語である、方法。

【 請求項 2 】

前記第1の専用のハードウェア命令が、前記専用のハードウェアハッシュテーブルイン
 プリメンテーションに値をプッシュするSET命令である、請求項1に記載の方法。

【 請求項 3 】

ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、ハードウェアプロセッサ上で専用のハードウェア命令を実行することによって、ハッシュテーブル値が前記ハードウェアプロセッサによって取り出され得るかどうかを判断するステップと、

前記ハッシュテーブル値が前記ハードウェアプロセッサに記憶されている場合、前記ハッシュテーブル値を返すためにハードウェアベースのハッシュテーブルルックアップを実行するステップと

をさらに含む、請求項2に記載の方法。

【請求項4】

前記ハッシュテーブル値が前記ハードウェアプロセッサ上にない場合、前記ソフトウェアハッシュテーブル内に記憶されている、前記ハッシュテーブル値と前記符号化されたハッシュテーブル整数キーにマップされた前記値の別の一部分とを取り出すために、ソフトウェアベースのハッシュテーブルルックアップを実行するステップと、

前記符号化されたハッシュテーブル整数キーにマップされた前記値の前記別の一部分を、第2の専用のハードウェア命令を介して前記ハードウェアプロセッサの専用メモリにロードすることによって、前記専用のハードウェアハッシュテーブルインプリメンテーションを更新するステップと

をさらに含む、請求項3に記載の方法。

【請求項5】

前記第2の専用のハードウェア命令が、前記ハードウェアプロセッサの前記専用メモリから値を取り出すGET命令である、請求項4に記載の方法。

【請求項6】

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップがコンパイル時に厳密に一度だけ遂行され、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分をロードするステップが、動的ソフトウェア言語動作の要求ごとに実行される、請求項5に記載の方法。

【請求項7】

前記動的ソフトウェア言語コードがJavaScript(登録商標)コードであり、

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップが、JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するようにJavaScript(登録商標)コードをコンパイルするステップを含む、請求項5に記載の方法。

【請求項8】

JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップがコンパイル時に厳密に一度だけ遂行され、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分をロードするステップが、JavaScript(登録商標)動作の要求ごとに実行される、請求項7に記載の方法。

【請求項9】

前記第1の専用のハードウェア命令が、メモリマップされたロケーションに値を記憶することによって、前記専用のハードウェアハッシュテーブルインプリメンテーションに前記値をプッシュするSTORE命令である、請求項1に記載の方法。

【請求項10】

前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップが、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、1次プロセッサに結合されたコプロセッサ上の専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップを含む、請求項1に記載の方法。

【請求項11】

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップがコンピューティングデバイ

スの第1のプロセッサによって実施され、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップが、前記コンピューティングデバイスの第2のプロセッサによって実施される、請求項1に記載の方法。

【請求項12】

前記コンピューティングデバイスの前記第2のプロセッサが専門のハッシュテーブルコプロセッサである、請求項11に記載の方法。

【請求項13】

ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、前記専門のハッシュテーブルコプロセッサ上で第2の専用のハードウェア命令を実行することによって、ハッシュテーブル値が取り出され得るかどうかを判断するステップと、

前記ハッシュテーブル値が取り出され得るものと判断される場合、前記コンピューティングデバイスの前記第1のプロセッサ上で前記ハッシュテーブル値を受信するために、前記専門のハッシュテーブルコプロセッサ上でハードウェアベースのハッシュテーブルルックアップを実行するステップと

をさらに含む、請求項12に記載の方法。

【請求項14】

メモリと、

前記メモリに結合されたプロセッサであって、

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように動的ソフトウェア言語コードをコンパイルするステップと、

前記符号化されたハッシュテーブル整数キーにマップされた値がソフトウェアによってアクセス可能となるように、前記符号化されたハッシュテーブル整数キーにマップされた前記値を前記メモリ内で保持されるソフトウェアハッシュテーブルに記憶するステップと、

前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップと

を含み、

動的ソフトウェア言語が、マネージドランタイムで実行される、ランタイムにおいて解釈される、またはランタイムにおいてコンパイルされるプログラムを書くために使用される任意の言語である

動作を実施するためのプロセッサ実行可能命令で構成される、プロセッサとを備える、コンピューティングデバイス。

【請求項15】

前記メモリが前記プロセッサ内のハッシュテーブルメモリである、請求項14に記載のコンピューティングデバイス。

【請求項16】

前記メモリが前記プロセッサの外部にある、請求項14に記載のコンピューティングデバイス。

【請求項17】

前記プロセッサが、前記第1の専用のハードウェア命令が前記専用のハードウェアハッシュテーブルインプリメンテーションに値をプッシュするSET命令であるような動作を実施するためのプロセッサ実行可能命令で構成される、請求項14に記載のコンピューティングデバイス。

【請求項18】

前記プロセッサが、

ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、ハードウェアプロセッサ上で専用のハードウェア命令を実行することによって、ハッシュテーブル値が前記ハードウェアプロセッサによって取り出され得るかどうかを判断するステップ

と、

前記ハッシュテーブル値が前記ハードウェアプロセッサに記憶されている場合、前記ハッシュテーブル値を返すためにハードウェアベースのハッシュテーブルルックアップを実行するステップと

をさらに含む動作を実施するためのプロセッサ実行可能命令で構成される、請求項17に記載のコンピューティングデバイス。

【請求項19】

前記プロセッサが、

前記ハッシュテーブル値が前記ハードウェアプロセッサ上にない場合、前記ソフトウェアハッシュテーブル内に記憶されている、前記ハッシュテーブル値と前記符号化されたハッシュテーブル整数キーにマップされた前記値の別の一部分とを取り出すために、ソフトウェアベースのハッシュテーブルルックアップを実行するステップと、

前記符号化されたハッシュテーブル整数キーにマップされた前記値の前記別の一部分を、第2の専用のハードウェア命令を介して前記ハードウェアプロセッサの専用メモリにロードすることによって、前記専用のハードウェアハッシュテーブルインプリメンテーションを更新するステップと

をさらに含む動作を実施するためのプロセッサ実行可能命令で構成される、請求項18に記載のコンピューティングデバイス。

【請求項20】

前記プロセッサが、前記第2の専用のハードウェア命令が前記ハードウェアプロセッサの前記専用メモリから値を取り出すGET命令であるような動作を実施するためのプロセッサ実行可能命令で構成される、請求項19に記載のコンピューティングデバイス。

【請求項21】

前記プロセッサが、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップがコンパイル時に厳密に一度だけ遂行され、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分をロードするステップが、動的ソフトウェア言語動作の要求ごとに実行されるような動作を実施するためのプロセッサ実行可能命令で構成される、請求項20に記載のコンピューティングデバイス。

【請求項22】

前記プロセッサが、

前記動的ソフトウェア言語コードがJavaScript(登録商標)コードであり、

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップが、JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するようにJavaScript(登録商標)コードをコンパイルするステップを含む

ような動作を実施するためのプロセッサ実行可能命令で構成される、請求項20に記載のコンピューティングデバイス。

【請求項23】

前記プロセッサが、JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップがコンパイル時に厳密に一度だけ遂行され、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分をロードするステップが、JavaScript(登録商標)動作の要求ごとに実行されるような動作を実施するためのプロセッサ実行可能命令で構成される、請求項22に記載のコンピューティングデバイス。

【請求項24】

前記プロセッサが、前記第1の専用のハードウェア命令がメモリマップされたロケーションに値を記憶することによって前記専用のハードウェアハッシュテーブルインプリメンテーションに前記値をプッシュするSTORE命令であるような動作を実施するためのプロセッサ実行可能命令で構成される、請求項14に記載のコンピューティングデバイス。

【請求項25】

前記プロセッサが、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップが、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、1次プロセッサに結合されたコプロセッサ上の専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップを含むような動作を実施するためのプロセッサ実行可能命令で構成される、請求項14に記載のコンピューティングデバイス。

【請求項26】

前記プロセッサが第1のプロセッサと第2のプロセッサとを備え、前記プロセッサが、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップが前記第1のプロセッサによって実施され、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップが前記コンピューティングデバイスの前記第2のプロセッサによって実施されるような動作を実施するためのプロセッサ実行可能命令で構成される、請求項14に記載のコンピューティングデバイス。

【請求項27】

前記第2のプロセッサが専門のハッシュテーブルコプロセッサである、請求項26に記載のコンピューティングデバイス。

【請求項28】

前記プロセッサが、
ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、前記専門のハッシュテーブルコプロセッサ上で第2の専用のハードウェア命令を実行することによって、ハッシュテーブル値が取り出され得るかどうかを判断するステップと、
前記ハッシュテーブル値が取り出され得るものと判断される場合、前記コンピューティングデバイスの前記第1のプロセッサ上で前記ハッシュテーブル値を受信するために、前記専門のハッシュテーブルコプロセッサ上でハードウェアベースのハッシュテーブルルックアップを実行するステップと
をさらに含む動作を実施するためのプロセッサ実行可能命令で構成される、請求項27に記載のコンピューティングデバイス。

【請求項29】

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように動的ソフトウェア言語コードをコンパイルするための手段と、
前記符号化されたハッシュテーブル整数キーにマップされた値を、メモリ内に保持されるソフトウェアハッシュテーブルに記憶するための手段と、
前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段と
を含み、
動的ソフトウェア言語が、マネージドランタイムで実行される、ランタイムにおいて解釈される、またはランタイムにおいてコンパイルされるプログラムを書くために使用される任意の言語である、コンピューティングデバイス。

【請求項30】

前記メモリが前記コンピューティングデバイスのプロセッサ内のハッシュテーブルメモリである、請求項29に記載のコンピューティングデバイス。

【請求項31】

前記メモリが前記コンピューティングデバイスのプロセッサの外部にある、請求項29に記載のコンピューティングデバイス。

【請求項32】

前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーシ

ョンにロードするための手段が、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、専用のハードウェアハッシュテーブルインプリメンテーションに前記値をプッシュするSET命令を介して前記専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段を含む、請求項29に記載のコンピューティングデバイス。

【請求項33】

ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、ハードウェアプロセッサ上で専用のハードウェア命令を実行することによって、ハッシュテーブル値が前記ハードウェアプロセッサによって取り出され得るかどうかを判断するための手段と、

前記ハッシュテーブル値が前記ハードウェアプロセッサに記憶されている場合、前記ハッシュテーブル値を返すためにハードウェアベースのハッシュテーブルルックアップを実行するための手段と

をさらに含む、請求項32に記載のコンピューティングデバイス。

【請求項34】

前記ハッシュテーブル値が前記ハードウェアプロセッサ上にない場合、前記ソフトウェアハッシュテーブル内に記憶されている、前記ハッシュテーブル値と前記符号化されたハッシュテーブル整数キーにマップされた前記値の別の一部分とを取り出すために、ソフトウェアベースのハッシュテーブルルックアップを実行するための手段と、

前記符号化されたハッシュテーブル整数キーにマップされた前記値の前記別の一部分を、第2の専用のハードウェア命令を介して前記ハードウェアプロセッサの専用メモリにロードすることによって、前記専用のハードウェアハッシュテーブルインプリメンテーションを更新するための手段と

をさらに含む、請求項33に記載のコンピューティングデバイス。

【請求項35】

前記符号化されたハッシュテーブル整数キーにマップされた前記値の前記別の一部分を第2の専用のハードウェア命令を介して前記ハードウェアプロセッサの専用メモリにロードすることによって前記専用のハードウェアハッシュテーブルインプリメンテーションを更新するための手段が、前記符号化されたハッシュテーブル整数キーにマップされた前記値の前記別の一部分を前記ハードウェアプロセッサの専用メモリから値を取り出すGET命令を介して前記ハードウェアプロセッサの前記専用メモリにロードすることによって前記専用のハードウェアハッシュテーブルインプリメンテーションを更新するための手段を含む、請求項34に記載のコンピューティングデバイス。

【請求項36】

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするための手段が、コンパイル時に厳密に一度だけハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするための手段を含み、

前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段が、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、動的ソフトウェア言語動作の要求ごとに第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段を含む、請求項35に記載のコンピューティングデバイス。

【請求項37】

前記動的ソフトウェア言語コードがJavaScript(登録商標)コードであり、

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするための手段が、JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するようにJavaScript(登録商標)コードをコンパイルするための手段を含む、

請求項35に記載のコンピューティングデバイス。

【請求項 38】

JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するための手段が、コンパイル時に厳密に一度だけJavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するための手段を含み、

前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段が、JavaScript(登録商標)動作の要求ごとに、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段を含む、請求項37に記載のコンピューティングデバイス。

【請求項 39】

前記第1の専用のハードウェア命令が、メモリマップされたロケーションに値を記憶することによって、前記専用のハードウェアハッシュテーブルインプリメンテーションに前記値をプッシュするSTORE命令である、請求項29に記載のコンピューティングデバイス。

【請求項 40】

前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段が、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、1次プロセッサに結合されたコプロセッサ上の専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段を含む、請求項29に記載のコンピューティングデバイス。

【請求項 41】

第1のプロセッサと、

第2のプロセッサとを備え、

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするための手段が、前記第1のプロセッサによってハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするための手段を含み、

前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段が、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、前記第2のプロセッサによって第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするための手段を含む、請求項29に記載のコンピューティングデバイス。

【請求項 42】

前記第2のプロセッサが専門のハッシュテーブルコプロセッサである、請求項41に記載のコンピューティングデバイス。

【請求項 43】

ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、前記専門のハッシュテーブルコプロセッサ上で第2の専用のハードウェア命令を実行することによって、ハッシュテーブル値が取り出され得るかどうかを判断するための手段と、

前記ハッシュテーブル値が取り出され得るものと判断される場合、前記コンピューティングデバイスの前記第1のプロセッサ上で前記ハッシュテーブル値を受信するために、前記専門のハッシュテーブルコプロセッサ上でハードウェアベースのハッシュテーブルルックアップを実行するための手段と

をさらに含む、請求項42に記載のコンピューティングデバイス。

【請求項 44】

プロセッサ実行可能命令を記憶した非一時的プロセッサ可読媒体であって、前記プロセッサ実行可能命令がプロセッサに、

ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように動的ソフトウェア言語コードをコンパイルするステップと、

前記符号化されたハッシュテーブル整数キーにマップされた値を、メモリ内に保持されるソフトウェアハッシュテーブルに記憶するステップと、

前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップと

を含む動作を実行させるように構成され、

動的ソフトウェア言語が、マネージドランタイムで実行される、ランタイムにおいて解釈される、またはランタイムにおいてコンパイルされるプログラムを書くために使用される任意の言語である、非一時的プロセッサ可読媒体。

【請求項45】

前記記憶されたプロセッサ実行可能命令がプロセッサ上で機能するように構成され、前記メモリが前記プロセッサ内のハッシュテーブルメモリである、請求項44に記載の非一時的プロセッサ可読媒体。

【請求項46】

前記記憶されたプロセッサ実行可能命令がプロセッサ上で機能するように構成され、前記メモリが前記プロセッサの外部にある、請求項44に記載の非一時的プロセッサ可読媒体。

【請求項47】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、前記第1の専用のハードウェア命令が前記専用のハードウェアハッシュテーブルインプリメンテーションに値をプッシュするSET命令であるような動作を実施させるように構成される、請求項44に記載の非一時的プロセッサ可読媒体。

【請求項48】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、ハードウェアプロセッサ上で専用のハードウェア命令を実行することによって、ハッシュテーブル値が前記ハードウェアプロセッサによって取り出され得るかどうかを判断するステップと、

前記ハッシュテーブル値が前記ハードウェアプロセッサに記憶されている場合、前記ハッシュテーブル値を返すためにハードウェアベースのハッシュテーブルルックアップを実行するステップと

をさらに含む動作を実施させるように構成される、請求項47に記載の非一時的プロセッサ可読媒体。

【請求項49】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、前記ハッシュテーブル値が前記ハードウェアプロセッサ上にない場合、前記ソフトウェアハッシュテーブル内に記憶されている、前記ハッシュテーブル値と前記符号化されたハッシュテーブル整数キーにマップされた前記値の別の一部分とを取り出すために、ソフトウェアベースのハッシュテーブルルックアップを実行するステップと、

前記符号化されたハッシュテーブル整数キーにマップされた前記値の前記別の一部分を、第2の専用のハードウェア命令を介して前記ハードウェアプロセッサの専用メモリにロードすることによって、前記専用のハードウェアハッシュテーブルインプリメンテーションを更新するステップと

をさらに含む動作を実施させるように構成される、請求項48に記載の非一時的プロセッサ可読媒体。

【請求項50】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、前記第2の専用のハードウェア命令が前記ハードウェアプロセッサの前記専用メモリから値を取り出すGET命令であ

るような動作を実施させるように構成される、請求項49に記載の非一時的プロセッサ可読媒体。

【請求項51】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップがコンパイル時に厳密に一度だけ遂行され、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分をロードするステップが動的ソフトウェア言語動作の要求ごとに行われるような動作を実施させるように構成される、請求項50に記載の非一時的プロセッサ可読媒体。

【請求項52】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、前記動的ソフトウェア言語コードがJavaScript(登録商標)コードであり、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップが、JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するようにJavaScript(登録商標)コードをコンパイルするステップを含むような動作を実施させるように構成される、請求項50に記載の非一時的プロセッサ可読媒体。

【請求項53】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、JavaScript(登録商標)ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するステップがコンパイル時に厳密に一度だけ遂行され、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分をロードするステップがJavaScript(登録商標)動作の要求ごとに行われるような動作を実施させるように構成される、請求項52に記載の非一時的プロセッサ可読媒体。

【請求項54】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、前記第1の専用のハードウェア命令が、メモリマップされたロケーションに値を記憶することによって前記専用のハードウェアハッシュテーブルインプリメンテーションに前記値をプッシュするSTORE命令であるような動作を実施させるように構成される、請求項44に記載の非一時的プロセッサ可読媒体。

【請求項55】

前記記憶されたプロセッサ実行可能命令が、プロセッサに、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップが、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、1次プロセッサに結合されたコプロセッサ上の専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップを含むような動作を実施させるように構成される、請求項44に記載の非一時的プロセッサ可読媒体。

【請求項56】

前記記憶されたプロセッサ実行可能命令が第1のプロセッサと第2のプロセッサとを備えるプロセッサによって実行可能であるように構成され、前記記憶されたプロセッサ実行可能命令が、前記プロセッサに、ハッシュテーブルストリングキーをハッシュテーブル整数キーとして符号化するように前記動的ソフトウェア言語コードをコンパイルするステップが前記第1のプロセッサによって実施され、前記符号化されたハッシュテーブル整数キーにマップされた前記値の一部分を、第1の専用のハードウェア命令を介して専用のハードウェアハッシュテーブルインプリメンテーションにロードするステップがコンピューティングデバイスの前記第2のプロセッサによって実施されるような動作を実施させるように構成される、請求項44に記載の非一時的プロセッサ可読媒体。

【請求項57】

前記第2のプロセッサが専門のハッシュテーブルコプロセッサである、請求項56に記載

の非一時的プロセッサ可読媒体。

【請求項58】

前記記憶されたプロセッサ実行可能命令が、前記プロセッサに、ハッシュテーブルルックアップを必要とする動的ソフトウェア言語動作ごとに、前記専門のハッシュテーブルコプロセッサ上で第2の専用のハードウェア命令を実行することによって、ハッシュテーブル値が取り出され得るかどうかを判断するステップと、

前記ハッシュテーブル値が取り出され得るものと判断される場合、前記コンピューティングデバイスの前記第1のプロセッサ上で前記ハッシュテーブル値を受信するために、前記専門のハッシュテーブルコプロセッサ上でハードウェアベースのハッシュテーブルルックアップを実行するステップと

をさらに含む動作を実施させるように構成される、請求項57に記載の非一時的プロセッサ可読媒体。

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT

International application No PCT/US2012/038797

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F9/44 ADD.		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EP0-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2002/108107 A1 (DARNELL STEPHEN [GB] ET AL) 8 August 2002 (2002-08-08) paragraph [0509] paragraph [0526]	1-58
Y	US 2007/113222 A1 (DIGNUM MARCELINO M [US] ET AL) 17 May 2007 (2007-05-17) paragraph [0053] paragraph [0100]	1-58
<input type="checkbox"/> Further documents are listed in the continuation of Box C.		
<input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents :		
"A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed		"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family
Date of the actual completion of the international search 19 July 2012		Date of mailing of the international search report 30/07/2012
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Ebert, Werner

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2012/038797

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2002108107 A1	08-08-2002	AU 2846999 A	05-06-2000
		EP 1208425 A2	29-05-2002
		JP 4783471 B2	28-09-2011
		JP 2003526135 A	02-09-2003
		JP 2010267291 A	25-11-2010
		US 2002029357 A1	07-03-2002
		US 2002032719 A1	14-03-2002
		US 2002032822 A1	14-03-2002
		US 2002040470 A1	04-04-2002
		US 2002042807 A1	11-04-2002
		US 2002049865 A1	25-04-2002
		US 2002104077 A1	01-08-2002
		US 2002108106 A1	08-08-2002
		US 2002108107 A1	08-08-2002
		US 2002112227 A1	15-08-2002
		US 2002165848 A1	07-11-2002
		US 2008016507 A1	17-01-2008
		WO 0029937 A2	25-05-2000

US 2007113222 A1	17-05-2007	US 2007113222 A1	17-05-2007
		WO 2007058959 A2	24-05-2007

フロントページの続き

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

(72)発明者 モハンマド・エイチ・レシャディ

アメリカ合衆国・カリフォルニア・92121・サン・ディエゴ・モアハウス・ドライブ・5775

(72)発明者 トーマス・サルトリウス

アメリカ合衆国・カリフォルニア・92121・サン・ディエゴ・モアハウス・ドライブ・5775

Fターム(参考) 5B013 DD03

5B033 BE03

5B081 AA09 CC13

5B376 FA10 FA25