

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3835764号
(P3835764)

(45) 発行日 平成18年10月18日(2006.10.18)

(24) 登録日 平成18年8月4日(2006.8.4)

(51) Int. Cl.

G06F 9/38 (2006.01)

F I

G06F 9/38 310A

請求項の数 30 (全 33 頁)

(21) 出願番号 特願2005-75625 (P2005-75625)
 (22) 出願日 平成17年3月16日(2005.3.16)
 (62) 分割の表示 特願平10-250754の分割
 原出願日 平成10年9月4日(1998.9.4)
 (65) 公開番号 特開2005-251210 (P2005-251210A)
 (43) 公開日 平成17年9月15日(2005.9.15)
 審査請求日 平成17年9月2日(2005.9.2)

(73) 特許権者 000005821
 松下電器産業株式会社
 大阪府門真市大字門真1006番地
 (74) 代理人 100109210
 弁理士 新居 広守
 (72) 発明者 鈴木 正人
 大阪府門真市大字門真1006番地 松下
 電器産業株式会社内

審査官 後藤 彰

最終頁に続く

(54) 【発明の名称】 プロセッサおよび記録媒体

(57) 【特許請求の範囲】

【請求項1】

プロセッサであって、命令に含まれる操作記述を一旦蓄積するための蓄積バッファと、蓄積バッファの蓄積および出力を制御する制御部と、命令を解読および実行する解読実行部とを備え、前記制御部は、フェッチされた命令に含まれる蓄積用の操作記述と他の操作記述とを判別し、判別された蓄積用の操作記述を前記蓄積バッファに書き込み、前記解読実行部への前記他の操作記述の出力と、前記解読実行部への前記蓄積バッファからの操作記述の出力とを制御することを特徴とするプロセッサ。

【請求項2】

前記蓄積用の操作記述は、1つの命令から分割された分割部分であり、前記制御部は、1つの命令を構成する複数の前記分割部分を前記蓄積バッファに蓄積することを特徴とする請求項1記載のプロセッサ。

【請求項3】

前記制御部は、前記蓄積バッファに蓄積された前記複数の分割部分から構成される少なくとも1つの命令と、フェッチされた他の命令とを前記解読実行部へ出力し、

10

20

前記前記解読実行部は、前記複数の分割部分から構成される少なくとも１つの命令と、前記他の命令とを並列に実行する

ことを特徴とする請求項２記載のプロセッサ。

【請求項４】

プロセッサであって、

命令に含まれる操作記述を一旦蓄積するための蓄積バッファと、

蓄積バッファの蓄積および出力を制御する制御部と、

複数の操作記述を並列に解読および実行する解読実行部とを備え、

前記制御部は、

フェッチされた命令に含まれる蓄積用の操作記述と他の操作記述とを判別し、

判別された蓄積用の操作記述を前記蓄積バッファに書き込み、

前記解読実行部への前記他の操作記述の出力と、前記解読実行部への前記蓄積バッファからの操作記述の出力とを制御する

ことを特徴とするプロセッサ。

【請求項５】

前記蓄積バッファは、前記解読実行部における並列数と同数の操作記述を蓄積する領域を有し、

前記制御部は、蓄積バッファに蓄積された操作記述を前記解読実行部に並列に出力することを特徴とする請求項４記載のプロセッサ。

【請求項６】

前記操作記述は、蓄積用か否かを示す蓄積ビットを有し、

前記制御部は、判別された蓄積用の操作記述が埋め込まれた命令中の位置に対応する蓄積バッファ中の位置の領域に、当該蓄積用の操作記述を書き込む

ことを特徴とする請求項５記載のプロセッサ。

【請求項７】

前記蓄積用の操作記述は、命令中の位置を示す位置ビットを有し、

前記蓄積バッファは、前記位置ビットに従う蓄積バッファ中の領域に、当該蓄積用の操作記述を蓄積する

ことを特徴とする請求項５記載のプロセッサ。

【請求項８】

前記制御部は、前記解読実行部の並列数と同数の操作記述を蓄積完了したことに起因して、蓄積バッファに蓄積された操作記述を前記解読実行部に並列に出力する

ことを特徴とする請求項５記載のプロセッサ。

【請求項９】

前記蓄積バッファに蓄積された操作記述は、前記解読実行部に出力されることに起因して消去される

ことを特徴とする請求項５記載のプロセッサ。

【請求項１０】

前記蓄積バッファに蓄積された操作記述は、前記解読実行部に複数回出力される

ことを特徴とする請求項５記載のプロセッサ。

【請求項１１】

前記蓄積バッファは、さらに、蓄積された操作記述を消去せずに実行するか否かを示すビットを保持し、

前記蓄積バッファに蓄積された操作記述は、前記解読実行部に出力されたとき、前記ビットに従って消去されないかまたは消去される

ことを特徴とする請求項５記載のプロセッサ。

【請求項１２】

フェッチされた命令は、前記解読実行部における並列数よりも少ない数の操作記述からなる命令で構成される

ことを特徴とする請求項５記載のプロセッサ。

10

20

30

40

50

【請求項 1 3】

前記蓄積バッファ中の領域は格納順位を有し、格納順位に従って蓄積用の操作記述を順に前記蓄積バッファ中の各領域に蓄積する

ことを特徴とする請求項 1 2 記載のプロセッサ。

【請求項 1 4】

前記制御部は、フェッチされた命令中の他の操作記述と、蓄積バッファに蓄積された蓄積用の操作記述とを、解読実行部に並列に出力する

ことを特徴とする請求項 4 記載のプロセッサ。

【請求項 1 5】

前記蓄積バッファは、複数段のバッファ領域を有し、

前記制御部は、各バッファ領域に蓄積された蓄積用の操作記述を順に出力する

ことを特徴とする請求項 1 4 記載のプロセッサ。

10

【請求項 1 6】

プロセッサに実行される命令を含むプログラムを記録する記録媒体であって、

前記プロセッサに蓄積用の操作記述と他の操作記述とを含む命令をフェッチさせ、

前記命令に含まれる前記蓄積用の操作記述を前記プロセッサ内の前記蓄積バッファに蓄積させるとともに

前記他の操作記述を前記プロセッサに実行させた後、前記蓄積バッファの操作記述を前記プロセッサに実行させる

プログラムを記録したことを特徴とする記録媒体。

20

【請求項 1 7】

前記蓄積用の操作記述は、1つの命令から分割された分割部分である

プログラムを記録したことを特徴とする請求項 1 6 記載の記録媒体。

【請求項 1 8】

前記蓄積バッファに蓄積された複数の前記分割部分から構成される少なくとも1つの命令と、フェッチされた他の命令とをプロセッサに並列に実行させる

ことを特徴とする請求項 1 7 記載の記録媒体。

【請求項 1 9】

プロセッサに並列実行される複数の操作記述からなる命令を含むプログラムを記録する記録媒体であって、

前記プロセッサに蓄積用の操作記述と他の操作記述とを含む命令をフェッチさせ、

前記命令に含まれる前記蓄積用の操作記述を前記プロセッサ内の前記蓄積バッファに蓄積させるとともに

前記他の操作記述を前記プロセッサに実行させた後、前記蓄積バッファの操作記述を前記プロセッサに実行させる

プログラムを記録したことを特徴とする記録媒体。

30

【請求項 2 0】

前記プロセッサの並列数と同数の操作記述を蓄積バッファに蓄積させる

プログラムを記録したことを特徴とする請求項 1 8 記載の記録媒体。

【請求項 2 1】

前記操作記述は、蓄積用か否かを示す蓄積ビットを有し、

蓄積用の操作記述が埋め込まれた命令中の位置に対応する蓄積バッファ中の位置の領域に、当該蓄積用の操作記述を書き込ませる

プログラムを記録したことを特徴とする請求項 1 9 記載の記録媒体。

40

【請求項 2 2】

前記蓄積用の操作記述は、命令中の位置を示す位置ビットを有し、

前記位置ビットに従う蓄積バッファ中の領域に蓄積用の操作記述を蓄積させる

プログラムを記録したことを特徴とする請求項 1 9 記載の記録媒体。

【請求項 2 3】

プロセッサの並列数と同数の操作記述を蓄積完了したことに起因して、蓄積バッファに

50

蓄積された操作記述をプロセッサに実行させる

プログラムを記録したことを特徴とする請求項 19 記載の記録媒体。

【請求項 24】

前記蓄積バッファから蓄積された操作記述をプロセッサに実行させることに起因して消去させる

プログラムを記録したことを特徴とする請求項 19 記載の記録媒体。

【請求項 25】

前記蓄積バッファに蓄積された操作記述を、プロセッサに複数回実行させる

プログラムを記録したことを特徴とする請求項 19 記載の記録媒体。

【請求項 26】

前記蓄積バッファに、蓄積された操作記述を消去せずに実行するか否かを示すビットを保持させ、

前記蓄積バッファに蓄積された操作記述をプロセッサに実行させたとき、前記ビットに従って消去させないかまたは消去させる

プログラムを記録したことを特徴とする請求項 19 記載の記録媒体。

【請求項 27】

プロセッサにおける並列数よりも少ない数の操作記述からなる命令で構成される

プログラムを記録したことを特徴とする請求項 19 記載の記録媒体。

【請求項 28】

前記蓄積バッファ中の領域の格納順位に従って蓄積用の操作記述を順に蓄積させる

プログラムを記録したことを特徴とする請求項 27 記載の記録媒体。

【請求項 29】

フェッチされた命令中の他の操作記述と、蓄積バッファに蓄積された蓄積用の操作記述とを、プロセッサに並列に実行させる

プログラムを記録したことを特徴とする請求項 19 記載の記録媒体。

【請求項 30】

前記蓄積バッファ内の複数段のバッファ領域に蓄積された蓄積用の操作記述を順にプロセッサに実行させる

プログラムを記録したことを特徴とする請求項 29 記載の記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コンパイラとプロセッサと記憶媒体とに関し、特に単一命令方式またはVLIW (Very Long Instruction Word)方式のプロセッサの実行コード効率を向上させる技術に関するものを含む。

【背景技術】

【0002】

近年の電子技術の発展により、高性能なプロセッサが普及し、あらゆる分野で用いられている。そのようなプロセッサでは命令の並列処理により高い性能を達成している。VLIWと呼ばれるアーキテクチャも命令の並列処理の1つの形態であり、VLIWアーキテクチャを採るプロセッサ(以下、「VLIWプロセッサ」という。)は、内部に複数の演算器を備え、1つの命令に置かれたスロットと呼ばれる複数のフィールドに指定されたオペレーションを同時並列に実行する。このようなVLIWプロセッサの機械命令プログラムは、コンパイラによって高級言語で記述されたプログラムにおけるオペレーションレベルでの並列性が検出されスケジューリングされた後に生成されたものである。機械命令プログラムは実行コードとも呼ばれる。

【0003】

(第1の従来技術)

図21は、第1の従来技術におけるプロセッサの構成図である。

【0004】

10

20

30

40

50

第1の従来技術におけるプロセッサは2つのオペレーションを並列実行するもので、図5に示すような第1と第2の2つのスロットからなる命令列で構成されるプログラムがROM1に格納され、それぞれのスロットに書かれたオペレーションが第1命令解読器4と第2命令解読器5とで解読された後、第1演算器13と第2演算器14とで実行される。

【0005】

(第2の従来技術)

図22は、第2の従来技術におけるプロセッサの構成図である。

【0006】

第2の従来技術におけるプロセッサは3つのオペレーションを並列実行するものだが、基本的な考え方は第1の従来技術におけるプロセッサと同一であり、図14に示すような第1から第3の3つのスロットからなる命令列で構成されるプログラムがROM41に格納され、それぞれのスロットに書かれたオペレーションが第1命令解読器45から第3命令解読器47で解読された後、第1演算器58から第3演算器60で実行される。つまり、1つの命令を構成するスロットの数が増えたに過ぎない。

【発明の開示】

【発明が解決しようとする課題】

【0007】

しかしながら、上記の従来技術ではいずれも、命令中に置かれたノーオペレーションコード(nopコード)によってプログラムサイズが増大する問題がある。

なお、プログラムサイズが増大はコード効率の低下とも表現される。VLIWプロセッサの命令の各スロットにはプロセッサが備える複数の演算器に対応したオペレーションが指定されるが、オペレーションの依存関係等により、常に並列実行可能なスロットの数だけのオペレーションがスケジューリングできるとは限らないからである。有効なオペレーションが置けない場合、コンパイラはそのスロットにnopコードを生成する。

【0008】

上記の第1の従来技術では、例えば図5に示すように、命令2ではBとCの2つの有効なオペレーションを指定することができるが、命令1では第2スロットに有効なオペレーションを指定することができずnopとなっている。また上記の第2の従来技術では、例えば図14に示すように、命令1では第2と第3スロットに、命令2では第3スロットに有効なオペレーションを指定することができずnopとなっている。このように一般にVLIWプロセッサは、命令の並列度が増すほど挿入されるnopコードの数が増加し、コード効率がさらに悪化するという問題がある。これは、コンパイラにおいて全スロットに有効なオペレーションがスケジューリングできる確率が並列度に逆比例することに起因する。

【0009】

そこで、本発明はかかる点に鑑みてなされたものであり、命令中の無駄領域を低減するコンパイラとプロセッサとを提供することを第1の目的とする。

【0010】

また、本発明の第2の目的は、VLIWプロセッサにおける命令の並列度の向上に伴うnopコードの増大を軽減するコンパイラとプロセッサとを提供することである。

【課題を解決するための手段】

【0011】

本願発明のプロセッサは、命令に含まれる操作記述を一旦蓄積するための蓄積バッファと、蓄積バッファの蓄積および出力を制御する制御部と、命令を解読および実行する解読実行部とを備え、前記制御部は、フェッチされた命令に含まれる蓄積用の操作記述と他の操作記述とを判別し、判別された蓄積用の操作記述を前記蓄積バッファに書き込み前記解読実行部への前記他の操作記述の出力と、前記解読実行部への前記蓄積バッファからの操作記述の出力とを制御することを特徴するものである。

【0012】

また、このような命令を実行するプロセッサは、命令中の蓄積ビットの値に基づいて命令を一旦蓄積バッファに格納したのち、蓄積バッファに蓄積された命令を実行することを

10

20

30

40

50

特徴とするプロセッサである。

【発明の効果】

【0013】

以上の説明から明かなように、本発明によればnopを減少させることができ、コードサイズを小さくすることができる。

【発明を実施するための最良の形態】

【0014】

以下、本発明の実施の形態について、図面を用いて詳細に説明する。

(実施の形態1)

実施の形態1は、nopの代わりに有効なオペレーションが配置された命令を一旦蓄積しその後実行するもので、コードサイズの減少を図るものである。 10

【0015】

1. コンパイラ

図1は、コンパイラの構成を示すブロック図である。

【0016】

コンパイラ102は、ユーザが記述したC言語プログラム101を翻訳し、機械命令プログラム112を出力する。

【0017】

コンパイラ102は、C言語プログラム101を読込用バッファ104に読み込むファイル読込部103と、読込用バッファ104に読み込まれたC言語プログラムの構文や意味を解析して中間コードを生成し中間コード用バッファ106に書き込む構文解析部105と、中間コード用バッファ106に格納された中間コードを入力して命令の2並列実行を目的とする命令のスケジューリングを行い、未圧縮の機械命令プログラムを生成し暫定出力バッファ108に書き込む機械命令生成部107と、暫定出力バッファ108に格納された未圧縮の機械命令プログラムを圧縮して目的とする機械命令プログラムを生成し出力用バッファ110に書き込む機械命令圧縮部109と、出力用バッファ110に格納された機械命令プログラムをファイルに出力するファイル出力部111とから構成される。ここで「機械命令プログラムの圧縮」とは、機械命令プログラムの各命令に含まれるnopコードを有効なオペレーションに置き換えることを言う。この圧縮を行う機械命令圧縮部109を除く各要素は、公知の技術に基づいて構成すればよいのでここでは説明を省略する。 20 30

【0018】

命令の順に未圧縮の機械命令プログラムを検索して、同順の第1スロットのnopコードと第2スロットのnopコードとのペアを抽出し、このnopコードペアの第1スロットおよび第2スロットを、該ペア以降に最初に現れる有効オペレーションのペアの第1スロットおよび第2スロットのオペレーションでそれぞれ置き換え、置き換えたことをマーキングするとともに、置き換えに使った有効オペレーションのペアを削除する。これにより、2つの有効なオペレーションを含む命令を、これよりも前に存在するnopの代わりに配置し、nopを削減するものである。

【0019】

図2から図4は、機械命令圧縮部109の処理フローを示したフローチャートである。機械命令圧縮部109の処理フローは、以下の動作例を用いて詳細に説明する。 40

【0020】

1.1 機械命令圧縮部109の動作例

図5は、未圧縮の機械命令プログラムの例示図であり、機械命令生成部107が上記した第1の従来技術にならって生成したものである。

【0021】

命令は第1と第2の2つのスロットで構成され、AからJの記号は有効なオペレーションが、nopはnopコードが生成されていることを示す。

【0022】

図 6 は、圧縮された機械命令プログラムの例示図であり、機械命令圧縮部 109 が図 5 の未圧縮の機械命令プログラムを以下の手順で圧縮したものである。

【0023】

命令は第 1 と第 2 の 2 つのロットで構成され、各ロットは 1 ビットの蓄積ビットとオペレーション (OP) フィールドとからなる。A から J の記号は図 5 と同様、有効なオペレーションを示す。

【0024】

以下に、図 5 のプログラムを入力とした場合における機械命令圧縮部 109 の動作について図 2 から図 6 を用いて説明する。

【0025】

図 2 に示すように、先ず初期化を行う。初期化は、命令ポインタ N を最初の命令、即ち図 5 の命令 1 に合わせることで、命令先取カウンタ m を 1 にすること、第 1 ロット nop カウンタ C1 および第 2 ロット nop カウンタ C2 を 0 にすること、第 1 ロットバッファカウンタ B1 を 0 にすること、第 2 ロットバッファカウンタ B2 を 0 にすることである (ステップ S201)。ここで、N、m、C1、C2、B1、B2 は機械命令圧縮部 109 の内部的に作られたパラメータである。

【0026】

次に、N で示される命令、即ち図 5 の命令 1 の型を評価する。命令 1 は第 1 ロットが有効なオペレーション A で第 2 ロットが nop コードなので、「OP(1):nop(2)型」が該当し処理 A へ飛ぶ (ステップ S202)。(1)、(2) は第 1 ロット、第 2 ロットを意味する。

【0027】

図 3 に示す処理 A では、最初に第 2 ロット nop カウンタ C2 をインクリメントし C2=1 と、第 2 ロットバッファカウンタ B2 をインクリメントし B2=1 とする (ステップ S301)。次いで機械命令圧縮部 109 の内部的に作られたパラメータ C1X に第 1 ロット nop カウンタ C1 の値を代入し、C1X=0 となる (ステップ S302)。次いで (N+m) で示される命令、即ち図 5 の命令 2 の型を評価する。命令 2 は第 1、第 2 ロットがそれぞれ有効なオペレーション A、B なので、「OP(1):OP(2)型」が該当しステップ S312 へ飛ぶ (ステップ S303)。今、B2=1 で B2=2 を満たしているが、ステップ S307 の C1X=1 を満たさず (C1=0) 次にステップ S305 へ飛ぶ。ここで、C1X=1 を条件としているのは、OP(1):OP(2) を OP(1):NOP(2) に置き換えるというように、第 2 ロットのみを置き換えることにより新たに置き換え対象が生成されることを防止するためである。すなわち、この条件を付加することにより、OP(1):OP(2) は最終的には NOP(1):NOP(2) に置き換えられ、削除されることになる。また、B2=2 を条件としているのは、後述する図 7 に示すプロセッサは第 1、2 ロットそれぞれについて 2 組のバッファを備えており、これを越える置き換えを防止するためである。ここでは (N+m) で示される命令、即ち図 5 の命令 2 は最後の命令でないのでステップ S306 へ移り (ステップ S305)、命令先取カウンタ m を 2 にして次の命令に進め、ステップ S303 に戻る (ステップ S306)。

【0028】

次に (N+m) で示される命令、今度は図 5 の命令 3 の型を評価する。命令 3 は第 1 ロットが有効なオペレーション D で第 2 ロットが nop コードなので、「OP(1):nop(2)型」が該当しステップ S305 へ飛ぶ (ステップ S303)。(N+m) で示される命令、即ち図 5 の命令 3 は最後の命令でないのでステップ S306 へ移り (ステップ S305)、命令先取カウンタ m を 3 にして次の命令に進め、ステップ S303 に戻る (ステップ S306)。

【0029】

次に (N+m) で示される命令、今度は図 5 の命令 4 の型を評価する。命令 4 は第 1 ロットが nop コードで第 2 ロットが有効なオペレーション E なので、「nop(1):OP(2)型」が該当しステップ S304 へ飛ぶ (ステップ S303)。ここで C1X に 1 が足されて C1X=1 になる (ステップ S304)。(N+m) で示される命令、即ち図 5 の命令 4 は最後の命令でないのでステップ S306 へ移り (ステップ S305)、命令先取カウンタ m を 4 にして次の命令に進め、ステップ S303 に戻る (ステップ S306)。

10

20

30

40

50

【 0 0 3 0 】

次に(N+m)で示される命令、即ち図5の命令5の型を評価する。命令5は第1、第2スロットがそれぞれ有効なオペレーションF、Gなので、「OP(1):OP(2)型」が該当しステップS312へ飛ぶ(ステップS303)。今、B2=1でB2 2を満たしており、かつステップS307のC1X 1(C1X=1)を満たしてステップS308へ飛ぶ。ここでOP(2)は有効なままなのでステップS309へ飛び(ステップS308)、Nで示される命令、即ち図5の命令1の第1スロットの蓄積ビットを"0"にセットするとともに、第2スロットの蓄積ビットを"1"にセットしてOPフィールドをnopの代わりにオペレーションGで埋める。このように、OP(2)が有効であることを確認しているのは、OP(2)が存在していてもすでにnopの代わりに配置され、実質的には存在しない命令である場合があるからである。こうして図6の命令1が生成される。続いて(N+m)で示される命令、即ち図5の命令5のOP(2)を無効にする(ステップS309)。このときOP(1)はまだ有効なままなので処理Aを終了し、ステップS206へ飛ぶ(ステップS310)。なお、後述するが、OP(1)が無効であるとき(すでに置き換えられているとき)、ステップS311で命令を削除する。

10

【 0 0 3 1 】

処理Aから戻って、Nで示される命令、即ち図5の命令1は最後の命令でないでのステップS207に移り(ステップS206)、命令ポインタNを次の命令、即ち図5の命令2に進め、命令先取カウンタmを1に戻し、ステップS202へ戻る(ステップS207)。

【 0 0 3 2 】

引き続いてNで示される命令、即ち図5の命令2の型を評価する。命令2は上記の通り「OP(1):OP(2)型」が該当しステップS205に移る(ステップS202)。ここでNで示される命令、即ち図5の命令2の第1および第2スロットの蓄積ビットを"0"にセットする。こうして図6の命令2が生成される。続いてNで示される命令、即ち図5の命令2は最後の命令でないでのステップS207に移り(ステップS206)、命令ポインタNを次の命令、即ち図5の命令3に進め、命令先取カウンタmを1に戻し、ステップS202へ戻る(ステップS207)。

20

【 0 0 3 3 】

引き続いてNで示される命令、即ち図5の命令3の型を評価する。命令3は上記の通り「OP(1):nop(2)型」が該当し処理Aへ飛ぶ(ステップS202)。

【 0 0 3 4 】

処理Aでは、最初に第2スロットnopカウンタC2をインクリメントしC2=2と、第2スロットバッファカウンタB2をインクリメントしB2=2とする(ステップS301)。次いでパラメータC1Xに第1スロットnopカウンタC1の値を代入し、C1X=0となる(ステップS302)。次いで(N+m)で示される命令、即ち図5の命令4の型を評価する。命令4は上記の通り「nop(1):OP(2)型」が該当しステップS304へ飛ぶ(ステップS303)。ここでC1Xに1が足されてC1X=1になる(ステップS304)。(N+m)で示される命令、即ち図5の命令4は最後の命令でないでのステップS306へ移り(ステップS305)、命令先取カウンタmを2にして次の命令に進め、ステップS303に戻る(ステップS306)。

30

【 0 0 3 5 】

次に(N+m)で示される命令、即ち図5の命令5の型を評価する。命令5は上記の通り「OP(1):OP(2)型」が該当しステップS307へ飛ぶ(ステップS303)。今、C1X=1なのでC1X 1を満たしてステップS308へ飛ぶ。ここでOP(2)は以前に無効にされているのでステップS305へ飛ぶ(ステップS308)。(N+m)で示される命令、即ち図5の命令5は最後の命令でないでのステップS306へ移り(ステップS305)、命令先取カウンタmを3にして次の命令に進め、ステップS303に戻る(ステップS306)。

40

【 0 0 3 6 】

次に(N+m)で示される命令、即ち図5の命令6の型を評価する。命令6は第1スロットがnopコードで第2スロットが有効なオペレーションHなので、「nop(1):OP(2)型」が該当しステップS304へ飛ぶ(ステップS303)。ここでC1Xに1が足されてC1X=2になる(ステップS304)。(N+m)で示される命令、即ち図5の命令6は最後の命令でないでのステップS30

50

6へ移り（ステップS305）、命令先取カウンタmを4にして次の命令に進め、ステップS303に戻る（ステップS306）。

【0037】

次に(N+m)で示される命令、即ち図5の命令7の型を評価する。命令7は第1、第2スロットがそれぞれ有効なオペレーションI、Jなので、「OP(1):OP(2)型」が該当しステップS312へ飛ぶ（ステップS303）。今、B2=2でB2 2を満たしており、かつステップS307のC1X 1(C1X=2)を満たしてステップS308へ飛ぶ。ここでOP(2)は有効なままなのでステップS309へ飛び（ステップS308）、Nで示される命令、即ち図5の命令3の第1スロットの蓄積ビットを"0"にセットするとともに、第2スロットの蓄積ビットを"1"にセットしてOPフィールドをnopの代わりにオペレーションJで埋める。こうして図6の命令3が生成される。続いて(N+m)で示される命令、即ち図5の命令7のOP(2)を無効にする（ステップS309）。このときOP(1)はまだ有効なままなので処理Aを終了し、ステップS206へ飛ぶ（ステップS310）。

10

【0038】

処理Aから戻って、Nで示される命令、即ち図5の命令3は最後の命令でないでのステップS207に移り（ステップS206）、命令ポインタNを次の命令、即ち図5の命令4に進め、命令先取カウンタmを1に戻し、ステップS202へ戻る（ステップS207）。

【0039】

引き続いてNで示される命令、即ち図5の命令4の型を評価する。命令4は上記の通り「nop(1):OP(2)型」が該当し処理Bへ飛ぶ（ステップS202）。

20

【0040】

処理Bでは、最初に第1スロットnopカウンタC1をインクリメントしC1=1と、第1スロットバッファカウンタB1をインクリメントしB1=1とする（ステップS401）。次いで機械命令圧縮部109の内部的に作られたパラメータC2Xに第2スロットnopカウンタC2の値を代入し、C2X=2となる（ステップS402）。次いで(N+m)で示される命令、即ち図5の命令5の型を評価する。命令5は上記の通り「OP(1):OP(2)型」が該当しステップS412へ飛ぶ（ステップS403）。今、B1=1でB1 2を満たしており、かつステップS407のC2X 1(C2X=2)を満たしてステップS408へ飛ぶ。ここでOP(1)は有効なままなのでステップS409へ飛び（ステップS408）、Nで示される命令、即ち図5の命令4の第2スロットの蓄積ビットを"0"にセットするとともに、第1スロットの蓄積ビットを"1"にセットしてOPフィールドをnopの代わりにオペレーションFで埋める。こうして図6の命令4が生成される。続いて(N+m)で示される命令、即ち図5の命令5のOP(1)を無効にする（ステップS409）。次にOP(2)は以前に無効にされているのでステップS411へ飛ぶ（ステップS410）。ここで(N+m)で示される命令、即ち図5の命令5を削除し、第1スロットnopカウンタC1および第2スロットnopカウンタC2をデクリメントしC1=0、C2=1と、第1スロットバッファカウンタB1および第2スロットバッファカウンタB2をデクリメントしB1=0、B2=1となる（ステップS411）。これで処理Bを終了し、ステップS206へ飛ぶ。

30

【0041】

処理Bから戻って、Nで示される命令、即ち図5の命令4は最後の命令でないでのステップS207に移り（ステップS206）、命令ポインタNを次の命令、即ち図5の命令6（命令5は削除された）に進め、命令先取カウンタmを1に戻し、ステップS202へ戻る（ステップS207）。

40

【0042】

引き続いてNで示される命令、即ち図5の命令6の型を評価する。命令6は上記の通り「nop(1):OP(2)型」が該当し処理Bへ飛ぶ（ステップS202）。

【0043】

処理Bでは、最初に第1スロットnopカウンタC1をインクリメントしC1=1と、第1スロットバッファカウンタB1をインクリメントしB1=1になる（ステップS401）。次いでパラメータC2Xに第2スロットnopカウンタC2の値を代入し、C2X=1となる（ステップS402）。次いで(N+m)で示される命令、即ち図5の命令7の型を評価する。命令7は上記の通り「OP(1)

50

「OP(2)型」が該当しステップS412へ飛ぶ(ステップS403)。今、B1=1でB1 2を満たしており、かつステップS407のC2X 1(C2X=1)を満たしてステップS408へ飛ぶ。ここでOP(1)は有効なままなのでステップS409へ飛び(ステップS408)、Nで示される命令、即ち図5の命令6の第2スロットの蓄積ビットを"0"にセットするとともに、第1スロットの蓄積ビットを"1"にセットしてOPフィールドをnopの代わりにオペレーションIで埋める。こうして図6の命令5が生成される。続いて(N+m)で示される命令、即ち図5の命令7のOP(1)を無効にする(ステップS409)。次にOP(2)は以前に無効にされているのでステップS411へ飛ぶ(ステップS410)。ここで(N+m)で示される命令、即ち図5の命令7を削除し、第1スロットnopカウンタC1および第2スロットnopカウンタC2をデクリメントしC1=0、C2=0と、第1スロットバッファカウンタB1、第2スロットバッファカウンタB2をデクリメントしB1=0、B2=0となる(ステップS411)。これで処理Bを終了し、ステップS206へ飛ぶ。

10

【0044】

処理Bから戻って、Nで示される命令、即ち図5の命令6は最後の命令なので(命令7は削除された)全ての処理を終了する(ステップS206)。

【0045】

以上のように、図5の未圧縮の機械命令プログラムは図6に示す圧縮された機械命令プログラムに変換される。なお、上記動作例で図3および図4における未通過のステップがあるが、図3および図4は2つのスロットについて相補的であるので説明を省略する。

【0046】

2. プロセッサ

20

図7は、プロセッサの概略構成図である。

【0047】

このプロセッサは、命令フェッチステージ(以下、IFステージ)、解読およびレジスタ読出しステージ(以下、DECステージ)、実行ステージ(以下、EXステージ)の3つのステージからなる3段パイプライン構造を成している。

【0048】

図7において、1は機械語プログラムを格納するROM、2と3は機械語命令(以下、命令と略す)の第1スロットと第2スロットの内容を格納するそれぞれI1ラッチとI2ラッチ、4と5はそれぞれI1ラッチ2とI2ラッチ3に保持された命令の第1スロットと第2スロットの内容を解読しプロセッサ各部を制御する第1命令解読器と第2命令解読器、6はオペランド格納するレジスタファイル、7と8はそれぞれI1ラッチ2とI2ラッチ3の内容の一部とレジスタファイル6の出力との2入力から1つを選択するD1セレクタとD2セレクタ、9と10はそれぞれD1セレクタ7とD2セレクタ8の出力を格納するD11ラッチとD12ラッチ、11と12はレジスタファイル6の出力を格納するD21ラッチとD22ラッチ、13はD11ラッチ9およびD21ラッチ11の内容を用いて算術論理演算を行う第1演算器、14はD12ラッチ10およびD22ラッチ12の内容を用いて算術論理演算を行う第2演算器で、第1演算器13と第2演算器14の出力はともにレジスタファイル6に接続される。15と16はそれぞれI1ラッチ2とI2ラッチ3に保持された命令の第1スロットと第2スロットの内容を保持するIB11バッファとIB12バッファで、両者を合わせてIB1バッファと記す。17と18はそれぞれI1ラッチ2とI2ラッチ3に保持された命令の第1スロットと第2スロットの内容を保持するIB21バッファとIB22バッファで、両者を合わせてIB2バッファと記す。IB1バッファおよびIB2バッファへは各スロットの蓄積ビットが"1"の時に内容が取込まれる。23、24はIB1バッファまたはIB2バッファをいずれかを選択して出力するセレクタ、19はROM1から読出された命令の第1スロットの内容またはセレクタ23のいずれかを選択してI1ラッチ2に出力するI1セレクタ、20はROM1から読出された命令の第2スロットの内容またはセレクタ24のいずれかを選択してI2ラッチ3に出力するI2セレクタ、21、22はI1ラッチ2、I2ラッチ3に格納されたデータの蓄積ビットが"1"のときnop(No Operation)を出力するnop生成器、25、26は蓄積ビットが"1"となったときは書き込み信号を"0"、"1"と反転し

30

40

50

て出力し、蓄積ビットが " 0 " のときは " 0 " を出力する書き込み信号生成器、27、28は命令の蓄積完了を検出するAND回路、29は蓄積した命令をデコード・実行する場合に命令フェッチを止めるための信号等生成するOR回路、30、31はクロックバッファである。なお、nop生成器21、22はI1ラッチ2、I2ラッチ3の出力のそれぞれのビットと、蓄積ビットを反転したものとの論理積を演算するAND回路で構成されており、蓄積ビットが " 1 " のときは、nopを意味する $(00 \cdots 0)_2$ を出力する。また、書き込み信号生成器25、26はT形フリップフロップとAND回路からなり、正転出力とT型フリップフロップのトリガ入力(I1ラッチ2、I2ラッチ3の蓄積ビット)との論理積をとるAND回路の出力をIB11バッファ15、IB12バッファ16への書き込み信号とし、反転出力とT型フリップフロップのトリガ入力との論理積をとるAND回路の出力をIB21バッファ17、IB22バッファ18への書き込み信号としている。

10

【0049】

レジスタファイル6は、レジスタR0からR7の汎用レジスタを含み、読出し4ポート、書き込み2ポートを有する。即ち、同時に4つのレジスタ(重複は可)の読出しと2つのレジスタ(重複は不可)の書き込みを許す。D1セクタ7およびD2セクタ8はそれぞれ第1命令解読器4および第2命令解読器5の指示により、命令に即値などの定数値を伴う場合はこれを選択する。

【0050】

このプロセッサはいわゆるVLIW(Very Long Instruction Word)形式の命令に基づいており、1つの命令で2つの演算などの操作が定義される。第1スロットのオペレーションは、I1ラッチ2に格納され第1命令解読器4で解読され第1演算器13で実行される。また第2スロットのオペレーションは、I2ラッチ3に格納され第2命令解読器5で解読され第2演算器14で実行される。このようにして同時に2つの操作を実行するためVLIW形式のプロセッサは効率が高い。

20

【0051】

2.1 プロセッサの動作例

以下に、図6の機械命令プログラムがROM1に格納された場合における上記構成をもつプロセッサの動作について図8を用いて説明する。

【0052】

図8は、図6の機械命令プログラムがROM1に格納された場合におけるプロセッサの動作タイミング図である。同図は、プロセッサの動作をパイプラインのIFステージでROM1から読出される命令、DECステージで解読される命令、EXステージで実行される命令と、IB1バッファおよびIB2バッファが保持する命令をマシンサイクルと呼ばれるタイミング毎に示している。以下、時間が経過する順にタイミング毎にその動作を説明する。なお図中、「:」はスロットの区切りを表し、左が第1スロット、右が第2スロットを意味し、「-」は有効なオペレーションが保持されていないもしくは作用していないことを表す。

30

【0053】

また、初期状態として、IB11バッファ15、IB12バッファ16、IB21バッファ17、IB22バッファ18はリセットされているものとする。

40

【0054】

(タイミングt1)・IFステージ: 命令1

命令1がROM1から読出され、第1スロット(蓄積ビットが " 0 " でオペレーションA)がI1ラッチ2に、第2スロット(蓄積ビットが " 1 " でオペレーションG)がI2ラッチ3に格納される。すなわち、IBバッファにはまだオペレーションが蓄積されていない(蓄積ビットが " 1 " でない)ので、I1SEL19、I2SEL20はいずれも、ROM1からの出力を選択し出力する。

【0055】

(タイミングt2)・DECステージ: 命令1

50

蓄積ビットが " 1 " である I 2 ラッチ 3 の内容 (蓄積ビットが " 1 " でオペレーション G) が I B 1 2 バッファ 1 6 に取込まれる。具体的には、蓄積ビットが " 1 " であるため、書き込み信号生成器 2 6 により I B 1 2 バッファ 1 6 の書き込み信号がイネーブルとなり、I 2 ラッチ 3 の内容が I B 1 2 バッファ 1 6 に蓄積されることとなる。また、I 2 ラッチ 3 に格納された命令 1 の第 2 スロットの蓄積ビットが " 1 " であるため、nop 生成器 2 2 は $\text{nop}(00 \cdots 0)_2$ を出力し、第 2 命令解読器 5 は EX ステージで実質的に何らの動作もしないようなデコード結果を出力する。

【 0 0 5 6 】

一方、I 1 ラッチ 2 に格納された命令 1 の第 1 スロットが第 1 命令解読器 4 で解読される。解読された結果としてオペレーション A であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納される。

・ IF ステージ : 命令 2

命令 2 が ROM 1 から読出され、第 1 スロット (蓄積ビットが " 0 " でオペレーション B) が I 1 ラッチ 2 に、第 2 スロット (蓄積ビットが " 0 " でオペレーション C) が I 2 ラッチ 3 に格納される。

【 0 0 5 7 】

(タイミング t3) ・ EX ステージ : 命令 1

D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納されたオペランドを第 1 演算器 1 3 に入力してオペレーション A の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。一方、オペレーション G は蓄積ビットが " 1 " で nop 生成器 2 2 により nop に無効化されているため、第 2 演算器 1 4 は作用しない。

・ DEC ステージ : 命令 2

I 1 ラッチ 2 に格納された命令 2 の第 1 スロットが第 1 命令解読器 4 で解読される。解読された結果としてオペレーション B であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納される。一方、I 2 ラッチ 3 に格納された命令 2 の第 2 スロットが第 2 命令解読器 5 で解読される。解読された結果としてオペレーション C であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 2 ラッチ 1 0 と D 2 2 ラッチ 1 2 に格納される。このとき、オペレーション B, C いずれの蓄積ビットも " 0 " であるため、いずれの I B バッファの書き込み信号もイネーブルとならず、書き込みは行われない。

・ IF ステージ : 命令 3

命令 3 が ROM 1 から読出され、第 1 スロット (蓄積ビットが " 0 " でオペレーション D) が I 1 ラッチ 2 に、第 2 スロット (蓄積ビットが " 1 " でオペレーション J) が I 2 ラッチ 3 に格納される。

【 0 0 5 8 】

(タイミング t4) ・ EX ステージ : 命令 2

D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納されたオペランドを第 1 演算器 1 3 に入力してオペレーション B の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。一方、D 1 2 ラッチ 1 0 と D 2 2 ラッチ 1 2 に格納されたオペランドを第 2 演算器 1 4 に入力してオペレーション C の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。

・ DEC ステージ : 命令 3

蓄積ビットが " 1 " である I 2 ラッチ 3 の内容 (蓄積ビットが " 1 " でオペレーション J) が I B 2 2 バッファ 1 8 に取込まれる。具体的には、蓄積ビットが " 1 " であるため、I B 1 2 バッファ 1 6 または I B 2 2 バッファ 1 8 にデータの書き込みをしようとするが、すでに I B 1 2 バッファ 1 6 にはデータを書き込んだので、書き込み信号生成器 2 6 により I B 2 2 バッファ 1 8 の書き込み信号がイネーブルになる。また、I 2 ラッチ 3 に格納された命令 3 の第 2 スロットの蓄積ビットが " 1 " であるため、nop 生成器 2 2 は

10

20

30

40

50

n o p を出力し、第 2 命令解読器 5 は E X ステージで実質的に何らの動作もしないようなデコード結果を出力する。

【 0 0 5 9 】

一方、I 1 ラッチ 2 に格納された命令 3 の第 1 スロットが第 1 命令解読器 4 で解読される。解読された結果としてオペレーション D であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納される。

・ IF ステージ：命令 4

命令 4 が ROM 1 から読出され、第 1 スロット（蓄積ビットが " 1 " でオペレーション F）が I 1 ラッチ 2 に、第 2 スロット（蓄積ビットが " 0 " でオペレーション E）が I 2 ラッチ 3 に格納される。

10

【 0 0 6 0 】

（タイミング t5）・ EX ステージ：命令 3

D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納されたオペランドを第 1 演算器 1 3 に入力してオペレーション D の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。一方、オペレーション J は蓄積ビットが " 1 " で n o p 生成器 2 2 により n o p に無効化されているため第 2 演算器 1 4 は作用しない。

・ DEC ステージ：命令 4

蓄積ビットが " 1 " である I 1 ラッチ 2 の内容（蓄積ビットが " 1 " でオペレーション F）が I B 1 1 バッファ 1 5 に取込まれる。具体的には、蓄積ビットが " 1 " であるため、書き込み信号生成器 2 5 により I B 1 1 バッファ 1 5 の書き込み信号がイネーブルとなり、I 1 ラッチ 2 の内容が I B 1 1 バッファ 1 5 に蓄積されることとなる。また、I 2 ラッチ 2 に格納された命令 4 の第 1 スロットの蓄積ビットが " 1 " であるため、n o p 生成器 2 1 は n o p を出力し、第 1 命令解読器 4 は E X ステージで実質的に何らの動作もしないようなデコード結果を出力する。

20

【 0 0 6 1 】

一方、I 2 ラッチ 3 に格納された命令 4 の第 2 スロットが第 2 命令解読器 5 で解読される。解読された結果としてオペレーション E であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 2 ラッチ 1 0 と D 2 2 ラッチ 1 2 に格納される。・ IF ステージ：I B 1 バッファ蓄積命令

30

I B 1 1 バッファ 1 5 と I B 1 2 バッファ 1 6 との蓄積ビットが共に " 1 " であるため、AND 回路 2 7 は蓄積バッファにデータが蓄積されたとして " 1 " を出力し、さらに OR 回路 2 9 が命令フェッチを中断すべく " 1 " を出力する。これにより、命令フェッチは中断される。これとともに、I B 1 1 バッファ 1 5 の蓄積ビットが " 1 " であるので、セクタ 2 3、2 4 は I B 1 バッファを選択・出力する。さらに OR 回路 2 9 の出力により、I 1 セクタ 1 9、I 2 セクタ 2 0 はそれぞれ I B 1 1 バッファ 1 5、I B 2 1 バッファ 1 6 を選択し、蓄積された命令は I 1 ラッチ 2、I 2 ラッチ 3 に格納される。これにより、I B 1 1 バッファ 1 5 および I B 2 1 バッファ 1 6 に格納された命令を使用したことになるので、クロックドバッファ 3 0 によりタイミングを調整して、I B 1 1 バッファ 1 5 および I B 2 1 バッファ 2 1 の内容をリセットし、蓄積ビットを " 0 " とする。なお、ここではバッファそのものをリセットしているが、蓄積ビットのみを " 0 " としても良い。なお、図面では省略しているが、I 1 セクタ 1 9 および I 2 セクタ 2 0 は蓄積された命令を選択するときは蓄積ビットを " 0 " にして、I 1 ラッチ 2、I 2 ラッチ 3 に出力する。n o p 生成器 2 1、2 2 が蓄積された命令を n o p に無効化する事を防止するためである。また、セクタ 2 3、2 4 の切り換え信号を I B 1 1 バッファ 1 5 の蓄積ビットのみとしているのは、蓄積された命令が実行されるときは常に I B 1 1 バッファ 1 5 および I B 1 2 バッファ 1 6（または I B 2 1 バッファ 1 7 および I B 2 2 バッファ 1 8）の蓄積ビットが " 1 " となっているため、I B 1 2 バッファ 1 6 の蓄積ビットまでを見る必要はなく、また I B 1 バッファに蓄積された命令を実行するということは I B 2 バッ

40

50

ァに蓄積された命令はまだ実行しない状態を意味するからである。このため、I B 1 1 バッファ 1 5 の蓄積ビットに限らず、いずれかの蓄積ビットの値により切り換え信号とできる。

【 0 0 6 2 】

(タイミング t6) ・ EXステージ：命令 4

オペレーション F は蓄積ビットが " 1 " で n o p 生成器 2 1 により n o p に無効化されているため、第 1 演算器 1 3 は作用しない。一方、D 1 2 ラッチ 1 0 と D 2 2 ラッチ 1 2 に格納されたオペランドを第 2 演算器 1 4 に入力してオペレーション E の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。

・ DECステージ：I B 1 バッファ蓄積命令

I 1 ラッチ 2 に格納された第 1 スロットが第 1 命令解読器 4 で解読される。解読された結果としてオペレーション F であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納される。一方、I 2 ラッチ 3 に格納された第 2 スロットが第 2 命令解読器 5 で解読される。解読された結果としてオペレーション G であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 2 ラッチ 1 0 と D 2 2 ラッチ 1 2 に格納される。

・ IFステージ：命令 5

命令 5 が ROM 1 から読出され、第 1 スロット (蓄積ビットが " 1 " でオペレーション I) が I 1 ラッチ 2 に、第 2 スロット (蓄積ビットが " 0 " でオペレーション H) が I 2 ラッチ 3 に格納される。

【 0 0 6 3 】

(タイミング t7) ・ EXステージ：I B 1 バッファ蓄積命令

D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納されたオペランドを第 1 演算器 1 3 に入力してオペレーション F の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。一方、D 1 2 ラッチ 1 0 と D 2 2 ラッチ 1 2 に格納されたオペランドを第 2 演算器 1 4 に入力してオペレーション G の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。

・ DECステージ：命令 5

蓄積ビットが " 1 " である I 1 ラッチ 2 の内容 (蓄積ビットが " 1 " でオペレーション I) が I B 2 1 バッファ 1 7 に取込まれる。具体的には、蓄積ビットが " 1 " であるため、I B 1 1 バッファ 1 5 または I B 2 1 バッファ 1 7 にデータの書き込みをしようとするが、すでに I B 1 1 バッファ 1 5 にはデータを書き込んだので、書き込み信号生成器 2 5 により I B 2 1 バッファ 1 7 の書き込み信号がイネーブルとなる。また、I 1 ラッチ 2 に格納された命令 5 の第 1 スロットの蓄積ビットが " 1 " であるため、n o p 生成器 2 1 は n o p を出力し、第 1 命令解読器 4 は E X ステージで実質的に何らの動作もしないようなデコード結果を出力する。

【 0 0 6 4 】

一方、I 2 ラッチ 3 に格納された命令 5 の第 2 スロットが第 2 命令解読器 5 で解読される。解読された結果としてオペレーション H であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 2 ラッチ 1 0 と D 2 2 ラッチ 1 2 に格納される。・ IFステージ：I B 2 バッファ蓄積命令

I B 2 1 バッファ 1 7 と I B 2 2 バッファ 1 8 との蓄積フラグが共に " 1 " であるため、AND 回路 2 7 は蓄積バッファにデータが蓄積されたとして " 1 " を出力し、さらに OR 回路 2 9 が命令フェッチを中断すべく " 1 " を出力する。これにより、命令フェッチは中断される。これとともに、I B 1 1 バッファ 1 5 の蓄積ビットが " 0 " である (I B 2 バッファに蓄積された命令が存在する可能性がある) ので、セクタ 2 3、2 4 は I B 2 バッファを選択・出力する。さらに、OR 回路 2 9 の出力により I 1 セクタ 1 9、I 2 セクタ 2 0 はそれぞれ I B 2 1 バッファ 1 7、I B 2 2 バッファ 1 8 を選択し、蓄積さ

10

20

30

40

50

れた命令はI 1 ラッチ 2、I 2 ラッチ 3 に格納される。これにより、I B 2 1 バッファ 1 7 および I B 2 2 バッファ 1 8 に格納された命令を使用したことになるので、クロックド バッファ 3 1 によりタイミングを調整して、I B 2 1 バッファ 1 7 および I B 2 2 バッファ 1 8 の内容をリセットし、蓄積フラグを " 0 " とする。

【 0 0 6 5 】

(タイミング t8) ・ EXステージ : 命令 5

オペレーション I は蓄積ビットが " 1 " で nop 生成器 2 1 により nop に無効化されているため、第 1 演算器 1 3 は作用しない。一方、D 1 2 ラッチ 1 0 と D 2 2 ラッチ 1 2 に格納されたオペランドを第 2 演算器 1 4 に入力してオペレーション H の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。

10

・ DECステージ : I B 2 バッファ蓄積命令

I 1 ラッチ 2 に格納された第 1 スロットが第 1 命令解読器 4 で解読される。解読された結果としてオペレーション I であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納される。一方、I 2 ラッチ 3 に格納された第 2 スロットが第 2 命令解読器 5 で解読される。解読された結果としてオペレーション J であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 2 ラッチ 1 0 と D 2 2 ラッチ 1 2 に格納される。

【 0 0 6 6 】

(タイミング t9) ・ EXステージ : I B 2 バッファ蓄積命令

20

D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納されたオペランドを第 1 演算器 1 3 に入力してオペレーション I の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。一方、D 1 2 ラッチ 1 0 と D 2 2 ラッチ 1 2 に格納されたオペランドを第 2 演算器 1 4 に入力してオペレーション J の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。

【 0 0 6 7 】

3 . 記録媒体

本発明の記録媒体の実施の形態として、図 6 の機械命令プログラム 1 1 2 を記録した磁気ディスク (フレキシブルディスクやハードディスクなど) 、光ディスク (CD-ROM や PD など) 、光磁気ディスク、半導体メモリ (ROM やフラッシュメモリ など) がある。

30

【 0 0 6 8 】

以上のように本実施の形態によれば、コンパイラの機械命令圧縮部 1 0 9 が、同順の第 1 スロットの nop コードと第 2 スロットの nop コードとのペアを抽出し、この nop コードペアの第 1 スロットおよび第 2 スロットを、該ペア以降に最初に現れる有効オペレーションのペアの第 1 スロットおよび第 2 スロットのオペレーションでそれぞれ置き換え、置き換えに使った有効オペレーションのペアを削除することにより命令中の無駄領域が低減され、プログラムサイズの削減を図ることができる。

【 0 0 6 9 】

また本実施の形態のプロセッサによれば、散在する従来の nop コードの位置に埋められた有効オペレーションを蓄積する I B 1 バッファおよび I B 2 バッファを設け、I B 1 バッファまたは I B 2 バッファのいずれかに有効オペレーションが 2 つ揃った時点でこれを実行することにより、従来の処理性能を保つつ圧縮された機械命令プログラムの実行が可能である。

40

【 0 0 7 0 】

さらに本実施の形態によれば、従来の nop コードの位置に該 nop コードと同じスロットにある有効オペレーションを埋めるという考え方に基づくため、第 1 スロットと第 2 スロットとの間でオペレーションを相互に転送する必要がなくなりプロセッサの構成が簡単になるという効果を有する。具体的には、I 1 ラッチ 2 のオペレーションは I B 1 1 バッファ 1 5 または I B 2 1 バッファ 1 7 にのみ蓄積し、それらに蓄積したオペレーションは I 1 ラッチ 2 にのみ戻せばよく、逆に I 2 ラッチ 3 のオペレーションは I B 1 2 バッファ 1 6

50

またはI B 2 2 バッファ 1 8 にのみ蓄積し、それらに蓄積したオペレーションはI 2 ラッチ 3 にのみ戻せばよいため、第 1 スロットと第 2 スロットとの間の転送路および転送制御手段を必要としない。

【 0 0 7 1 】

なお、本実施形態のプロセッサでは、I 1 セクタ 1 9 および I 2 セクタ 2 0 をそれぞれ I 1 ラッチ 2 および I 2 ラッチ 3 の入力側に設けているが、それぞれ I 1 ラッチ 2 および I 2 ラッチ 3 の出力側に設け、第 1 命令解読器 4 および第 2 命令解読器 5 の入力を選択するようにしてもよい。このようにする際は、I B 1 バッファおよび I B 2 バッファへの入力を IF ステージにおいて ROM 1 から直接行うように変更しなければならないが、I B 1 バッファおよび I B 2 バッファへの取り込みや I 1 セクタ 1 9 および I 2 セクタ 2 0 の選択については本実施形態と同様に蓄積ビットの値で制御すればよい。

10

【 0 0 7 2 】

また、本実施形態のプロセッサでは、I B 1 バッファと I B 2 バッファとの 2 つの蓄積バッファを設けているが、いかなる数であってもよい。蓄積バッファの数が増えるほど nop コードを有効オペレーションで埋める機会が増加し、プログラムサイズの一層の削減が図れる。このことは、例えば本実施形態のプロセッサにおける I B 2 バッファがないものとする、図 5 の命令 3 の第 2 スロットの nop コードは有効オペレーションで埋められないことから容易に伺い知れる。

【 0 0 7 3 】

(実施の形態 2)

20

実施の形態 2 は、実施の形態 1 から、nop コードのスロットの有効オペレーションでの埋め方の自由度を高めたものである。

【 0 0 7 4 】

1 . コンパイラ

コンパイラの構成は、機械命令圧縮部 1 0 9 の動作を除いて実施の形態 1 に記したものと同一である。機械命令圧縮部 1 0 9 は図 1 0 から図 1 2 に示すもので、次の原理に基づいて動作する。

【 0 0 7 5 】

命令の順に未圧縮の機械命令プログラムを検索して、第 1 スロットと第 2 スロットのいずれかにかかわらず出現順序が連続する 2 つの nop コードを抽出し、これらの nop コードのスロットを、該 2 つの nop コードの以降に最初に現れる有効オペレーションのペアの第 1 スロットおよび第 2 スロットのオペレーションでそれぞれ置き換え、置き換えたことをマーキングするとともに、置き換えに使った有効オペレーションのペアを削除し、削除されたペアの直前の命令の第 1 スロットと第 2 スロットのいずれかに削除したことをマーキングする。すなわち、実施の形態 1 のコンパイラでは各スロット毎に nop を削除していたが、本実施の形態におけるコンパイラはスロットを意識せず、nop を出現順に有効なオペレーションに置き換えるものである。このため、いずれかのスロットに nop が集中していた場合でも有効なオペレーションに置き換えることができる。

30

【 0 0 7 6 】

1 . 1 機械命令圧縮部 1 0 9 の動作例

40

図 9 は、圧縮された機械命令プログラムの例示図であり、機械命令圧縮部 1 0 9 が図 5 の未圧縮の機械命令プログラムを上述の手順で圧縮したものである。圧縮された命令は第 1 と第 2 の 2 つのスロットで構成され、各スロットは蓄積ビットと位置ビットとオペレーション (OP) フィールドとからなる。A から J の記号は有効なオペレーションを示す。蓄積ビットと位置ビットは次のようにエンコードしている。

0 0、0 1 何もしない 1 0 I B 1 バッファに蓄積すべし 1 1 I B 2 バッファに蓄積すべし

具体的に説明すると、図 5 の命令 5 のオペレーション F とオペレーション G とを命令 1 と命令 3 との nop コードのスロットに埋め、命令 7 のオペレーション I とオペレーション J とを命令 4 と命令 6 との nop コードのスロットに埋め、埋められた以上のスロットの蓄積ビ

50

ットを01にセットし、命令5と命令7とを削除する。オペレーションFとオペレーションGとオペレーションIとオペレーションJとは、この順にIB1バッファの第1スロット、第2スロット、IB2バッファの第1スロット、第2スロットに蓄積されることを前提にしており、削除された命令5の直前の命令4の第2スロットの蓄積ビットには10を、削除された命令7の直前の命令6の第2スロットの蓄積ビットには11をセットする。その他のスロットの蓄積ビットは00である。このようにして生成された機械命令プログラムが図9に示すものである。なお図9の命令5は図5の命令6から生成したものである。

【0077】

図10と図2とを比べると、nopカウンタが1つである点(S501)、位置ビットをセットする点(S505)が異なる。nopカウンタを1つとしたのは、本実施の形態では、実施の形態1とは異なりスロットを意識する必要がないからである。ただし、このnopカウンタは図2のnopカウンタとは全く異なる用途に使用されるもので、位置ビットの値を決定するためにnopが出願する度に"0"、"1"を繰り返すものである。

【0078】

図11、12は図3、4と基本的に同じであるが、位置ビットの値をnopカウンタによって決定しているところが大きく異なる(S609,S709)。また、前述したnopカウンタの用途のため、命令を削除したときはC=0とする点も異なる(S611,S711)。

【0079】

2. プロセッサ

図13は、プロセッサのIFステージ部分の概略構成図である。

【0080】

DECステージおよびEXステージの図示していない部分は図7と同じ構成であり、また実施の形態1と同一の構成要素には同一の符号を付している。図7と比べると、セクタ32、33を有している点異なる。すなわち、位置ビットの値により、I1ラッチ2に格納された命令であってもIB12バッファ16またはIB22バッファ18に、I2ラッチ3に格納された命令であってもIB11バッファ15またはIB21バッファ17に命令を蓄積することが可能となり、実施の形態1と比べてさらにnopを軽減することができる。他の動作は、実施の形態1と同じであるため説明は省略する。

【0081】

3. 記録媒体

本発明の記録媒体の実施の形態として、図9の機械命令プログラムを記録した磁気ディスク(フレキシブルディスクやハードディスクなど)、光ディスク(CD-ROMやPDなど)、光磁気ディスク、半導体メモリ(ROMやフラッシュメモリなど)がある。

【0082】

以上のように本実施の形態によれば、コンパイラの機械命令圧縮部109が、第1スロットと第2スロットのいずれかにかかわらず出現順序が連続する2つのnopコードを抽出し、このnopコードスロットを、該2つのnopコードの以降に最初に現れる有効オペレーションのペアの第1スロットおよび第2スロットのオペレーションでそれぞれ置き換え、置き換えに使った有効オペレーションのペアを削除することにより命令中の無駄領域が低減され、プログラムサイズの削減を図ることができる。

【0083】

また本実施の形態のプロセッサによれば、散在する従来のnopコードの位置に埋められた有効オペレーションを蓄積するIB1バッファおよびIB2バッファを設け、実行させるべき位置の直前の命令中の蓄積ビットでIB1バッファまたはIB2バッファのいずれかを指定して蓄積したオペレーションを実行することにより、従来の処理性能を保つつ圧縮された機械命令プログラムの実行が可能である。

【0084】

さらに本実施の形態によれば、スロットの位置にかかわらず出現順にnopコードを有効オペレーションで埋めるという考え方に基づくため、nopコードが第1スロットと第2スロットとのいずれにあるかを識別する必要がなくなり、コンパイラの構成が実施の形態1

10

20

30

40

50

のものより簡単になるという効果を有する。

【0085】

なお、本実施形態のプロセッサでは、I 1 セクタ 1 9 および I 2 セクタ 2 0 をそれぞれ I 1 ラッチ 2 および I 2 ラッチ 3 の入力側に設けているが、それぞれ I 1 ラッチ 2 および I 2 ラッチ 3 の出力側に設け、第 1 解読器 4 および第 2 解読器 5 の入力を選択するようにしてもよい。このようにする際は、I B 1 バッファおよび I B 2 バッファへの入力を IF ステージにおいて ROM 1 から直接行うように変更し、ROM 1 から読出された命令の蓄積ビットの値によって I B 1 セクタ 3 1 と I B 2 セクタ 3 2 とを制御するように変更しなければならないが、I B 1 バッファおよび I B 2 バッファへの取り込みや I 1 セクタ 1 9 および I 2 セクタ 2 0 の選択については本実施形態と同様に蓄積ビットの値で制御すればよい。

10

【0086】

また、本実施形態のプロセッサでは、I B 1 バッファと I B 2 バッファとの 2 つの蓄積バッファを設けているが、いかなる数であってもよい。蓄積バッファの数が増えるほど nop コードを有効オペレーションで埋める機会が増加し、プログラムサイズの一層の削減が図れる。このことは、例えば本実施形態のプロセッサにおける I B 2 バッファがないものとする、図 5 の命令 4 の第 1 スロットの nop コードは有効オペレーションで埋められないことから容易に伺い知れる。

【0087】

(実施の形態 3)

20

実施の形態 3 は、2 つのスロットしかない命令で 3 つのオペレーションを並列実行する VLIW アーキテクチャのコンパイラおよびプロセッサである。

【0088】

1. コンパイラ

コンパイラの構成は、機械命令生成部 1 0 7 と機械命令圧縮部 1 0 9 との動作を除いて実施の形態 1 に記したものと同一である。機械命令生成部 1 0 7 は、中間コード用バッファ 1 0 6 に格納された中間コードを入力して命令の 3 並列実行 (実施の形態 1 は 2 並列実行) を目的とする命令のスケジューリングを行い、未圧縮の機械命令プログラムを生成し暫定出力バッファ 1 0 8 に書き込む。機械命令圧縮部 1 0 9 は次の原理に基づいて動作する。

30

【0089】

命令の順に未圧縮の機械命令プログラムを検索して、第 3 スロットを除く第 1 スロットと第 2 スロットのいずれかにかかわらず出現順序が連続する 3 つの nop コードを抽出し、これらの nop コードのスロットを、該 3 つの nop コードの以降に最初に現れる 3 つの有効オペレーションが指定される命令の第 1 スロットから第 3 スロットのオペレーションでそれぞれ置き換え、置き換えたことをマーキングするとともに、置き換えに使った 3 つの有効オペレーションが指定される命令を削除し、削除されたペアの直前の命令の第 1 スロットと第 2 スロットのいずれかに削除したことをマーキングする。

【0090】

1. 1 機械命令圧縮部 1 0 9 の動作例

40

図 1 5 は、圧縮された機械命令プログラムの例示図であり、機械命令圧縮部 1 0 9 が図 1 4 の未圧縮の機械命令プログラムを上述の手順で圧縮したものである。圧縮された命令は第 1 と第 2 の 2 つのスロットで構成され、各スロットは 2 ビットの蓄積ビットとオペレーション (OP) フィールドとからなる。A から H の記号は有効なオペレーションを示す。蓄積ビット (左側) と実行ビット (右側) の 2 ビットは次のようにエンコードしている。

0 0 何もしない 1 0 オペレーションは置き換えられたものであり、I B バッファに第 1、第 2、第 3 スロットの順に逐次蓄積すべし 0 1 直後の命令が削除されたので、I B バッファの命令を実行すべし 1 1 (未使用)

具体的に説明すると、図 1 4 の命令 5 のオペレーション F とオペレーション G とオペレーション H とを命令 1 の第 2 スロットと命令 3 の第 2 スロットと命令 4 の第 1 スロットとの n

50

opコードの-slotに埋め、埋められた以上の-slotの蓄積ビットを01にセットし、命令5を削除する。オペレーションFとオペレーションGとオペレーションHとは、この順にIBバッファの第1-slot、第2-slot、第3-slotに蓄積されることを前提にしており、削除された命令5の直前の命令4の第2-slotの蓄積ビットは"1"と実行ビットは"0"とセットする。その他の-slotの蓄積ビットは"0"と実行ビットは"0"とセットする。このようにして生成された機械命令プログラムが図15に示すものである。なお「IBバッファ」は次に説明する。

【0091】

2. プロセッサ

図16は、プロセッサの概略構成図である。

10

【0092】

図7と比べると、2つの-slotしかない命令で3つのオペレーションを並列実行するために、2つの-slotの命令をIB3バッファ41を含む3つのバッファに蓄積することにより内部で3つの-slotの命令に変換するものである。そして、3つ目の-slotの命令を与えるためのI3ラッチ38、nop生成器39、第3命令解読器40を有し、さらに3つ目の-slotの命令を実行するためのD3セクタ34、D13ラッチ35、D23ラッチ36及び第3演算器37を有する点で異なる。また、リングカウンタ42により、IB1バッファ15、IB2バッファ16、IB3バッファ41の書き込み信号を順にイネーブルにする。

【0093】

20

2.1 プロセッサの動作例

以下に、図15の機械命令プログラムがROM1に格納された場合における上記構成をもつプロセッサの動作について図17を用いて説明する。

【0094】

図17は、図15の機械命令プログラムがROM1に格納された場合におけるプロセッサの動作タイミング図である。同図は、プロセッサの動作をパイプラインのIFステージでROM41から読出される命令、DECステージで解読される命令、EXステージで実行される命令と、IBバッファが保持する命令をマシンサイクルと呼ばれるタイミング毎に示している。以下、時間が経過する順にタイミング毎にその動作を説明する。なお図中、「:」は-slotの区切りを表し、左が第1-slot、中央が第2-slot、右が第3-slotを意味し、「-」は有効なオペレーションが保持されていないもしくは作用していないことを表す。

30

【0095】

(タイミングt1)

初期状態として、IB1バッファ15、IB2バッファ16、IB3バッファ41がリセットされ、それぞれに $(0 \cdots 00)_2$ が格納されているものとする。また、リングカウンタ42も初期状態として $(001)_2$ にセットされ、I1ラッチ2またはI2ラッチ3に蓄積ビットが"1"の最初のオペレーションが格納されると $(100)_2$ となり、IB1バッファ15にオペレーションが蓄積されることとなる。

・IFステージ：命令1

40

命令1がROM1から読出され、第1-slot(オペレーションA)がI1ラッチ2に、第2-slot(オペレーションF)がI2ラッチ3に格納される。I3ラッチ38にはIB3バッファ41の $(0 \cdots 00)_2$ が格納される。

【0096】

(タイミングt2)・DECステージ：命令1

蓄積ビットが"1"であるI2ラッチ3の内容(オペレーションF)がIB1バッファ15に取込まれる。具体的には、蓄積ビットが"1"である最初のオペレーションなのでリングカウンタ42が $(100)_2$ を出力することにより、IB1バッファ15の書き込み信号がイネーブルとなり、I2ラッチ3の内容がIB1バッファ15に蓄積される。

【0097】

50

I 1 ラッチ 2 に格納された命令 1 の第 1 スロットが第 1 命令解読器 4 で解読される。解読された結果としてオペレーション A であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納される。一方、I 2 ラッチ 3 に格納された命令 1 の第 2 スロットの蓄積ビットが " 1 " であるため、n o p 生成器 2 2 は n o p を出力し、第 2 命令解読器 5 は E X ステージで実質的に何らの動作もしないようなデコード結果を出力する。また、3 つのスロットの命令を実行する場合以外は第 3 演算器 3 7 を動作させる必要がないため、実行ビットが " 0 " のときは、n o p 生成器 3 9 は n o p を出力する。

・ IF ステージ：命令 2

命令 2 が ROM 1 から読出され、第 1 スロット（オペレーション B）が I 1 ラッチ 2 に、第 2 スロット（オペレーション C）が I 2 ラッチ 3 に格納される。I 3 ラッチ 3 1 には再び I B 3 バッファ 4 1 の $(0 \cdots 00)_2$ が格納される。

【 0 0 9 8 】

（タイミング t3）・EX ステージ：命令 1

D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納されたオペランドを第 1 演算器 1 3 に入力してオペレーション A の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。一方、第 2 演算器 1 4 と第 3 演算器 3 7 は n o p 生成器 2 2、3 9 により無効化されているため作用しない。

・ DEC ステージ：命令 2

I 1 ラッチ 2 に格納された命令 2 の第 1 スロットが第 1 命令解読器 4 で解読される。解読された結果としてオペレーション B であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 1 ラッチ 9 と D 2 1 ラッチ 1 1 に格納される。一方、I 2 ラッチ 3 に格納された命令 2 の第 2 スロットが第 2 命令解読器 5 で解読される。解読された結果としてオペレーション C であることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値が D 1 2 ラッチ 1 0 と D 2 2 ラッチ 1 2 に格納される。また、I 3 ラッチ 3 8 の実行ビットは " 0 " であるため、n o p 生成器 3 9 は n o p を出力し、第 3 命令解読器 4 0 は E X ステージで実質的に何らの動作もしないようなデコード結果を出力する。

・ IF ステージ：命令 3

命令 3 が ROM 1 から読出され、第 1 スロット（蓄積ビットが $(00)_2$ でオペレーション D）が I 1 ラッチ 2 に、第 2 スロット（蓄積ビットが $(01)_2$ でオペレーション G）が I 2 ラッチ 3 に格納される。I 3 ラッチ 3 8 には再び I B 3 バッファ 4 1 の $(0 \cdots 00)_2$ が格納される。

【 0 0 9 9 】

（タイミング t4）・EX ステージ：命令 2

D 1 1 ラッチ 9 と D 2 1 ラッチ 5 5 に格納されたオペランドを第 1 演算器 1 3 に入力してオペレーション B の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。一方、D 1 2 ラッチ 1 1 と D 2 2 ラッチ 1 2 に格納されたオペランドを第 2 演算器 1 4 に入力してオペレーション C の演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。また、第 3 演算器 3 7 は n o p 生成器 3 9 により無効化されているので作用しない。

・ DEC ステージ：命令 3

蓄積ビットが $(10)_2$ である I 2 ラッチ 3 の内容（オペレーション G）が I B 2 バッファ 1 6 に取込まれる。具体的には、タイミング t1 とほとんど同様の動作であるが、I B 1 バッファ 1 5 にはすでにオペレーション F が蓄積されているので、リングカウンタ 4 2 が $(010)_2$ を出力することにより、I B 2 バッファ 1 6 の書き込み信号がイネーブルとなり、I B 2 バッファ 1 6 にオペレーションが蓄積される。

【 0 1 0 0 】

I 1 ラッチ 2 に格納された命令 3 の第 1 スロットが第 1 命令解読器 4 で解読される。解

10

20

30

40

50

読された結果としてオペレーションDであることが判明する。この解読に基づいてレジスタファイル6から汎用レジスタが読出され、読出された値または命令中の定数値がD11ラッチ9とD21ラッチ55に格納される。一方、I2ラッチ3に格納された命令3の第2スロットの蓄積ビットが"1"であるため、nop生成器22はnopを出力し、第2命令解読器5はEXステージで実質的に何らの動作もしないようなデコード結果を出力する。また、実行フラグは"0"であるため、nop生成器39はnopを出力し、第3命令解読器40はEXステージで実質的に何らの動作もしないようなデコード結果を出力する。

・IFステージ：命令4

命令4がROM1から読出され、第1スロット（オペレーションH）がI1ラッチ2に、第2スロット（オペレーションE）がI2ラッチ3に格納される。I3ラッチ38には再びIB3バッファ41の(0...00)₂が格納される。

【0101】

（タイミングt5）・EXステージ：命令3

D11ラッチ9とD21ラッチ55に格納されたオペランドを第1演算器13に入力してオペレーションDの演算を行う。演算結果は必要に応じてレジスタファイル6の汎用レジスタに格納する。一方、第2演算器14と第3演算器37とはnop生成器22、39により無効化されているので作用しない。

・DECステージ：命令4

蓄積ビットが"1"であるI1ラッチ2の内容（オペレーションH）がIB3バッファ41に取込まれる。このとき、IB1バッファ15、IB2バッファ16には既にオペレーションが蓄積されているので、リングカウンタ42は(001)₂を出力することによりIB3バッファ41の書き込み信号がイネーブルとなり、IB3バッファ41にオペレーションが蓄積される。また、I1ラッチ2に格納された命令4の第1スロットの蓄積ビットが"1"であるため、nop生成器21はnopを出力し、第1命令解読器4はEXステージで実質的に何らの動作もしないようなデコード結果を出力する。

【0102】

一方、I2ラッチ3に格納された命令4の第2スロットが第2命令解読器5で解読される。解読された結果としてオペレーションEであることが判明する。この解読に基づいてレジスタファイル6から汎用レジスタが読出され、読出された値または命令中の定数値がD12ラッチ11とD22ラッチ12に格納される。また、実行フラグは"0"であるため、nop生成器39はnopを出力し、第3命令解読器40は実行ステージで実質的に何らの動作もしないようなデコード結果を出力する。

・IFステージ：IBバッファ蓄積命令

I2ラッチ3に格納された命令4の第2スロットの実行ビットが"1"であるため、命令フェッチ制御部により命令フェッチを中断する。これとともに、I1セクタ19、I2セクタ20がそれぞれIB1バッファ15、IB2バッファ16を選択し、I1ラッチ2、I2ラッチ3、I3ラッチ38にはIB1バッファ15、IB2バッファ16、IB3バッファ41の内容が格納される。そして、I3ラッチ38の実行ビットが"1"となると、IBバッファの内容をリセットする。

【0103】

（タイミングt6）・EXステージ：命令4

第1演算器13、第3演算器37はnop生成器21、nop生成器39により無効化されているので作用しない。一方、D12ラッチ10とD22ラッチ12に格納されたオペランドを第2演算器14に入力してオペレーションEの演算を行う。演算結果は必要に応じてレジスタファイル6の汎用レジスタに格納する。

・DECステージ：IBバッファ蓄積命令

I1ラッチ2に格納された第1スロットが第1命令解読器4で解読される。解読された結果としてオペレーションFであることが判明する。この解読に基づいてレジスタファイル6から汎用レジスタが読出され、読出された値または命令中の定数値がD11ラッチ9

10

20

30

40

50

とD 2 1 ラッチ 5 5 に格納される。一方、I 2 ラッチ 3 に格納された第 2 スロットが第 2 命令解読器 5 で解読される。解読された結果としてオペレーションGであることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値がD 1 2 ラッチ 1 1 とD 2 2 ラッチ 1 2 に格納される。また、I 3 ラッチ 3 に格納された第 3 スロットが第 3 命令解読器 4 0 で解読される。すなわち、実行ビットが " 1 " であるため、nop生成器 3 9 はI 3 ラッチ 3 8 の内容をそのまま出力し、解読された結果としてオペレーションHであることが判明する。この解読に基づいてレジスタファイル 6 から汎用レジスタが読出され、読出された値または命令中の定数値がD 1 3 ラッチ 3 5 とD 2 3 ラッチ 3 6 に格納される。

【 0 1 0 4 】

10

(タイミング t7) ・ EXステージ : I B バッファ蓄積命令

D 1 1 ラッチ 9 とD 2 1 ラッチ 5 5 に格納されたオペランドを第 1 演算器 1 3 に入力してオペレーションFの演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。一方、D 1 2 ラッチ 1 1 とD 2 2 ラッチ 1 2 に格納されたオペランドを第 2 演算器 1 4 に入力してオペレーションGの演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。また、D 1 3 ラッチ 3 5 とD 2 3 ラッチ 3 6 に格納されたオペランドを第 3 演算器 3 7 に入力してオペレーションHの演算を行う。演算結果は必要に応じてレジスタファイル 6 の汎用レジスタに格納する。

【 0 1 0 5 】

3 . 記録媒体

20

本発明の記録媒体の実施の形態として、図 1 5 の機械命令プログラムを記録した磁気ディスク (フレキシブルディスクやハードディスクなど) 、光ディスク (CD-ROMやPDなど) 、光磁気ディスク、半導体メモリ (ROMやフラッシュメモリなど) がある。

【 0 1 0 6 】

以上のように本実施の形態によれば、コンパイラの機械命令圧縮部 1 0 9 が、第 3 スロットを除く第 1 スロットと第 2 スロットのいずれかにかかわらず出現順序が連続する 3 つのnopコードを抽出し、これらのnopコードのスロットを、該 3 つのnopコードの以降に最初に現れる 3 つの有効オペレーションが指定される命令の第 1 スロットから第 3 スロットのオペレーションでそれぞれ置き換え、置き換えに使った 3 つの有効オペレーションが指定される命令を削除することにより命令中の無駄領域が低減され、プログラムサイズの削減を図ることができる。特に本実施の形態によれば、従来 3 つのスロットからなる命令で 3 並列実行していた所を、従来nopコードとなるスロットを利用して 2 つスロットからなる命令で実行できるため、極めてコード効率が高い。上記に示した動作例では、図 1 2 の 3 スロット × 5 命令 = 1 5 スロットが、図 1 5 の 2 スロット × 4 命令 = 8 スロットに圧縮されていることがわかる。

30

【 0 1 0 7 】

また本実施の形態のプロセッサによれば、散在する従来のnopコードの位置に埋められた有効オペレーションを蓄積する I B バッファを設け、実行させるべき位置の直前の命令中の蓄積ビットで I B バッファを指定して蓄積したオペレーションを実行することにより、従来の処理性能を保ちつつ圧縮された機械命令プログラムの実行が可能である。

40

【 0 1 0 8 】

なお、本実施形態のプロセッサでは、I 1 セレクタ 1 9 および I 2 セレクタ 2 0 をそれぞれ I 1 ラッチ 2 および I 2 ラッチ 4 3 の入力側に設けているが、それぞれ I 1 ラッチ 2 および I 2 ラッチ 3 の出力側に設け、第 1 命令解読器 4 および第 2 命令解読器 4 の入力を選択するようにしてもよい。このようにする際は、I B バッファへの入力を IFステージにおいて ROM 1 から直接行うように変更し、ROM 1 から読出された命令の蓄積ビットの値によって I B セレクタ 6 6 を制御するように変更しなければならないが、I B バッファへの取り込みや I 1 セレクタ 1 9 および I 2 セレクタ 2 0 の選択については本実施形態と同様に蓄積ビットの値で制御すればよい。

【 0 1 0 9 】

50

また、本実施形態のプロセッサでは、I Bバッファという1つの蓄積バッファを設けているが、複数設けてもよい。蓄積バッファの数が増えるほどnopコードを有効オペレーションで埋める機会が増加し、プログラムサイズの一層の削減が図れる。

【0110】

さらにまた、本実施形態のプロセッサでは、3つの命令解読器と3つの演算器を設けて最大3並列実行を達成しているが、これらを4つずつ設けて4並列実行してもよいし、あるいはそれ以上であってもよい。4並列実行の場合は、本実施形態と同様に2つスロットからなる命令の未圧縮時にnopコードとなるスロットを4つ利用して有効オペレーションを埋めてもよいし、3つスロットからなる命令の未圧縮時にnopコードとなるスロットを4つ利用して有効オペレーションを埋めてもよい。但し前者の場合、I Bバッファをもう1スロット分だけ多く備える必要がある。前者は後者に比べて未圧縮時にnopコードとなるスロットが極めて多い場合に有効で、相当のコード効率の向上が望める。このようにすることにより、VLIWプロセッサにおける命令の並列度が向上しても、nopコードの増大を大幅に軽減することができる。

【0111】

(実施の形態4)

実施の形態4は、実施の形態3から、第3スロットのオペレーションだけを第1あるいは第2スロットのnopコードのスロットに埋めるように変えたものである。

【0112】

1. コンパイラ

コンパイラの構成は、機械命令圧縮部109の動作を除いて実施の形態3に記したものと同一である。機械命令圧縮部109は次の原理に基づいて動作する。

【0113】

命令の順に未圧縮の機械命令プログラムを検索して、第3スロットを除く第1スロットと第2スロットのいずれかにかかわらず1つのnopコードを抽出し、このnopコードのスロットを、該nopコードの以降に最初に現れる第3スロットに有効オペレーションが指定される命令の該オペレーションで置き換え、置き換えたことをマーキングするとともに、置き換えに使った有効オペレーションが指定される命令の第3スロットを削除し、該命令の第1スロットと第2スロットのいずれかに削除したことをマーキングする。

【0114】

1.1 機械命令圧縮部109の動作例

図18は、圧縮された機械命令プログラムの例示図であり、機械命令圧縮部109が図12の未圧縮の機械命令プログラムを上述の手順で圧縮したものである。圧縮された命令は第1と第2の2つのスロットで構成され、各スロットは2ビットの蓄積ビットとオペレーション(OP)フィールドとからなる。AからHの記号は有効なオペレーションを、nopは有効でないnopコードを示す。蓄積ビットの2ビットは次のようにエンコードしている。
00 何もしない 01 オペレーションは置き換えられたものであり、I Bバッファに蓄積すべし 10 第3スロットが削除されたので、第3スロットにはI Bバッファのオペレーションを実行すべし 11 (未使用)

具体的に説明すると、第3スロットに置かれたオペレーションである、図14の命令5のオペレーションHを、命令1の第2スロットのnopコードのスロットに埋め、埋められたスロットの蓄積ビットを01にセットし、命令5の第3スロットを削除する。オペレーションHは、I Bバッファに蓄積されることを前提にしており、第3スロットが削除された命令5の第2スロットの蓄積ビットには10をセットする(第1スロットの蓄積ビットであってもよい)。その他のスロットの蓄積ビットは00である。このようにして生成された機械命令プログラムが図18に示すものである。ここでは、命令3の第2スロットと命令4の第1スロットのnopコードは置き換えられずに残っている。なお「I Bバッファ」は次に説明する。

【0115】

2. プロセッサ

10

20

30

40

50

図19は、プロセッサのIFステージ部分の概略構成図である。

【0116】

DECステージおよびEXステージの図示していない部分は図16と同じ構成であり、また図16と同一の構成要素には同一の符号を付している。このプロセッサは、図14に示すものと比べると、IBバッファ50を1つだけ有するものである点で異なる。このため、図16と比べるとIBバッファが1つで足りることはもちろんの事、3つのバッファに左から蓄積するためのセクタ41、42が不要となり回路が簡単化できる。動作は、蓄積先がIBバッファ50に固定される以外は、実施の形態3と同じであるため説明は省略する。

【0117】

3. 記録媒体

本発明の記録媒体の実施の形態として、図18の機械命令プログラムを記録した磁気ディスク（フレキシブルディスクやハードディスクなど）、光ディスク（CD-ROMやPDなど）、光磁気ディスク、半導体メモリ（ROMやフラッシュメモリなど）がある。

【0118】

以上のように本実施の形態によれば、コンパイラの機械命令圧縮部109が、第3スロットを除く第1スロットと第2スロットのいずれかにかかわらず1つのnopコードを抽出し、このnopコードのスロットを、該nopコードの以降に最初に現れる第3スロットに有効オペレーションが指定される命令の該オペレーションで置き換え、置き換えに使った有効オペレーションが指定される命令の第3スロットを削除することにより命令中の無駄領域が低減され、プログラムサイズの削減を図ることができる。特に本実施の形態によれば、従来3つのスロットからなる命令で3並列実行していた所を、従来nopコードとなるスロットを利用して2つスロットからなる命令で実行できるため、極めてコード効率が高い。上記に示した動作例では、図12の3スロット×5命令=15スロットが、図18の2スロット×5命令=10スロットに圧縮されていることがわかる。

【0119】

また本実施の形態のプロセッサによれば、従来のnopコードの位置に埋められた有効オペレーションを蓄積するIBバッファを設け、命令中の蓄積ビットでIBバッファを指定して該命令のオペレーションと蓄積したオペレーションとを並列に実行することにより、従来の処理性能を保つつ圧縮された機械命令プログラムの実行が可能である。

【0120】

なお、本実施形態のプロセッサでは、IBバッファという1つの蓄積バッファを設けているが、複数設けてもよい。蓄積バッファの数が多くなるほどnopコードを有効オペレーションで埋める機会が増加し、プログラムサイズの一層の削減が図れる。例えば、命令3の第2スロットと命令4の第1スロットのnopコードは置き換えられずに残っているが、未圧縮（図14）の命令5の直後に第3スロットに有効オペレーションが置かれた命令が1つ後続する場合、または2つ後続する場合、それぞれ、これらのnopコードの一方または両方をその有効オペレーションで埋めることができる。

【0121】

さらにまた、本実施形態のプロセッサでは、3つの命令解読器と3つの演算器を設けて最大3並列実行を達成しているが、これらを4つずつ設けて4並列実行してもよいし、あるいはそれ以上であってもよい。4並列実行の場合は、本実施形態と同様に2つスロットからなる命令の未圧縮時にnopコードとなるスロットを2つ利用して有効オペレーションを埋めてもよいし、3つスロットからなる命令の未圧縮時にnopコードとなるスロットを1つ利用して有効オペレーションを埋めてもよい。但し前者の場合、IBバッファをもう1スロット分だけ多く備える必要がある。前者は後者に比べて未圧縮時にnopコードとなるスロットが極めて多い場合に有効で、相当のコード効率の向上が望める。このようにすることにより、VLIWプロセッサにおける命令の並列度が向上しても、nopコードの増大を大幅に軽減することができる。

【0122】

10

20

30

40

50

以上、本発明に係るコンパイラ及びプロセッサについて、上記の４つの実施形態に基づいて説明したが、本発明はこれら実施形態に限られないことは勿論である。即ち、

(１) 上記の４つの実施形態では、１つの命令に２つまたは３つのオペレーションを指定するVLIW形式のアーキテクチャとしているが、１つの命令で１つのオペレーションを指定するVLIW形式でないアーキテクチャでもよい。

【０１２３】

特に、固定長命令の場合には未使用領域を持つ命令が多く定義されることがある。例えば、MIPS RISCアーキテクチャによるプロセッサ“R3000”は３２ビット固定長命令を実行するが、このプロセッサの演算命令は図２３（ａ）に示すように、１２ビットのオペレーションフィールド（「op1」と「op2」で示す）とそれぞれが５ビットの３つのレジスタフィールド（ソースオペランドの「rs」および「rt」と、デスティネーションオペランドの「rd」で示す）から構成され、さらに５ビットの「res」で示す未使用領域を有する。本発明によれば、このような単一オペレーション命令中に生じる無駄領域の発生も回避される。具体的には、図２３（ｂ）に示すようにコンパイラが６つの命令Aから命令Fのそれぞれの未使用領域a～fを利用して、命令F以降に実行されるべき１つの命令を分割して配置するとともにこの命令を削除し、プロセッサ内に設けた命令蓄積レジスタにこれらを順に蓄積し、命令Fの実行後にこのレジスタの内容を実行する。こうすることにより、プログラム中の無駄領域が解消されコード効率が向上する。また、命令蓄積レジスタの内容の実行は、命令Fの直後でなく命令Fに続く他の命令の実行後でもよいし、また命令Fと並列に実行してもよい。特に後者の思想は、１つの命令で１つのオペレーションを指定するVLIW形式でないアーキテクチャにおいて局所的ではあるが２つのオペレーションを指定するVLIW形式のアーキテクチャを実現できるため有用である。また、このような命令蓄積レジスタを複数設けることにより、さらに３並列以上のVLIWアーキテクチャも実現可能である。なお、６つの命令Aから命令Fは必ずしもすき間なく連続している必要はない。

【０１２４】

(２) 上記の４つの実施形態では、命令蓄積レジスタ（ＩＢ１バッファ、ＩＢ２バッファ、ＩＢバッファが相当）を読出すと同時に内容を消去しているが、消去せずに複数回読出して再利用してもよい。例えば、実施の形態３および実施の形態４では２ビットの蓄積ビットが１１である状態を使用していないのでこれを利用し、蓄積ビットが１１の時はＩＢバッファを消去せずに実行する、とすることができる。こうすることにより、例えばプログラムがループを構成するような同じ命令を繰り返し実行する場合に、度々同じ命令を何度もＩＢバッファに蓄積する必要がなくなり、一層コード効率がよくなる。また、命令蓄積レジスタを、読出した直後に内容が消去されるものと消去されず再利用可能なものと２種類設けることも可能である。

【０１２５】

(３) 上記の４つの実施形態では、コンパイラにおいて、機械命令生成部１０７が一旦、従来と同じ機械命令プログラムを生成した後に機械命令圧縮部１０９がこれを圧縮しているが、両者の機能を一体にして、従来と同じ機械命令プログラムを生成することなく目的の圧縮された機械命令プログラムを直接生成するようにしてもよい。

【０１２６】

(４) 上記の４つの実施形態のプロセッサは、命令フェッチ、解読、実行の３段パイプラインで構成されるとしているが、パイプラインの段数は何段であってもよいし、パイプラインを採らなくともよい。

【図面の簡単な説明】

【０１２７】

【図１】実施の形態１に係るコンパイラの構成を示すブロック図

【図２】実施の形態１に係るコンパイラの機械命令圧縮部１０９の処理フローを示したフローチャート

【図３】実施の形態１に係るコンパイラの機械命令圧縮部１０９の処理フローを示したフローチャート

10

20

30

40

50

【図 4】実施の形態 1 に係るコンパイラの機械命令圧縮部 109 の処理フローを示したフローチャート

【図 5】未圧縮の機械命令プログラムの例示図

【図 6】実施の形態 1 に係る圧縮された機械命令プログラムの例示図

【図 7】実施の形態 1 に係るプロセッサの概略構成図

【図 8】実施の形態 1 に係るプロセッサの図 6 の機械命令プログラムに対応した動作タイミング図

【図 9】実施の形態 2 に係る圧縮された機械命令プログラムの例示図

【図 10】実施の形態 2 に係るコンパイラの機械命令圧縮部 109 の処理フローを示したフローチャート

10

【図 11】実施の形態 2 に係るコンパイラの機械命令圧縮部 109 の処理フローを示したフローチャート

【図 12】実施の形態 2 に係るコンパイラの機械命令圧縮部 109 の処理フローを示したフローチャート

【図 13】実施の形態 2 に係るプロセッサの IF ステージ部分の概略構成図

【図 14】未圧縮の機械命令プログラムの例示図

【図 15】実施の形態 3 に係る圧縮された機械命令プログラムの例示図

【図 16】実施の形態 3 に係るプロセッサの概略構成図

【図 17】実施の形態 3 に係るプロセッサの図 13 の機械命令プログラムに対応した動作タイミング図

20

【図 18】実施の形態 4 に係る圧縮された機械命令プログラムの例示図

【図 19】実施の形態 4 に係るプロセッサの IF ステージ部分の概略構成図

【図 20】実施の形態 4 に係るプロセッサの図 16 の機械命令プログラムに対応した動作タイミング図

【図 21】第 1 の従来技術におけるプロセッサの概略構成図

【図 22】第 2 の従来技術におけるプロセッサの概略構成図

【図 23】他の従来技術および他の実施形態に係る命令のフォーマット図

【符号の説明】

【0128】

1、41 ROM

30

2、42 I1 ラッチ

3、43 I2 ラッチ

4、45 第 1 命令解読器

5、46 第 2 命令解読器

6、48 レジスタファイル

7、49 D1 セレクタ

8、50 D2 セレクタ

9、52 D11 ラッチ

10、53 D12 ラッチ

11、55 D21 ラッチ

40

12、56 D22 ラッチ

13、58 第 1 演算器

14、59 第 2 演算器

15、33 IB11 バッファ

16、34 IB12 バッファ

17、35 IB21 バッファ

18、36 IB22 バッファ

19、64 I1 セレクタ

20、65 I2 セレクタ

21、37、67、72 制御回路

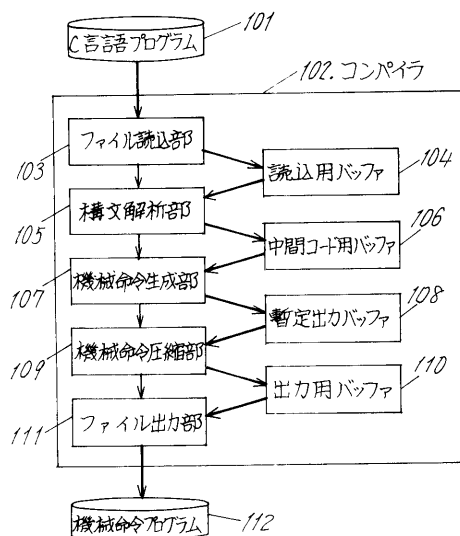
50

3 1 I B 1 セレクタ
 3 2 I B 2 セレクタ
 4 4 I 3 ラッチ
 4 7 第 3 命令読取器
 5 1 D 3 セレクタ
 5 4 D 1 3 ラッチ
 5 7 D 2 3 ラッチ
 6 0 第 3 演算器
 6 1 I B 1 バッファ
 6 2 I B 2 バッファ
 6 3 I B 3 バッファ
 6 6 I B セレクタ
 7 1 I B バッファ
 1 0 1 C 言語プログラム
 1 0 2 コンパイラ
 1 0 3 ファイル読込部
 1 0 4 読込用バッファ
 1 0 5 構文解析部
 1 0 6 中間コード用バッファ
 1 0 7 機械命令生成部
 1 0 8 暫定出力用バッファ
 1 0 9 機械命令圧縮部
 1 1 0 出力用バッファ
 1 1 1 ファイル出力部
 1 1 2 機械命令プログラム

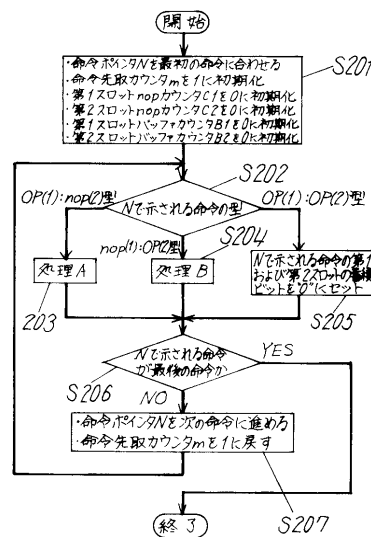
10

20

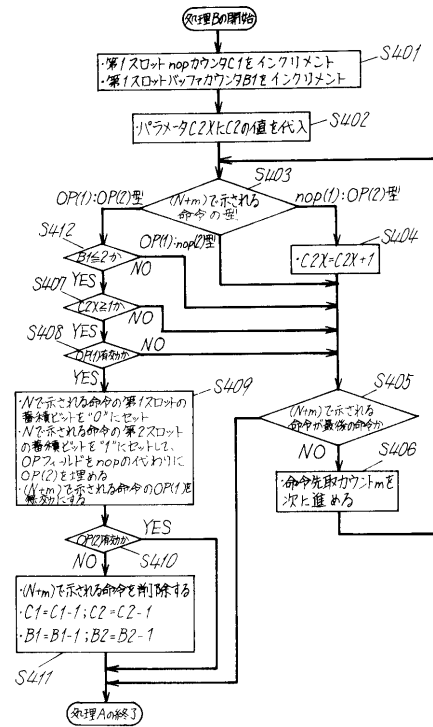
【図 1】



【図 2】



【 図 4 】



【圖 7】

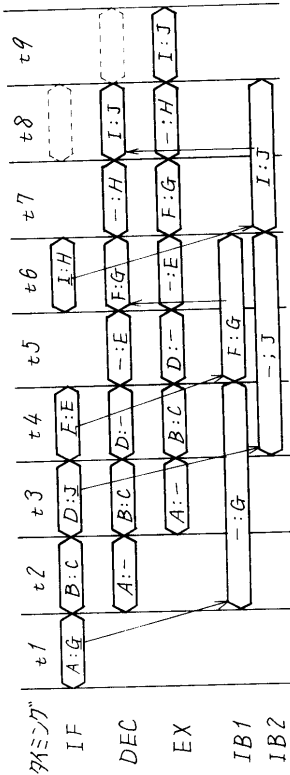
[illegible]

第1スロット 第2スロット

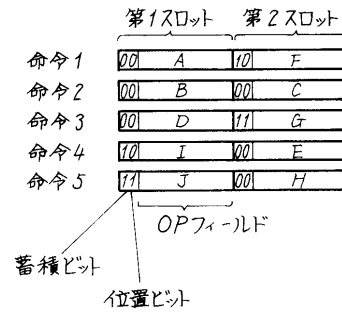
命令1	0	A	1	G
命令2	0	B	0	C
命令3	0	D	1	J
命令4	1	F	0	E
命令5	1	I	0	H

蓄積ビット OPフィールド

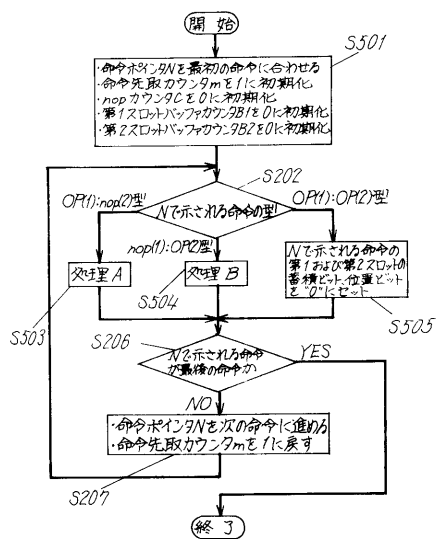
【図 8】



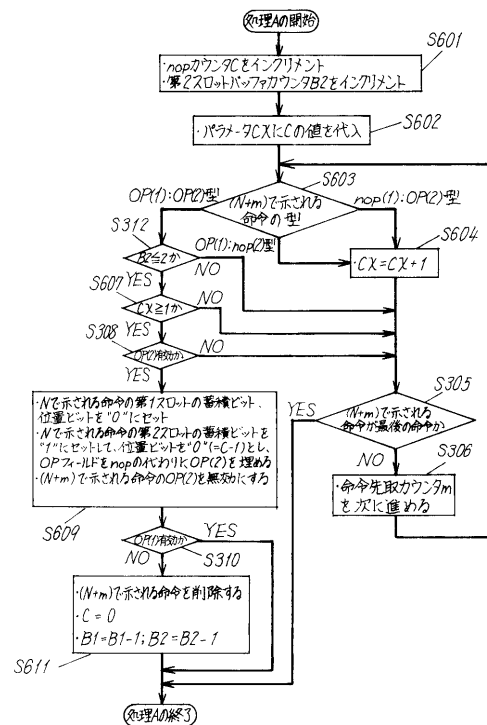
【図 9】



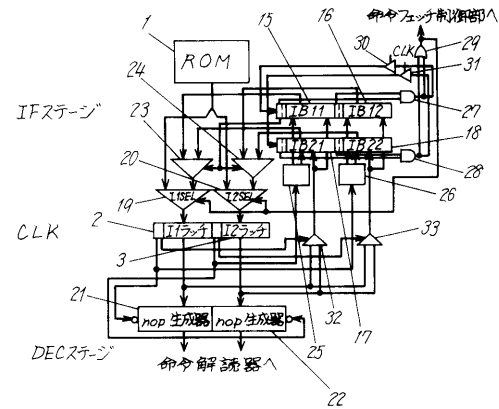
【図 10】



【図 11】

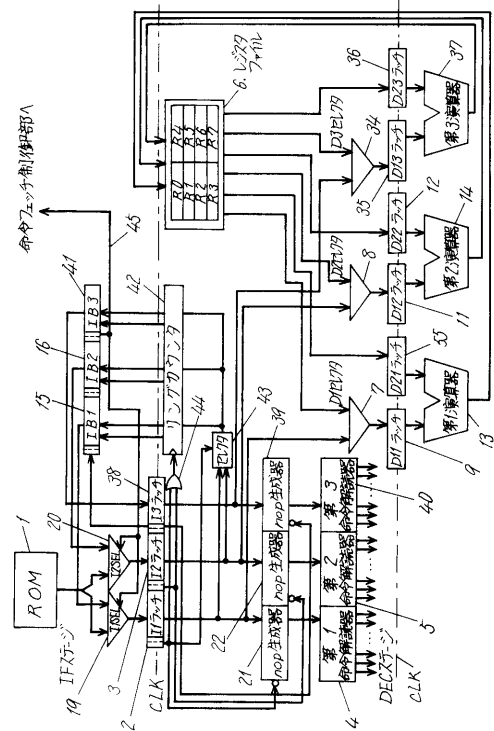


【 図 1 3 】



	第1スロット	第2スロット	第3スロット
命令1	A	nop	nop
命令2	B	C	nop
命令3	D	nop	nop
命令4	nop	E	nop
命令5	F	G	H

【 図 1 6 】



フロントページの続き

(56)参考文献 特開平 8 - 1 6 1 1 6 9 (J P , A)

特開平 1 0 - 9 1 4 3 8 (J P , A)

加藤工明、外 2 名、長命令語(LIW)コンピュータにおける命令実行遅延方式、電子情報通信学会
論文誌 (J 7 4 - D - 1) , 1 9 9 1 年 9 月 2 5 日 , 第 9 号 , p . 6 1 3 - 6 2 2

(58)調査した分野(Int.Cl. , DB名)

G 0 6 F 9 / 3 8