

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6148191号
(P6148191)

(45) 発行日 平成29年6月14日(2017.6.14)

(24) 登録日 平成29年5月26日(2017.5.26)

(51) Int.Cl.

G06F 9/38 (2006.01)

F 1

G 06 F 9/38 3 1 O F

請求項の数 19 (全 15 頁)

(21) 出願番号 特願2014-51609 (P2014-51609)
 (22) 出願日 平成26年3月14日 (2014.3.14)
 (65) 公開番号 特開2014-182829 (P2014-182829A)
 (43) 公開日 平成26年9月29日 (2014.9.29)
 審査請求日 平成29年2月2日 (2017.2.2)
 (31) 優先権主張番号 13/836,392
 (32) 優先日 平成25年3月15日 (2013.3.15)
 (33) 優先権主張国 米国(US)

早期審査対象出願

(73) 特許権者 390019839
 三星電子株式会社
 Samsung Electronics
 Co., Ltd.
 大韓民国京畿道水原市靈通区三星路129
 129, Samsung-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do, Republic
 of Korea
 (74) 代理人 100110364
 弁理士 実広 信哉

最終頁に続く

(54) 【発明の名称】直接依存命令及びネストされた依存命令のためのスケジューラー内の推測的ソース準備をリセットする効果的な方法

(57) 【特許請求の範囲】

【請求項1】

複数の命令の処理方法において、

依存命令のプロデューサー命令によって前記依存命令のソースを推測的にウェークアップする段階と、

前記依存命令によって、ネストされた依存命令のソースを前記依存命令にしたがって推測的にウェークアップする段階と、

前記依存命令の前記ソースと前記ネストされた依存命令の前記ソースとがアクセス準備になったことを示すように前記依存命令の前記ソースのソース準備ビットをセッティングし、前記ネストされた依存命令の前記ソースのソース準備ビットをセッティングする段階と、

取消命令が前記プロデューサー命令を取消することと予想されるサイクルの回数に基いて、命令スケジューラによる実行に対して前記依存命令の選択の取消を制御するために前記依存命令の前記ソースの取消タイマーをセッティングし、前記命令スケジューラによる実行に対して前記ネストされた依存命令の選択の取消を制御するために前記ネストされた依存命令の前記ソースの取消タイマーをセッティングする段階と、

前記プロデューサー命令の取消指示に対応する受信された取消命令にしたがって前記依存命令の前記ソースの前記取消タイマー及び前記ネストされた依存命令の前記ソースの前記取消タイマーが満了され、前記プロデューサー命令が取消された場合、前記依存命令及び前記ネストされた依存命令を同時に取消する段階と、を含み、

前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとは、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとが所定値に到達した時に満了される複数の命令の処理方法。

【請求項 2】

前記所定値は、0ではない請求項1に記載の複数の命令の処理方法。

【請求項 3】

仮に前記依存命令と前記ネストされた依存命令とが再びウェークアップされなければ、前記依存命令と前記ネストされた依存命令とが前記命令スケジューラによる実行のために選択されることができないことを示すように前記依存命令の前記ソースの前記ソース準備ビットと前記ネストされた依存命令の前記ソースの前記ソース準備ビットとをリセットする段階をさらに含む請求項1に記載の複数の命令の処理方法。

10

【請求項 4】

前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとをリセットする段階をさらに含む請求項3に記載の複数の命令の処理方法。

【請求項 5】

前記リセットする段階は、前記依存命令と前記ネストされた依存命令とを同時に取消する段階が遂行された後に遂行される請求項3に記載の複数の命令の処理方法。

【請求項 6】

前記依存命令が前記プロデューサー命令によって推測的にウェークアップされ、所定時間が経過された後に前記ネストされた依存命令は、前記依存命令によってウェークアップされ、

20

前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとは、前記取消命令が前記プロデューサー命令を取消することと予想される前記サイクルの前記回数に基づいて互いに異なる値を有するようにセッティングされる請求項5に記載の複数の命令の処理方法。

【請求項 7】

前記依存命令が前記プロデューサー命令によって推測的にウェークアップされ、所定時間が経過された後に前記ネストされた依存命令は、前記依存命令によってウェークアップされ、

30

前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとは、前記取消命令が前記プロデューサー命令を取消することと予想される前記サイクルの前記回数に基づいて互いに異なる値を有するようにセッティングされる請求項5に記載の複数の命令の処理方法。

【請求項 8】

前記プロデューサー命令は、推測的に選択される請求項1に記載の複数の命令の処理方法。

【請求項 9】

前記依存命令の前記ソースの前記取消タイマーを設定する段階は、前記プロデューサー命令によって前記依存命令の前記ソースを推測的にウェークアップする時間から前記取消命令が前記プロデューサー命令を取消することと予想されるサイクルの前記回数に基づいて前記依存命令の前記ソースの前記取消タイマーを設定する段階を含み、

40

前記ネストされた依存命令の前記ソースの前記取消タイマーを設定する段階は、前記プロデューサー命令によって前記ネストされた依存命令の前記ソースを推測的にウェークアップする時間から前記取消命令が前記プロデューサー命令を取消することと予想されるサイクルの前記回数に基づいて前記依存命令の前記ソースの前記取消タイマーを設定する段階を含む複数の命令の請求項1に記載の処理方法。

【請求項 10】

スケジューリング動作の遂行のためにプロデューサー命令を選択するように構成される

50

命令スケジューラと、

コントローラと、を含み、

前記コントローラは、

依存命令のプロデューサー命令によって前記依存命令のソースを推測的にウェークアップし、

前記依存命令にしたがってネストされた依存命令のソースを推測的にウェークアップし、

前記依存命令の前記ソースと前記ネストされた依存命令の前記ソースとがアクセス準備になったことを示すように前記依存命令の前記ソースのソース準備ビットをセッティングし、前記ネストされた依存命令の前記ソースのソース準備ビットをセッティングし、

10

取消命令が前記プロデューサー命令を取消することと予想されるサイクルの回数に基づいて前記命令スケジューラによる実行のために前記ネストされた依存命令の選択の取消を制御する前記ネストされた依存命令の前記ソースの取消タイマー及び前記命令スケジューラによる実行のために前記依存命令の選択の取消を制御する前記依存命令の前記ソースの取消タイマーをセッティングし、

前記プロデューサー命令の取消指示に対応する受信された取消命令にしたがって前記プロデューサー命令が取消され、前記依存命令の前記ソースの前記取消タイマー及び前記ネストされた依存命令の前記ソースの前記取消タイマーが満了された場合、前記依存命令及び前記ネストされた依存命令が前記命令スケジューラによる実行のために選択されることができないことを示すように前記依存命令及び前記ネストされた依存命令を同時に取消し、

20

前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとは、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとが所定値に到達した時に満了される命令処理装置。

【請求項 1 1】

前記所定値は、0 ではない請求項 1 0 に記載の命令処理装置。

【請求項 1 2】

前記コントローラは、仮に前記依存命令と前記ネストされた依存命令とが再びウェークアップされなければ、前記依存命令と前記ネストされた依存命令とが前記命令スケジューラによる実行のために選択されることができないことを示すように前記依存命令の前記ソースの前記ソース準備ビットと前記ネストされた依存命令の前記ソースの前記ソース準備ビットとをリセットしり請求項 1 0 に記載の命令処理装置。

30

【請求項 1 3】

前記コントローラは、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとをリセットする請求項 1 2 に記載の命令処理装置。

【請求項 1 4】

前記コントローラは、前記依存命令と前記ネストされた依存命令とを同時に取消した後に前記依存命令の前記ソースの前記ソース準備ビットと前記ネストされた依存命令の前記ソースの前記ソース準備ビットとをリセットする請求項 1 2 に記載の命令処理装置。

40

【請求項 1 5】

前記依存命令が前記プロデューサー命令によって推測的にウェークアップされ、所定時間が経過された後に前記ネストされた依存命令は、前記依存命令によってウェークアップされ、

前記コントローラは、互いに異なる値を有するように、前記取消命令が前記プロデューサー命令を取消すことと予想されるサイクルの前記回数に基づいて、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとをセッティングするように構成される請求項 1 4 に記載の命令処理装置。

【請求項 1 6】

前記依存命令が前記プロデューサー命令によって推測的にウェークアップされ、所定時

50

間が経過された後に前記ネストされた依存命令は、前記依存命令によってウェークアップされ、

前記コントローラは、互いに異なる値を有するように、前記取消命令が前記プロデューサー命令を取消することと予想されるサイクルの前記回数に基づいて、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとをセッティングする請求項 10 に記載の命令処理装置。

【請求項 17】

前記プロデューサー命令は、推測的に選択される請求項 10 に記載の命令処理装置。

【請求項 18】

前記コントローラは、前記プロデューサー命令によって前記依存命令の前記ソースを推測的にウェークアップする時間から前記取消命令が前記プロデューサー命令を取消することと予想される前記サイクルの前記回数に基づいて前記依存命令の前記ソースの前記取消タイマーを判別するように構成され、

10

前記コントローラは、前記プロデューサー命令によって前記ネストされた依存命令の前記ソースを推測的にウェークアップする時間から前記取消命令が前記プロデューサー命令を取消することと予想される前記サイクルの前記回数に基づいて前記ネストされた依存命令の前記ソースの前記取消タイマーを判別するように構成される請求項 10 に記載の命令処理装置。

【請求項 19】

前記依存命令の前記ソースの前記ソース準備ビット、前記ネストされた依存命令の前記ソースの前記ソース準備ビット、前記依存命令の前記ソースの前記取消タイマー、及び前記ネストされた依存命令の前記ソースの前記取消タイマーは、前記命令スケジューラのエントリとしてセッティングされる請求項 10 に記載の命令処理装置。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明の態様は命令 (instructions) を取消し、ソース準備状態 (source-ready state) をリセットする装置及び方法に関し、さらに詳細には取消タイマー (cancel timer) を利用して直接依存命令 (direct dependent instruction) 及びネストされた依存命令 (nested dependent instruction) を取消し、ソース準備ビット (source ready bits) をリセットする装置及び方法に関する。

30

【背景技術】

【0002】

現在のマイクロプロセッサは命令 (instruction) の推測的なスケジューリング (speculative scheduling) 技術を採択している。マイクロプロセッサのスケジューラー内の生成命令 (producer instruction) がロードミス (load misses)、欠陥 (faults)、データ誤謬 (data hazards) 等の原因によって取消される時、前記生成命令によって推測的にウェークアップ (wake up) された“ソース準備 (source ready)” 状態の依存命令は再び選択されることできるようにスケジューラーでリセットされる必要がある。ここで、依存命令 (dependent instructions) はダイレクト命令 (direct instructions) 又は非ダイレクト (indirect) 命令であり得、非ダイレクト命令はネストされた (nested) 命令であり得る。

40

【0003】

従来には生成命令が取消された時、スケジューラーの依存命令の“ソース準備”状態をリセットするための方法として取消された生成命令の目的タグ (destination tags) がスケジューラー内のすべての命令のソースタグ (source tags) と比較する方法が使用された。仮に取消された生成命令の目的タグとスケジュー

50

ラー内のいずれか 1 つの命令のソースタッグとがマッチングされれば、スケジューラーはスケジューラー内の依存命令のソース準備状態をリセットする。

【 0 0 0 4 】

しかし、前記の比較方法はスケジューラー内のすべての命令に対して遂行されなければならないので、比較動作を遂行するのに時間及びパワーが多く消耗され、マイクロプロセッサのリソース (r e s o u r c e) を多く使用する問題がある。

【先行技術文献】

【特許文献】

【 0 0 0 5 】

【特許文献 1】米国特許公開第 2 0 1 0 / 0 3 3 2 8 0 6 号公報

10

【特許文献 2】米国特許第 8 , 2 3 9 , 6 6 1 号公報

【発明の概要】

【発明が解決しようとする課題】

【 0 0 0 6 】

本発明は構造が簡単であり、パワー及び時間が節約される命令処理装置及び方法を提供することにある。

【課題を解決するための手段】

【 0 0 0 7 】

本発明の態様による複数の命令の処理方法は、依存命令の生成命令によって前記依存命令のソースを推測的にウェークアップする段階と、前記依存命令によってネストされた依存命令のソースを前記依存命令にしたがって推測的にウェークアップする段階と、前記依存命令の前記ソースと前記ネストされた依存命令の前記ソースとがアクセス準備になったことを示すように前記依存命令の前記ソースのソース準備ビットをセッティングし、前記ネストされた依存命令の前記ソースのソース準備ビットをセッティングする段階と、前記生成命令が取消されることと予想される取消時間に基づいて前記依存命令の取消タイマーをセッティングし、前記ネストされた依存命令の前記ソースの取消タイマーをセッティングし、前記取消タイマーは、各々前記ソース準備ビットがセッティングされた時、カウントダウンを始まる段階と、仮に前記依存命令の前記ソースの前記取消タイマー及び前記ネストされた依存命令の前記ソースの前記取消タイマーが満了され、前記生成命令が取消された場合に前記依存命令及び前記ネストされた依存命令を同時に取消す段階と、を含む。

20

【 0 0 0 8 】

態様として、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとは、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとが所定値に到達した時に満了される。

【 0 0 0 9 】

態様として、前記所定値は、 0 ではない。

【 0 0 1 0 】

態様として、仮に前記依存命令と前記ネストされた依存命令とが再びウェークアップされなければ、前記依存命令と前記ネストされた依存命令とが選択 (p i c k) される準備にならないことを示すように前記依存命令の前記ソースの前記ソース準備ビットと前記ネストされた依存命令の前記ソースの前記ソース準備ビットとをリセットする段階をさらに含む。

30

【 0 0 1 1 】

態様として、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとをリセットする段階をさらに含む。

【 0 0 1 2 】

態様として、前記リセットする段階は、前記依存命令と前記ネストされた依存命令とを同時に取消す段階が遂行された後に遂行される。

【 0 0 1 3 】

40

50

態様として、前記依存命令が前記生成命令によって推測的にウェークアップされた後、所定時間が経過された後に前記ネストされた依存命令は、前記依存命令によってウェークアップされ、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとは、前記取消時間に基づいて互に異なる値を有するよう にセッティングされる。

【0014】

態様として、前記依存命令が前記生成命令によって推測的にウェークアップされ、所定時間が経過された後に前記ネストされた依存命令は、前記依存命令によってウェークアップされ、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとは、前記取消時間に基づいて互に異なる値を有するよう 10 設定される。

【0015】

態様として、前記生成命令は、推測的に選択される。

【0016】

態様として、前記生成命令が取消されることと予想される前記取消時間は、前記生成命令によって前記依存命令の前記ソースを推測的にウェークアップする時間から前記取消が前記生成命令のために主張されることと予想されるクロックサイクルの数に基づいて決定される。

【0017】

本発明の態様による命令処理装置は、スケジューリング動作の遂行のために生成命令を選択するスケジューラーと、コントローラと、を含み、前記コントローラは、前記依存命令の生成命令によって依存命令のソースを推測的にウェークアップし、前記依存命令によって、ネストされた依存命令のソースを前記依存命令にしたがって推測的にウェークアップし、前記依存命令の前記ソースと前記ネストされた依存命令の前記ソースとがアクセス準備がされたことを示すように、前記依存命令の前記ソースのソース準備ビットをセッティングし、前記ネストされた依存命令の前記ソースのソース準備ビットをセッティングし、前記生成命令が取消されることと予想される取消時間に基づいて前記依存命令の取消タイマーと前記ネストされた依存命令の前記ソースの取消タイマーとが各々前記ソース準備ビットがセッティングされた時、カウントダウンを開始するようにセッティングし、仮に前記依存命令の前記ソースの前記取消タイマー及び前記ネストされた依存命令の前記ソースの前記取消タイマーが満了され、前記生成命令が取消された場合に前記依存命令及び前記ネストされた依存命令を同時に取消す。

【0018】

態様として、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとは、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとが所定値に到達した時に満了される。

【0019】

態様として、前記所定値は、0 ではない。

【0020】

態様として、前記コントローラは、仮に前記依存命令と前記ネストされた依存命令とが再びウェークアップされなければ、前記依存命令と前記ネストされた依存命令とが選択 (pick) される準備にならないことを示すように前記依存命令の前記ソースの前記ソース準備ビットと前記ネストされた依存命令の前記ソースの前記ソース準備ビットとをリセットする。

【0021】

態様として、前記コントローラは、前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとをリセットする。

【0022】

態様として、前記コントローラは、前記依存命令と前記ネストされた依存命令とを同時

10

20

30

40

50

に取消した後に前記依存命令の前記ソースの前記ソース準備ビットと前記ネストされた依存命令の前記ソースの前記ソース準備ビットとをリセットする。

【0023】

態様として、前記依存命令が前記生成命令によって推測的にウェークアップされ、所定時間が経過された後に前記ネストされた依存命令は、前記依存命令によってウェークアップされ、前記コントローラは、互に異なる値を有するように前記取消時間に基づいて前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとをセッティングする。

【0024】

態様として、前記依存命令が前記生成命令によって推測的にウェークアップされ、所定時間が経過された後に前記ネストされた依存命令は、前記依存命令によってウェークアップされ、前記コントローラは、互に異なる値を有するように前記取消時間に基づいて前記依存命令の前記ソースの前記取消タイマーと前記ネストされた依存命令の前記ソースの前記取消タイマーとをセッティングする。

10

【0025】

態様として、前記生成命令は、推測的に選択される。

【0026】

態様として、前記コントローラは、クロックサイクルの数に基づいて前記生成命令によって前記依存命令の前記ソースを推測的にウェークアップする時間から前記生成命令が取消されることと予想される前記取消時間を決定する。

20

【0027】

態様として、前記依存命令の前記ソースの前記ソース準備ビット、前記ネストされた依存命令の前記ソースの前記ソース準備ビット、前記依存命令の前記ソースの前記取消タイマー、及び前記ネストされた依存命令の前記ソースの前記取消タイマーは、前記スケジューラーのエントリにセッティングされる。

【発明の効果】

【0028】

本発明の技術的思想にしたがう態様は依存命令のメイン生成命令から1つの取消指示(`cancel indication`)を使用することによって、簡単であり、パワー及び時間を節約することができる依存命令及びネストされた命令を同時に取消し、依存命令のソース-準備状態をリセットする方法を提供する。

30

【図面の簡単な説明】

【0029】

【図1】本発明の実施形態によるプロセッサの一部構造を示す図面である。

【図2】本発明の実施形態による動作実行スケジューリングのためのプロセッサのスケジューラー内の命令のエントリを示す図面である。

【図3】本発明の実施形態によるソース準備ビットのセッティング及びリセッティング処理を示す図面である。

【図4】本発明の実施形態による各々取消タイマーとして動作する2つの依存命令及び1つの生成命令を含む命令パイプラインを示す図面である。

40

【図5】本発明の技術的思想の実施形態による図4の取消及びリセッティング動作を説明する順序図である。

【発明を実施するための形態】

【0030】

本発明の長所及び特徴、そしてそれを達成する方法は添付される図面と共に詳細に後術されている実施形態を通じて説明される。しかし、本発明はここで説明される実施形態に限定されなく、他の形態に具体化されることもあり得る。本実施形態は本発明が属する技術分野で通常の知識を有する者に本発明の技術的思想を容易に実施できるように詳細に説明するために提供されることである。

【0031】

50

図面において、本発明の実施形態は図示された特定形態に制限されることではなく説明を明確するために誇張されたことである。また、明細書の全体に掛けて同一の参照番号で表示された部分は同一の構成要素を示す。本明細書で‘及び／又は’という表現は前後に羅列された構成要素の中で少なくとも1つを含む意味に使用される。また、‘連結される／結合される’という表現は他の構成要素と直接的に連結させるか、或いは他の構成要素を通じて間接的に連結されることを含む意味として使用される。本明細書で単数形は文句で特別に言及しない限り複数形も含む。また、明細書で使用される‘含む’又は‘包含する’と言及された構成要素、段階、動作、及び素子は1つ以上の他の構成要素、段階、動作、素子、及び装置の存在又は追加を意味する。

【0032】

10

図1は本発明の実施形態によるプロセッサの一部構造を示す図面である。図1のプロセッサはデコーダー101、再命名器(rename器)102、スケジューラー103、及びコントローラ104を含む。図1のプロセッサは多様なタイプのプロセッサに適用され得る。例えば、本発明の実施形態において、図1のプロセッサはマルチ-コアプロセッサ(multiple-core processor)に適用され得る。図1のプロセッサはフェッチャー(fetcher)、キャッシュ(cache)、遂行ユニット等のように広く公知された構造をさらに包含することができ、本図面では本発明の実施形態を明確に説明するために前記広く公知された構造が省略された。

【0033】

20

図1のプロセッサにおいて、メモリ(図示せず)からの命令がフェッチャー(図示せず)によってデコーダー101にフェッチ(fetch)される。デコーダー101は命令を下位レベル、具体的なマイクロ動作(micro-operations; 以下、micro-ops)によってデコーディングする。したがって、命令のストリーム(stream)がマイクロ動作(micro-ops)にマッピングされる。デコーディングされたマイクロ動作は複数のソース(source)又は複数の目的地(destinations)の名前を変更する再命名器102に送られ、以後スケジュール動作を遂行するようにスケジューラー103に送られる。1つのマイクロ動作は一般的に2つ又は3つのソースオペランド(source operand、又は原始オペランド)を有し、1つ又は2つの目的地オペランド(destination operand)を有する。ソースオペランドはソースデータ(source data)を含むレジスターを明示するのに使用され、目的地オペランドはマイクロ動作の結果が書き込まれたレジスターを明示するのに使用される。再命名器102はマイクロ動作のソースレジスター(source registers)及び／又は目的地レジスター(destination registers)の名前を変更する。レジスター名前変更は、例えばマイクロ動作の故障の時に読み出し後書き込み危険(write-after-read hazards)、書き込み後書き込み危険(write-after-write hazards)のようなデータ危険(data hazards)を無くすための技術である。スケジューラー103はマイクロ動作を選択し、マイクロ動作順序の本来の順序にしたがうか、或いはしたがわない順序にマイクロ動作の実行をスケジューリングする。コントローラ104はスケジューラー103の全般的な動作を制御する。たとえコントローラ104が1つの分離された要素として図示されているが、コントローラ104はスケジューラー103と結合された1つの要素として具現され得る。

【0034】

30

図2は本発明の実施形態による動作実行スケジューリングのためのプロセッサのスケジューラー内の命令のエントリ(entry)を示す図面である。ここで、スケジューラーは図1で言及されたスケジューラー103であり得る。

【0035】

40

図2に示したように、本発明の実施形態にしたがって、生成命令100、生成命令101、生成命令100、101に依存する依存命令200がスケジューラー103内に位置する。たとえ図2では単なる生成命令100、101のみが図示されているが、依存命令

50

200は依存命令200のソース(sources)の数にしたがって他の生成命令に依存することができ、依存命令200の他のソースは他の生成命令に依存することができる。依存命令200がスケジューラー103内に在る時、スケジューラー103は本発明の実施形態にしたがって依存命令200のソース当たりソース準備フィールド(source ready field)15及び取消タイマーフィールド(cancel timer field)16を割当する。ソース準備フィールド15はソース準備ビット(source ready field)を含み、取消タイマーフィールド16は取消時間(cancel timer)又は取消時間値(cancel timer value)を含む。本実施形態において、ソース準備ビットと取消タイマーとは依存命令にこの2つのフィールドが割当された時点にはセット(set)されなく、“0”を維持する。しかし、ソース準備ビットは準備(ready)状態に、例えば“1”にセッティングされ、取消タイマーは生成命令が推測的に(speculatively)依存命令をウェークアップ(wake up)する時、以下で説明されるようにコントローラ104によって所定の値(predetermined value)にセッティングされる。10

【0036】

依存命令200が生成命令100、101によって推測的にウェークアップされ(又は、例えば、前記説明されたように依存命令200のソースの数に基づいた追加的な生成命令によって推測的にウェークアップされる)、ソースディペンダンシー(source dependency)がソース命令200の各ソースと合う時、コントローラ104はスケジューラー103内の依存命令の各ソースのためのソース準備フィールド15のソース準備ビットをセットする。例えば、生成命令100、101が推測的に依存命令200をウェークアップし、ソースディペンダンシーが依存命令200の各ソースと合う時、コントローラ104は各ソースのソースデータが何回のサイクル(cycles)の間に利用可能であることを示すように各ソースのソース準備ビットを“1”的値にセットする。そうでない場合に各ソースのソース準備ビットの値は“0”を維持する。この場合、例えば、ソースディペンダンシーが依存命令200の各ソースと互に異なる時点に合うことがあり得る。依存命令のすべてのソースに対してソース準備ビットが“1”的値にセットされれば、依存命令100は実行のためにスケジューラー103から選択する準備になったことと考慮される。20

【0037】

ソース準備ビットのセッティングに加えて、生成命令100、101が依存命令200を推測的にウェークアップし、ソースディペンダンシーが依存命令200の各ソースと合う時、コントローラ204は依存命令200の各ソースのための取消タイマーフィールド16内の取消タイマーを所定値にセットする。取消タイマーはコントローラ104によって依存命令200の取消時間(又は取消タイミング)を制御するのに使用され、取消された依存命令200の各ソースのソース準備ビットをリセット(reset)するのに使用される。取消タイマーに対して設定された予め定まれた所定時間値が経過され、生成命令100がスケジューリングから取消される時、依存命令200は依存命令200が取消され、依存命令200の各ソースのためのソース準備ビットと取消タイマーとはリセットされて依存命令200は後に再び選択されて利用することができるようになる。ここで、本発明の実施形態による取消タイマーの予め定まれた所定時間は生成命令100を取消すのに必要とすることと予想される時間に対応するサイクルに該当するように設定され得る。30

【0038】

生成命令100、101に依存する依存命令200の各ソースに対する予め定まれた時間値をセットする取消タイマーはソース準備ビットが依存命令のソースのためにセットされれば、時間が流れ始める。仮に取消タイマーが満了される時、生成命令が取消されれば、以後の依存命令は取消され、依存命令の各ソースに対するソース準備ビットと取消タイマーとはリセットされる。仮に生成命令が取消されなく、実行のためにスケジュールされれば、取消タイマーは依存命令のスケジューリングに影響を及ぼない。40

【0039】

本発明の実施形態において、依存命令200を取消すために言及された取消タイマーはスケジューラー103内の依存命令200の各ソースをセットすることと説明された。しかし、取消タイマーが命令の取消のための目的にセットされる限り、取消タイマーは命令の各ソースのためにセットするのに使用されないことがあり、スケジューラー103のリソース(resource)を節約するために各命令をセットするのに使用され得る。

【0040】

図3は本発明の実施形態によるソース準備ビットのセッティング及びリセッティング処理を示す図面である。

【0041】

ここで、生成命令(producer-1)及び依存命令は図1で説明されたプロセッサのスケジューラー103内に位置する。この実施形態において、依存命令は1つのシングルソースを有することと仮定される。また、この実施形態において、ソースのための取消タイマーは依存命令がウェークアップされる時点から生成命令(producer-1)が取消されるのに必要である所要される時間に対応するサイクルの数に基づいてセットされる。取消タイマーはまた予め定まれた取消タイマーの満了値にしたがってセットされ、これは取消タイマーが予め定まれた最小タイマー満了値に到達する時、生成命令(producer-1)によって要請された取消動作が意図された自分の命令取消動作であることを判断することを可能であるようにする。言い換えれば、生成命令の取消タイマーは取消タイマーが予め定まれた取消タイマー満了値に到達する時、満了される。したがって、仮に生成命令(producer-1)の取消のために所要されるサイクルの数がN-1であれば、取消タイマーはNに設定される。例えば、取消のために所要されるサイクルが3サイクル(例えば、N-1サイクル)であれば、取消タイマーは4(例えば、Nサイクル)に設定され得、予め定まれた最小タイマーの満了値は“1”である。したがって、取消タイマーが予め定まれた取消タイマー満了値“1”に到達する時、例えば、N-1サイクルが経過され、生成命令(producer-1)が取消される時、依存命令は同様に取消され、以後、ソースのソース準備ビットはリセットされる。前記の例を使用して、仮に3サイクル(例えば、N-1サイクル)が経過され、取消タイマーが初期に4(例えば、N)にセッティングされれば、予め定まれた取消タイマーの満了値は“1”に到達する。予め定まれた取消タイマー満了値に到達する時、生成命令(producer-1)は要請された取消が自分に対する意図された取消命令であることを判断することができる。たとえ前記の実施形態において、予め定まれた取消タイマーの満了値が“1”に選択されたが、他の0ではない(non-zero)取消タイマー満了値が選択され得る。例えば、仮に取消するのに所要されるサイクルが3サイクル(例えば、N-2サイクル)であれば、取消タイマーは5(例えば、N)に設定され得る。取消タイマーが3サイクルの以後に“2”的に到達する時、即ち予め定まれた取消タイマー満了値である“2”的に到達する時、生成命令(producer-1)は要請された取消が自らに対する取消命令であることを判断することができる。ここで、Nの値は所定の自然数であり得る。

【0042】

図3を参照すれば、図1のプロセッサのクロックサイクルの観点でスケジューラー103内のソース準備ビットと取消タイマーセッティング及びリセッティング処理動作が下のように説明される。

【0043】

サイクル0(Cycle0)で、生成命令(producer-1)と依存命令とがスケジューラー103内に位置し、ソース準備フィールドと取消タイマーフィールドとが依存命令に割当される。仮に生成命令(producer-1)がスケジューラー103で選択されなければ、スケジューラー103の各フィールド内のソース準備ビットと取消タイマーとはセットされなく、したがって、それらの値は“0”を維持する。サイクル0の間に、仮に生成命令(producer-1)がスケジューラー103で選択され、生成命令(producer-1)が推測的に依存命令をウェークアップすれば、ソース準備

10

20

30

40

50

ビットがセットされ、ソース準備ビットがセットされることと同時又はその直前や直後に取消タイマーもやはりサイクル0が終了される際にはセットされる。

【0044】

サイクル1(Cycle1)で、生成命令(producer-1)が選択され、生成命令(producer-1)が推測的に依存命令をウェークアップしたので、依存命令のソースのためのソース準備ビットは“1”にセットされて依存命令が選択される準備になったことを示す。また、取消タイマーもやはり依存命令のソースのためにNにセットされ、予め定まれた取消タイマーの満了値は“1”であり得る。ソースのための取消タイマーNはサイクルの数に基づき、サイクルの数は先に説明されたように予め定まれた取消タイマー満了値の観点で生成命令(producer-1)が取消されるのに所要される時間に対応する。取消タイマーはウェークアップする(waking up)時点でからカウントダウン(count down)を開始し、予め定まれた取消タイマーの満了値に到達する時に満了される。例えば、N-1サイクルが超過される時、取消タイマーによって予め定まれた取消タイマーの満了値“1”に到達したことを示される。

10

【0045】

サイクルN(CycleN)で、仮に取消タイマーが満了(例えば、予め定まれた取消タイマー満了値に到達)され、生成命令(producer-1)に対する取消が発生すれば、依存命令は再びウェークアップされる時まで選択される準備にならなかった状態になる。したがって、サイクルNが終了される時に、依存命令のソースのためのソース準備ビットと取消タイマーとは、例えば、“0”的にリセットされて依存命令が再びウェークアップされる時まで選択される準備にならなかった状態であることを示す。

20

【0046】

サイクルN+1(CycleN+1)で、ソース準備ビットと取消タイマーとは“0”を維持するので、依存命令は再びウェークアップされない。

【0047】

前記本発明の実施形態は依存命令のソースのためのソース準備ビットをリセットするために単なるソース当たり小さい容量の取消タイマーを必要とする。これはソース準備セットをリセットするために従来に要求された最小になった命令の目的地タグとそれらの依存命令とをスケジューラー103内のすべての命令のソースタグと比較しなければならない多い数の比較動作が必要としないこと意味する。

30

【0048】

一方、本発明の技術的な思想の実施形態は図4で説明されるように生成命令の直接依存命令(direct dependent instruction)のみでなく、非直接依存命令(indirect dependent instruction)に対しても適用され得る。非直接依存命令はシャドー依存命令(shadow dependent instruction)であると称される。

【0049】

図4は本発明の実施形態による各々の取消タイマーとして動作する2つの依存命令及び1つの生成命令を含む命令パイプラインを示す図面である。

【0050】

40

図4で、パイプラインは生成命令である生成命令A(LDR R1, [R0])、生成命令Aに依存する直接依存命令である依存命令B(ADD R2, R1, 5)、依存命令Bに依存する非直接依存命令であるシャドー依存命令C(ADD R3, R2, 10)を含む。非直接依存命令であるシャドー依存命令Cはまたネストされた依存命令であると称される。したがって、シャドー依存命令Cは生成命令Aに対してシャドーディペンダンシー(shadow dependency)又はネストされたディペンダンシー(nested dependency)を有する。依存命令Bとシャドー依存命令Cとは取消動作を直接的にトリガー(trigger)しないシングルサイクルレイテンシ命令(single cycle latency instruction)であることと描写され、反面に生成命令Aはサイクル#5(cycle#5)で取消されることができるマル

50

チサイクルレイテンシ命令 (multi cycle latency instruction) であることと描写される。

【0051】

以下では、本発明の技術的思想の実施形態にしたがって図4に図示された命令パイプライン内の依存命令の取消動作及び依存命令のためのソース準備ビットのリセット動作が図1乃至図3を参照して説明される。図5は本発明の技術的思想の実施形態による図4の取消及びリセッティング動作を説明する順序図である。

【0052】

本実施形態において、依存命令Bとシャドー依存命令Cとは各々1つのシングルソースを有していることと仮定され、予め定まれた所定の取消タイマー値は“1”であると仮定される。図5に示したように、サイクル#3で、生成命令A (LDR R1, [R0]) は推測的に依存命令B (ADD R2, R1, 5) をウェークアップして依存命令BがソースデータR1をフォワーディングソースパス (forwarding path) から獲得 (grab) するようにする (S100段階)。

10

【0053】

その後、同一のサイクル#3で、ソースのためのソース準備ビットと取消タイマーとが依存命令Bのためにセットされる (S200段階)。依存命令Bのための取消タイマーは生成命令Aによって依存命令Bがウェークアップされる時点から生成命令Aが取消されるのに所要される時間に対応するサイクルの数に基づいて設定される。本発明の実施形態において、取消タイマーはサイクル#5に相応する“2”に設定される。依存命令Bのためのソース準備ビット及び取消タイマーが設定された直後、依存命令Bのための取消タイマーは時間が流れ始める。

20

【0054】

以後、依存命令Bはシャドー依存命令C (ADD R3, R2, 10) をサイクル#4でウェークアップしてフォワーディングパスからソースデータR2を獲得 (grab) するようにする (S300段階)。

【0055】

その後、サイクル#4で、ソースのためのソース準備ビットと取消タイマーとがシャドー依存命令Cのためにセットされる (S400段階)。シャドー依存命令Cのための取消タイマーは依存命令Bによってシャドー依存命令Cがウェークアップされた時点から生成命令Aによって取消動作が遂行されるのに所要される時間に相応するサイクルの数に基づいて設定される。本実施形態で、取消タイマーはサイクル#5に相応する“1”に設定される。シャドー依存命令Cのためにソース準備ビットと取消タイマーとが設定された直後、シャドー依存命令Cのソースのための取消タイマーは時間が流れ始める。

30

【0056】

以後、サイクル#5で、生成命令Aが取消されたか、そして依存命令B及びシャドー依存命令Cのための取消タイマーが相応するサイクル#5で予め定まれた取消タイマー満了値“1”に相応するか否かが判断される (S500段階)。

【0057】

仮に生成命令Aが取消され、取消タイマーが相応するサイクル#5で予め定まれた値“1”に対応すれば、依存命令B及びシャドー依存命令Cは全て同時に取消され、依存命令B及びシャドー依存命令Cのためのソース準備ビットと取消タイマーはリセットされる (S600段階)。しかし、仮に生成命令Aが取消サイクル#5で取消されなければ、生成命令Aと依存命令B及びシャドー依存命令Cを含んで生成命令Aの依存命令が遂行される (S700段階)。

40

【0058】

前記例示的に説明された実施形態による命令の取消及びソースリセットスキームは単なる各依存命令の取消タイマー及びそれらの基礎生成命令 (例えば、図4の生成命令A) の取消因子 (cancel indicator) を観察することによって複数の依存命令及び/又は複数のシャドー依存命令に同時に適用され得る。前記の実施形態において、仮

50

に生成命令に対する取消要請があり、依存命令のための取消タイマーが予め定まれた取消タイマー値を満足すれば、すべての依存命令及びネストされた依存命令が取消され、それらのソース準備ビット及び取消タイマーが実質的に同時にリセットされる。命令取消に基づいた取消タイマー及びソース準備ビットリセットのこのようないくつかの簡単な接近方法は命令タグ及びそれらの依存命令タグを比較しなければならない従来の高費用の比較方法の必要性を消滅させ、したがって、プロセッサの命令プロセスによるパワー消費を最小化する。

【0059】

上述した説明は例示的なこととして理解されなければならなく、本発明の技術的思想はこれに限定されない。例えは、ソース準備ビットと取消タイマーとの例示的な値は何らかの数としても設定され得る。ソース準備ビット及び取消タイマーの実質的なビットサイズ(又は幅)は何らかの値としても設定され得る。また、先に説明されたサイクルの数はやはり何らかの数としても設定され得る。さらに、前記の例が推測的な命令(s p e c u l a t i v e i n s t r u c t i o n s)の主要実施形態として説明されたが、本発明で開示された内容はこれに限定されなく、他のタイプの命令に対しても適用され得る。また、前記の実施形態で依存命令が1つのシングルソースを有することと仮定されたが、本発明の技術的思想は2つ又はその以上のソースを有する依存命令に対しても適用され得る。このような変更、応用、追加、改善は発明の技術的な思想の範囲にあるはずである。したがって、本発明の技術的な思想は前記の実施形態に制限されることではなく、本出願の請求項及び請求項の範囲内の他の実施形態やはり本発明の技術的な思想に含まれることが理解できる。

【0060】

本発明の実施形態を説明するために使用された用語の中で“1つ”又は“シングル”という用語はそれが明確に制限する場合を除いては単数のことの以外に複数の要素を指すこともあり得る。本発明の技術的な思想を逸脱しない範囲内で他の類似であるか、或いは非類似な変更が可能である。したがって、本発明の技術的な思想は添付された請求項のみでなく、異なる方式の請求項が可能である。

【符号の説明】

【0061】

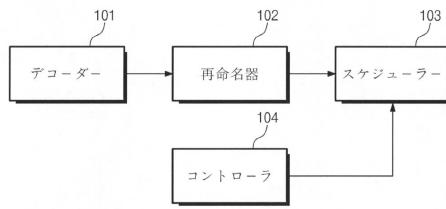
101	・	・	・	デコーダー
102	・	・	・	再命名器
103	・	・	・	スケジューラー
104	・	・	・	コントローラ
Entry	・	・	・	エントリ
producer	instruction	・	・	生成命令
dependent	instruction	・	・	依存命令
source	ready	field	・	ソース準備フィールド
cancel	timer	field	・	取消タイマーフィールド

10

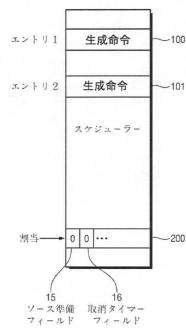
20

30

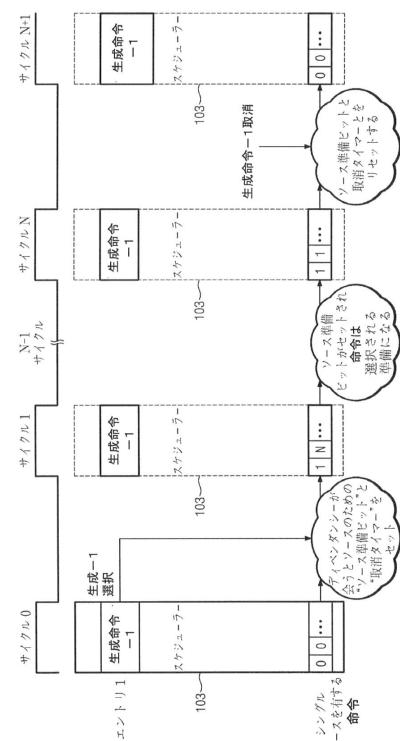
【図1】



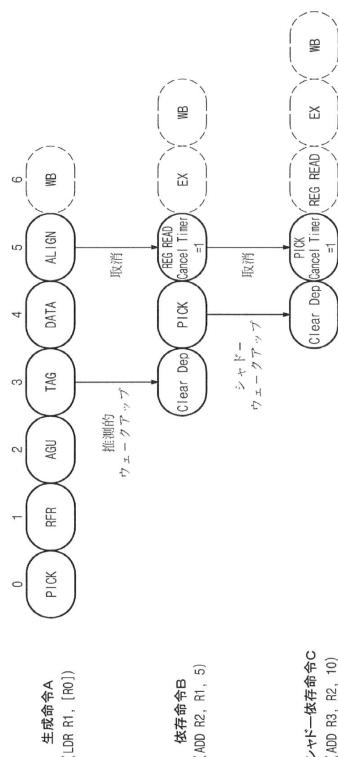
【図2】



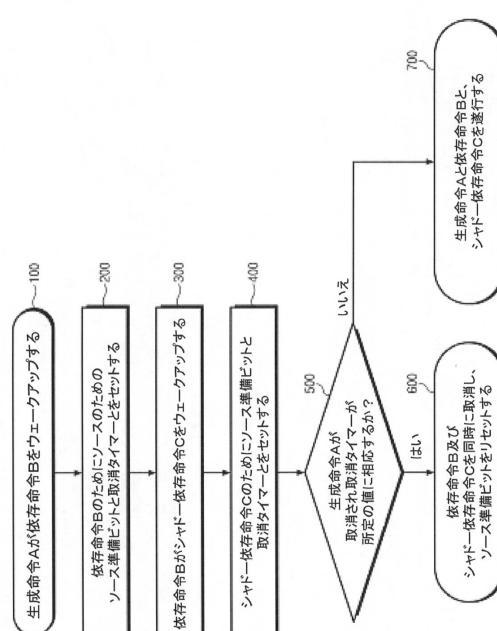
【図3】



【図4】



【図5】



フロントページの続き

(72)発明者 ラヴィ・イエンガー

アメリカ合衆国・テキサス・78759・オースティン・セトン・センター・パークウェイ・46
00・アパートメント・1712

(72)発明者 ブラッドリー・ジーン・バージェス

アメリカ合衆国・テキサス・78745・オースティン・サンセット・トレイル・21

(72)発明者 サンディープ・クマール・ドゥベイ

アメリカ合衆国・テキサス・78730・オースティン・エフエム・2222・10301・アパートメント・623

審査官 大桃 由紀雄

(56)参考文献 米国特許出願公開第2007/0204135(US, A1)

米国特許出願公開第2011/0078697(US, A1)

米国特許第6792524(US, B1)

特開2009-181163(JP, A)

特開2006-228241(JP, A)

特開2008-139964(JP, A)

(58)調査した分野(Int.Cl., DB名)

G 06 F 9 / 38