

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-26583

(P2010-26583A)

(43) 公開日 平成22年2月4日(2010.2.4)

(51) Int.Cl.	F I	テーマコード (参考)
G 0 6 F 9/38 (2006.01)	G 0 6 F 9/38 3 1 0 E	5 B 0 0 5
G 0 6 F 12/08 (2006.01)	G 0 6 F 12/08 5 6 5	5 B 0 1 3
	G 0 6 F 12/08 5 4 3 B	

審査請求 有 請求項の数 15 O L (全 17 頁)

(21) 出願番号	特願2008-183828 (P2008-183828)	(71) 出願人	596053079
(22) 出願日	平成20年7月15日 (2008.7.15)		広島市
			広島県広島市中区国泰寺町1丁目6番34号
		(71) 出願人	396023993
			株式会社半導体理工学研究センター
			神奈川県横浜市港北区新横浜3丁目17番地2 友泉新横浜ビル6階
		(74) 代理人	100101454
			弁理士 山田 卓二
		(74) 代理人	100081422
			弁理士 田中 光雄
		(74) 代理人	100098280
			弁理士 石野 正弘

最終頁に続く

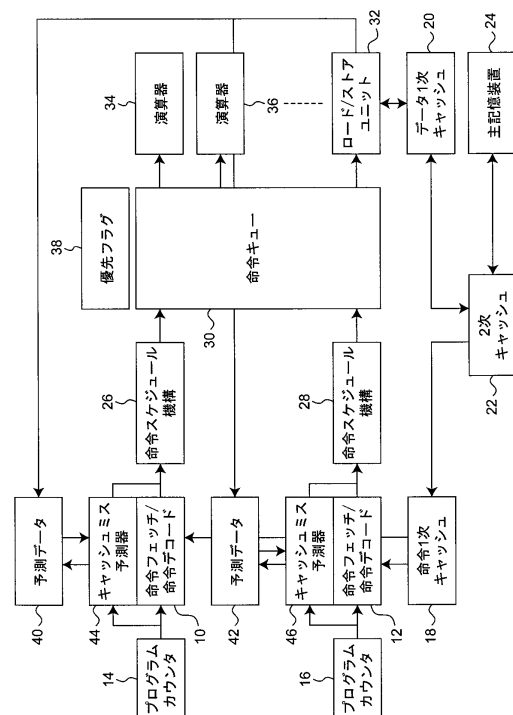
(54) 【発明の名称】 プロセッサ

(57) 【要約】

【課題】複数のプログラムを同時に実行するプロセッサにおいて、キャッシュミスによる影響をより効果的に抑える。

【解決手段】複数のプログラムを同時に実行するプロセッサにおいて、主記憶装置またはキャッシュから第1と第2のプログラムの命令語が読み出され、読み出された命令語は第1の記憶手段にキューイングされる。第2と第3の記憶手段は、それぞれ、第1と第2のプログラムに含まれる命令語の実行に要するサイクル数見積りのための実行サイクル数見積り情報を記憶し、第1と第2のキャッシュミス予測器は、それぞれ、実行サイクル数見積り情報をそれぞれ読み出す。命令語のスケジューリング位置は、この実行サイクル数見積り情報に従って決定される。好ましくは、実行サイクル数見積り情報は、キャッシュミスの検出/非検出に応じて動的に変更される。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

複数のプログラムを同時に実行するプロセッサであって、

前記複数のプログラムの中の第 1 のプログラムの実行位置を示す第 1 のプログラムカウンタと、

前記複数のプログラムの中の第 2 のプログラムの実行位置を示す第 2 のプログラムカウンタと、

第 1 のプログラムカウンタの内容により主記憶装置またはキャッシュから第 1 のプログラムの命令語を読み出す第 1 の命令読み出し機構と、

第 2 のプログラムカウンタの内容により主記憶装置またはキャッシュから第 2 のプログラムの命令語を読み出す第 2 の命令読み出し機構と、

第 1 及び第 2 の命令読み出し機構により読み出された命令語をキューイングするための第 1 の記憶手段と、

第 1 の記憶手段から取り出した命令語のうちロード機能を有する命令を、キャッシュにアクセスして実行するキャッシュ制御機構と、

第 1 の記憶手段から取り出した命令語により、ロード機能により取り出したデータを用いて演算を行う演算機構と、

第 1 のプログラムに含まれる命令語の実行に要するサイクル数見積りのための実行サイクル数見積り情報を記憶する第 2 の記憶手段と、

第 2 のプログラムに含まれる命令語の実行に要するサイクル数見積りのための実行サイクル数見積り情報を記憶する第 3 の記憶手段と、

第 1 のプログラムカウンタの内容により第 2 の記憶手段のエントリ位置を特定して実行サイクル数見積り情報を読み出す第 1 のキャッシュミス予測器と、

第 2 のプログラムカウンタの内容により第 3 の記憶手段のエントリ位置を特定して実行サイクル数見積り情報を読み出す第 2 のキャッシュミス予測器と、

第 1 および第 2 の命令読み出し機構から入力される命令語を第 1 の記憶手段にキューイングする命令スケジュール機構であって、前記命令語が、前記キャッシュ制御機構によって処理される第 1 の命令語の実行結果を必要とし前記演算機構によって処理される第 2 の命令語である場合、前記第 1 の命令語に対応する、第 1 または第 2 のキャッシュミス予測器からの前記実行サイクル数見積り情報の内容に従って、第 1 の記憶手段における前記第 2 の命令語のスケジューリング位置を決定し、一方、前記実行サイクル数見積り情報に従うとスケジューリング可能な位置が存在しない場合には前記第 2 の命令語のスケジューリングを保留する命令スケジュール機構と

を備えることを特徴とするプロセッサ。

【請求項 2】

請求項 1 に記載されたプロセッサであって、

前記キャッシュ制御機構は、第 1 の記憶手段から取り出した命令語についてキャッシュミスを検出した場合には、前記命令語の実行サイクル数見積り情報を記憶している第 2 または第 3 の記憶手段における前記命令語のエントリ位置に、キャッシュミス検出情報を送り、第 2 または第 3 の記憶手段は、前記キャッシュミス検出情報を受け取ると、前記命令語の実行サイクル数見積り情報を増加させまたは最大値に飽和させ、前記キャッシュ制御機構は、前記命令語についてキャッシュミスを検出しない場合には、第 2 または第 3 の記憶手段の前記命令語のエントリ位置に、キャッシュミス非検出情報を送り、第 2 または第 3 の記憶手段は、前記キャッシュミス非検出情報を受け取ると、前記命令語の実行サイクル数見積り情報を減少させまたは最小値に飽和させることを特徴とするプロセッサ。

【請求項 3】

請求項 2 に記載されたプロセッサであって、

第 1 の記憶手段は、キューイングされる命令語を、その命令語がどのプログラムに属するかを示す情報とともに記憶し、

さらに、第 1 および第 2 のプログラムの中の 1 つのプログラムに属する命令語が一定時

10

20

30

40

50

間に実行された数を測定する第 1 の測定手段を備え、

前記第 1 の測定手段により測定された実行数がしきい値より小さい場合に、前記第 2 および第 3 の記憶手段の中の方、記憶されている命令語の実行サイクル数見積り情報を増加することを特徴とするプロセッサ。

【請求項 4】

請求項 1 または 2 に記載されたプロセッサであって、

さらに、第 1 のプログラムと第 2 のプログラムのうちいずれを優先的に実行するかを指定する情報を記憶し、前記キャッシュ制御機構に送る第 4 の記憶手段を備え、

第 1 の記憶手段は、キューイングされる命令語を、その命令語がどのプログラムに属するかを示す情報とともに記憶し、

さらに、第 4 の記憶手段によって指定されたプログラムが走行中である場合における、第 4 の記憶手段によって指定されずかつキャッシュミスの発生頻度がしきい値より低いプログラムの走行時間と、第 4 の記憶手段によって指定されずかつキャッシュミスの発生頻度がしきい値より高いプログラムの走行時間との第 1 の比と、第 4 の記憶手段によって指定されたプログラムが走行中でない場合における、第 4 の記憶手段によって指定されずかつキャッシュミスの発生頻度がしきい値より低い前記プログラムの走行時間と、第 4 の記憶手段によって指定されずかつキャッシュミスの発生頻度がしきい値より高い前記プログラムの走行時間との第 2 の比を測定する第 2 の測定手段を備え、

前記命令スケジュール機構は、第 1 の比が第 2 の比より小さいと第 2 の測定手段が測定したとき、第 1 の比を第 2 の比よりも大きくするようにスケジューリングを行うことを特徴とするプロセッサ。

【請求項 5】

請求項 1 または 2 に記載されたプロセッサであって、

第 1 の記憶手段は、キューイングされる命令語を、その命令語がどのプログラムに属するかを示す情報とともに記憶し、

さらに、第 1 のプログラムと第 2 のプログラムのうちいずれを優先的に実行するかを指定する情報を記憶し、前記キャッシュ制御機構に送る第 4 の記憶手段と、

第 4 の記憶手段によって指定されたプログラムに属する命令語が一定時間に実行された数を測定する第 3 の測定手段を備え、

前記第 3 の測定手段は、測定された前記実行数がしきい値より低い場合に、当該事象をイベントにより通知することを特徴とするプロセッサ。

【請求項 6】

請求項 1 または 2 に記載されたプロセッサであって、

第 1 の記憶手段は、キューイングされる命令語を、その命令語がどのプログラムに属するかを示す情報とともに記憶し、

さらに、第 1 のプログラムと第 2 のプログラムのうちいずれを優先的に実行するかを指定する情報を記憶し、前記キャッシュ制御機構に送る第 4 の記憶手段と、

第 4 の記憶手段によって指定されたプログラムに属する命令語が一定時間に実行された数を測定する第 3 の測定手段と、

第 4 の記憶手段によって指定されないプログラムに属する命令語の実行を一時的に停止させる停止手段とを備え、

前記第 3 の測定手段は、測定された前記実行数がしきい値より低い場合には、前記停止手段に、第 4 の記憶手段によって指定されないプログラムの実行を一時的に停止させることを特徴とするプロセッサ。

【請求項 7】

請求項 1 または 2 に記載されたプロセッサであって、

第 1 の記憶手段は、キューイングされる命令語を、その命令語がどのプログラムに属するかを示す情報とともに記憶し、

さらに、複数のプログラムのうちいずれを優先的に実行するかを指定する情報を記憶し、前記キャッシュ制御機構に送る第 4 の記憶手段と、

10

20

30

40

50

第４の記憶手段によって指定されないプログラムに属する命令語が一定時間に実行された数を測定する第４の測定手段と、

第４の記憶手段によって指定されないプログラムごとに命令語の実行を一時的に停止させる停止手段を備え、

前記第４の測定手段は、測定された前記実行数がしきい値より低い場合には、前記停止手段に、指定されない個々のプログラムの実行を一時的に停止させることを特徴とするプロセッサ

【請求項８】

請求項１または２に記載されたプロセッサであって、

さらに、第１のプログラムと第２のプログラムのうちいずれを優先的に実行するかを指定する情報を記憶し、前記キャッシュ制御機構に送る第４の記憶手段を備え、

第１の記憶手段は、キューイングされる命令語を、その命令語がどのプログラムに属するかを示す情報とともに記憶し、

前記キャッシュ制御機構は、第４の記憶手段によって指定されたプログラムに属さない命令語を実行し、かつ、その実行においてキャッシュミスを検知した場合、キャッシュミス検知情報を前記命令スケジュール機構に通知し、

前記命令スケジュール機構は、前記キャッシュミス検知情報を受け取った場合に、第１の記憶手段において、実行された前記命令語が属するプログラムにおいて、前記命令語の実行結果を必要とする命令語を含む後続命令を特定し、全て消去する

ことを特徴とするプロセッサ。

【請求項９】

請求項８に記載されたプロセッサであって、

前記命令スケジュール機構は、前記キャッシュミスが解消されたとの情報を前記キャッシュ制御機構から通知されると、第１の記憶手段に対して、消去した前記後続命令のキューイングを再開することを特徴とするプロセッサ。

【請求項１０】

請求項８に記載されたプロセッサであって、

前記キャッシュ制御機構が、第４の記憶手段によって指定されたプログラムに属さない前記第１の命令語を実行し、かつ、キャッシュミスを検知した場合であっても、前記命令スケジュール機構は、第２または第３の記憶手段から読み出した前記第２の命令語の実行サイクル数見積り情報に従うと前記第２の命令語のスケジューリング可能な位置が存在しない場合には、前記第１の命令語が属するプログラムを構成する命令語の消去を行わないことを特徴とするプロセッサ。

【請求項１１】

請求項８に記載されたプロセッサであって、

さらに、第４の記憶手段によって指定されたプログラムに属する命令語が一定時間に実行された数を測定する第３の測定手段を備え、

前記命令スケジュール機構は、測定された前記実行数がしきい値より小さい場合に、第１の記憶手段において、前記命令語が属しないプログラムに属する命令語を特定し、全て消去することを特徴とするプロセッサ。

【請求項１２】

請求項８に記載されたプロセッサであって、

さらに、第４の記憶手段によって指定されたプログラムに属する命令語が一定時間に実行された数を測定する第３の測定手段を備え、

前記測定手段により測定された前記実行数がしきい値より小さい場合に、当該事象をイベントにより通知することを特徴とするプロセッサ。

【請求項１３】

請求項１または２に記載されたプロセッサであって、さらに、

第１のプログラムと第２のプログラムのうちいずれを優先的に実行するかを指定する情報を記憶する第４の記憶手段と、

キューイングされている外部キャッシュリクエストを、前記外部キャッシュリクエストがどのプログラムに属するかを示す情報とともに記憶する外部キャッシュリクエストキューと、

第４の記憶手段により指定されたプログラムに属さない命令語により外部キャッシュミスを検知した場合に、前記外部キャッシュリクエストキューの中から、前記指定されたプログラムに属さない命令語に関連する外部キャッシュリクエストを特定し消去する外部キャッシュコントローラと

を備えることを特徴とするプロセッサ。

【請求項１４】

請求項１または２に記載されたプロセッサであって、

10

さらに、第１のプログラムと第２のプログラムのうちいずれを優先的に実行するかを指定する情報を記憶する第４の記憶手段と、

キューイングされている入出力リクエストを、前記入出力リクエストキューがどのプログラムに属するかを示す情報とともに記憶する入出力リクエストキューと、

第４の記憶手段により指定されたプログラムに属さない命令語により入出力待ちを検知した場合に、前記入出力リクエストキューの中から、前記指定されたプログラムに属さない命令語に関連する入出力リクエストを特定し消去する入出力コントローラと

を備えることを特徴とするプロセッサ。

【請求項１５】

請求項１３または１４に記載されたプロセッサであって、さらに、第４の記憶手段(優先フラグ)によって指定されたプログラムに属する命令語が一定時間に実行された数を測定する第３の測定手段を備え、前記第３の測定手段は前記実行数がしきい値より小さいことを検出したときに、当該事象をイベントにより通知することを特徴とするプロセッサ。

20

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、複数のプログラムを同時に実行するプロセッサに関する。

【背景技術】

【０００２】

現在までに、大規模サーバや高性能ワークステーションに使われる高性能プロセッサは、ほとんどホモジニアス・マルチスレッド型プロセッサへ移行している。このような利用環境では、システム全体のスループット向上が重視され、ＯＳはゆるやかなプロセススケジューリングによりプロセス間の公平性を保障するのが一般的である。一方、組み込み機器の多機能化および高性能化に伴い、組み込み機器用プロセッサについても、マルチスレッド型プロセッサへの移行が始まっている。ただし、低消費電力でなければならない点、および、特定のプロセス（音声、画像などの処理）については最低性能の保障が要求される点が、従来型のホモジニアス・マルチプロセッサシステムと異なっている。

30

【０００３】

複数のプログラムを同時に実行するプロセッサにおいて、低電力化に関しては、異種命令混在実行方式が採用できる。ここで、最低性能保障の障害となるのが、予測できないキャッシュミスによる性能低下である。キャッシュミスによる影響を抑えて最低性能を保障するための従来の技術について以下に説明する。

40

【０００４】

(１) １つの手法では、プロセスごとにキャッシュを分割する。しかし、キャッシュ分割により各スレッドが利用できるキャッシュ容量が半減すること自体、キャッシュミスを増加させるので、この手法は本末転倒である。

【０００５】

(２) Jichuan Chanらによる論文(Cooperative Cache Partitioning for Chip Multiprocessors (ICS2007)) に記載された手法では、共有キャッシュのまま、キャッシュの利用区分を変化させる。すなわち、マルチコアの共有キャッシュの割り付けを動的に変更し

50

てQoS保障を図る。キャッシュの利用区分を変化させることにより、ある範囲のワーキングセットの組合せに対してミス率を若干下げることが可能である。しかし、キャッシュミス自体を制御できないので、QoS保障が粗い。

【0006】

(3) A. Fedorovaらによる論文(Improving Performance Isolation on Chip Multiprocessors via an OS Schedule (PACT0007))に記載された手法では、OSのスケジューラによって性能保障をしないプロセスの優先度を意図的に下げてQoS保障を図る。OSが特定プロセスでないプロセスの優先度を下げること、なるべく多くのタイムスライスを特定プロセスに集中できるが、優先度が低いプロセスの実行が極端に低下する危険があり、ある一定以上に性能を保障することが困難である。

10

【0007】

(4) 特開2001-356903号公報に記載された手法では、キャッシュミス発生時に後続命令をパイプラインからすべてフラッシュすることにより、キャッシュミスによる悪影響を削減し、QoS保障を図る。しかし、この手法では、ストールの原因になっているかどうかにかかわらず、全ての命令をフラッシュするため、フラッシュが不要な場合にもフラッシュするので、フラッシュされたプロセスの性能が悪化しすぎてしまうという問題がある。

【0008】

したがって、複数のプログラムを同時に実行するプロセッサにおいて、キャッシュミスによる影響をより効果的に抑えることが望まれる。また、特定のプロセスについての最低性能保障を効果的に行えることが望まれる。

20

【特許文献1】特開2001-356903号公報

【非特許文献1】Jichuan Chan and Gurindar S. Sohi, Cooperative Cache Partitioning for Chip Multiprocessors, ICS2007, p242 (ACM)

【非特許文献2】A. Fedorova, M. Seltzer, and M.D. Smith, Improving Performance Isolation on Chip Multiprocessors via an OS Scheduler, PACT2007, p25 (IEEE Computer Society)

【非特許文献3】片岡晶人、中西正樹、山下茂、中島康人、VLIW型命令キューを持つOROC HIの命令スケジューリング機構、情報研報、2007-ARC-172, pp.25-30(2007)

【非特許文献4】中田尚、中島康彦、異種命令混在実行のためVLIW型命令キューの設計、IPSJ SIG Technical Report, 2007-ARC-175, pp.89-94(2007)

30

【発明の開示】

【発明が解決しようとする課題】

【0009】

本発明の目的は、複数のプログラムを同時に実行するプロセッサにおいて、キャッシュミスによる影響を抑えることである。

【課題を解決するための手段】

【0010】

本発明に係るプロセッサは、複数のプログラムを同時に実行するプロセッサである。このプロセッサは、前記複数のプログラムの中の第1のプログラムの実行位置を示す第1のプログラムカウンタと、前記複数のプログラムの中の第2のプログラムの実行位置を示す第2のプログラムカウンタと、第1のプログラムカウンタの内容により主記憶装置またはキャッシュから第1のプログラムの命令語を読み出す第1の命令読み出し機構と、第2のプログラムカウンタの内容により主記憶装置またはキャッシュから第2のプログラムの命令語を読み出す第2の命令読み出し機構と、第1及び第2の命令読み出し機構により読み出された命令語をキューイングするための第1の記憶手段と、第1の記憶手段から取り出した命令語のうちロード機能を有する命令を、キャッシュにアクセスして実行するキャッシュ制御機構と、第1の記憶手段から取り出した命令語により、ロード機能により取り出したデータを用いて演算を行う演算機構と、第1のプログラムに含まれる命令語の実行に要するサイクル数見積りのための実行サイクル数見積り情報を記憶する第2の記憶手段と

40

50

、第2のプログラムに含まれる命令語の実行に要するサイクル数見積りのための実行サイクル数見積り情報を記憶する第3の記憶手段と、第1のプログラムカウンタの内容により第2の記憶手段のエントリ位置を特定して実行サイクル数見積り情報を読み出す第1のキャッシュミス予測器と、第2のプログラムカウンタの内容により第3の記憶手段のエントリ位置を特定して実行サイクル数見積り情報を読み出す第2のキャッシュミス予測器と、第1および第2の命令読み出し機構から入力される命令語を第1の記憶手段にキューイングする命令スケジューリング機構であって、前記命令語が、前記キャッシュ制御機構によって処理される第1の命令語の実行結果を必要とし前記演算機構によって処理される第2の命令語である場合、前記第1の命令語に対応する、第1または第2のキャッシュミス予測器からの前記実行サイクル数見積り情報の内容に従って、第1の記憶手段における前記第2の命令語のスケジューリング位置を決定し、一方、前記実行サイクル数見積り情報に従うとスケジューリング可能な位置が存在しない場合には前記第2の命令語のスケジューリングを保留する命令スケジューリング機構とを備える。

10

20

30

40

50

【0011】

好ましくは、さらに、第1のプログラムと第2のプログラムのうちいずれを優先的に実行するかを指定する情報を記憶し、キャッシュ制御機構に送る第4の記憶手段を備える。第1の記憶手段は、キューイングされる命令語を、その命令語がどのプログラムに属するかを示す情報とともに記憶する。キャッシュ制御機構は、第4の記憶手段によって指定されたプログラムに属さない命令語を実行し、かつ、その実行においてキャッシュミスを検知した場合、キャッシュミス検知情報を命令スケジューリング機構に通知する。命令スケジューリング機構は、このキャッシュミス検知情報を受け取った場合に、第1の記憶手段において、実行された前記命令語が属するプログラムにおいて、前記命令語の実行結果を必要とする命令語を含む後続命令を特定し、全て消去する。

【発明の効果】

【0012】

キャッシュミス予測機構による命令スケジューリングにより、キャッシュ制御機構と演算機構の各々に対して命令語が発行される時刻の差を動的に変化させるので、演算効率を維持しつつキャッシュミスの影響を隠蔽できる。

また、キャッシュミスによるストール発生時に選択的に命令をフラッシュするので、ストールを防止できる。

【発明を実施するための最良の形態】

【0013】

以下、添付の図面を参照して発明の実施の形態を説明する。本発明の実施の形態では、複数のプログラムを同時に実行するプロセッサにおいて、キャッシュミスによる影響を抑えるため、また、キャッシュミスが起きても特定のプロセスについて最低性能の保障をするため、種々の対策を個別にまたは組み合わせて実行する。

【0014】

図1は、複数のプログラムを同時に実行するプロセッサの第1の実施形態の構成を示す。このプロセッサは、第1と第2の命令読み出し機構10, 12を含む。各々の命令読み出し機構10, 12は、命令フェッチ機構(IF)と命令デコード機構(ID)を含む。第1の命令読み出し機構10は、第1のプログラムの実行位置を示す第1のプログラムカウンタ14の内容により主記憶装置またはキャッシュから第1のプログラムの命令語を読み出し、第2の命令読み出し機構12は、第2のプログラムの実行位置を示す第2のプログラムカウンタ16の内容により主記憶装置またはキャッシュから第2のプログラムの命令語を読み出す。図1に示す例では、命令1次キャッシュ18、データ1次キャッシュ20、2次キャッシュ22および主記憶装置24が用いられる。命令スケジューリング機構26, 28は、それぞれ命令読み出し機構10, 12から受け取った命令語をスケジューリングして、命令キューを登録する記憶手段30にキューイングする。以下では、記憶手段30に記憶された命令キューも参照数字30で表す。命令キュー30は、命令スケジューリング機構26, 28がスケジューリングした命令語を登録するための記憶装置(記憶手段)で

ある。命令キュー 30 は、複数のキューからなり、複数の命令語が命令キュー 30 から並列に発行される。キャッシュ制御機構 32 は、命令キュー 30 から並列に発行される命令語のうち、ロード命令 / ストア命令を実行する実行部である。ロード機能を有する命令は、データ 1 次キャッシュ 20 にアクセスして実行される。また、複数の演算機構 34, 36 は、命令キュー 30 から取り出した命令語により、キャッシュ制御機構 32 がロード機能により取り出したデータを用いて演算を行う演算器 (ALU) である。演算機構 34, 36 は、異なる演算を行うが、同じ演算を行ってもよい。なお、キャッシュ制御機構 32 も、演算機構 34, 36 と同様に、命令キュー 30 からの命令を実行する実行部であるが、ここでは、キャッシュ 20 にアクセスする点で区別している。キャッシュ制御機構 32 は、キャッシュ 20 からキャッシュミスの有無の情報を受け取る。なお、図示していないが、各演算結果を他の演算器の入力などに直接に入力するためのバイパスが装備されている。

10

【0015】

第 1 の命令読み出し機構 10 と第 2 の命令読み出し機構 12 は、この実施形態では互いに異種の命令を処理するが、同種の命令を処理するものであってもよい。たとえば、第 1 の命令読み出し機構 10 は、OS や一般アプリケーションに適した汎用の命令 (たとえば ARM 命令) を処理し、第 2 の命令読み出し機構 12 は、メディアアプリケーションに適した命令 (たとえば FRV 命令) を処理する。メディアアプリケーションでは高い命令レベルの並列性が期待されるが、OS や一般アプリケーションではそのような並列性は期待されない。そこで、複数のプログラムのうち、優先的に実行されるべきプログラムを指定する優先フラグ 38 を記憶する記憶手段を設ける。以下では、記憶手段に記憶された優先フラグも参照数字 38 で表す。たとえば、メディアアプリケーションについて最低性能を保障したい場合、優先フラグ 38 に、メディアアプリケーションのためのプログラムを優先することを指定しておく。命令スケジューリング機構 26, 28 は、優先フラグ 38 に基づいて、指定されたプログラム (優先プログラム) の命令のキューイングを優先的に行う。各々の演算結果を他の演算器の入力に直接入力するためのバイパスを装備することにより、1 サイクルにより演算が完了できる命令の間にレジスタ依存関係が存在する場合においても、毎サイクル演算を実行できる。命令キュー 30 からの命令を受け取り実行するキャッシュ制御機構 32 と演算機構 34, 36 が実行する命令の間には依存関係があるため、スケジューリング機構 26, 28 が命令キュー 30 内でスケジューリングする命令の位置に制約がある。

20

30

【0016】

ここでは、異種命令を効率的に低消費電力で処理するため、命令キュー 30 は VLIW 型命令キューとして構成し、全体の動作周波数の向上や消費電力の削減を図る (片岡晶人ほか、VLIW 型命令キューを持つ OROCHI の命令スケジューリング機構、情報研報、2007-ARC-172, pp.25-30(2007)、および、中田尚ほか、異種命令混在実行のため VLIW 型命令キューの設計、IPSJ SIG Technical Report, 2007-ARC-175, pp.89-94(2007) 参照)。VLIW 型命令キューは、演算器ごとに入り口が分かれていて、各エントリは、デコーダ側から演算器側に向かうシフトレジスタである。命令語のキューイングは任意の位置に可能である。第 1 の命令読み出し機構 10 は、汎用の命令を単純な内部命令に分解して、スケジューリングを容易にする。命令キュー 30 では、異種の命令が混在してスケジューリングされる。命令スケジューリング機構 26, 28 は、リネームステージにおいてマップテーブルで命令の間の依存関係を検査し、命令キュー 30 の適切な位置にスケジューリングする。依存関係がなければ、既に命令が入っている位置より前方にキューイングできる。同一タイミングでスケジューリングされる命令の間に依存関係がある場合は、命令順において後方になるようにスケジューリングする。命令は一旦入った場所からは移動できない。また、命令の間の追い越しができない。命令発行は常に命令キュー 30 の演算器側の端のみから行う。命令が発行されると、命令語は演算器側に 1 つシフトされる。命令キュー 30 に登録する際に命令スケジューリング機構 26, 28 により命令スケジューリングを行うことにより、命令発行機構および演算器周辺回路が単純化される。

40

50

【 0 0 1 7 】

命令スケジューリング機構 26, 28 は、命令読み出し機構 10, 12 からの情報にしたがって、命令を投入すべき演算器 32 ~ 36 につながっているキューを選択する。同じキューを使用する命令が 2 個以上ある時は、命令順で早いものを優先する。この結果衝突した命令とそれ以降の命令のスケジューリングは次以降に行われる。次に、マップテーブル（図示しない）からの情報にしたがって、全てのソースオペランドが供給される位置を見つける。なお、ソースオペランドが供給される位置を見つけるためには、VLIW 型命令キューの各エントリのデスティネーションレジスタ番号とソースレジスタ番号の一致比較を行い、できるだけ演算器に近いエントリを探すことになる。ただし、バイパスが使用できない演算器、すなわち、ロード命令からのオペランド供給は 1 サイクル遅れることを考慮する必要がある。命令発行の際には、図示しないディスパッチ/リードステージにおいて、VLIW 型命令キューの演算器側の端のエントリが発行可能かどうかを検査し、発行可能であればレジスタ値を読み出し、演算器の入力バッファに投入する。この VLIW キュー 30 では、ロード命令がキャッシュミスを起こした場合、後続命令がキャッシュ制御機構 32 を使用するか、ロード結果を使用するまで、命令発行を継続できる。

10

【 0 0 1 8 】

第 1 の実施形態のプロセッサでは、キャッシュミスによる影響を抑えるため、簡素な構成のキャッシュミス予測機構を命令スケジューリング機構 26, 28 に付加し、キャッシュミス予測を考慮して命令スケジューリングを行う。具体的には、2 つのキャッシュミス予測機構は、2 つのプログラムに対応して、第 2 の記憶手段 40 と第 1 のキャッシュミス予測器 44 からなり、また、第 3 の記憶手段 42 と第 2 のキャッシュミス予測器 46 からなる。第 2 の記憶手段 40 と第 3 の記憶手段 42 は、それぞれ、第 1 のプログラムおよび第 2 のプログラムに含まれる命令語の実行に要するサイクル数見積りのための実行サイクル数見積り情報をテーブルに記憶し、第 1 のキャッシュミス予測器 44 および第 2 のキャッシュミス予測器 46 は、それぞれ、第 1 のプログラムカウンタ 14 および第 2 のプログラムカウンタ 16 の内容により第 1 の記憶手段 40 および第 2 の記憶手段 42 のエントリ位置を特定して実行サイクル数見積り情報を読み出す。命令スケジューリング機構 26, 28 は、スケジューリングの際に、この実行サイクル数見積り情報を用いてキューイングを行い、キャッシュ制御機構 32 と演算機構 34, 36 に対して命令語が発行される時刻の差を動的に変化させる。第 1 の記憶手段 40 および第 2 の記憶手段 42 に記憶される各命令語の実行サイクル数見積り情報は、あらかじめ適当な値に設定しておく。

20

30

【 0 0 1 9 】

具体的には、命令読み出し機構 10, 12 がフェッチした命令語が、キャッシュ制御機構 32 によって処理される第 1 の命令語の実行結果を必要とし演算機構 34, 36 によって処理される第 2 の命令語である場合、命令スケジューリング機構 26, 28 は、記憶手段 40, 42 から読み出した、第 1 の命令語に対応する実行サイクル数見積り情報の内容に従って、命令キュー 30 における第 2 の命令語の位置をあらかじめ離すようにスケジューリングする。また、実行サイクル数見積り情報に従うと命令キュー 30 にスケジューリング可能な位置が存在しない場合には、命令スケジューリング機構 26, 28 は第 2 の命令語のスケジューリングを保留する。すなわち、命令の間隔を十分に離すだけの余地がなければスケジューリングしない。これにより、第 1、第 2 の記憶手段 40, 42 に記憶されている実行サイクル数見積り情報に応じて命令の間隔をキャッシュミスの影響を受けないように離すことができるので、キャッシュミスによる影響を隠蔽できる。

40

【 0 0 2 0 】

第 2 の実施形態のプロセッサでは、図 1 に示すプロセッサの構成において、第 2 の記憶手段 40 または第 3 の記憶手段 42 は、キャッシュミスの検出情報および非検出情報に応じてキャッシュミス予測情報を適当な値に更新する。具体的には、キャッシュ制御機構 32 は、命令キュー 30 から取り出した命令語についてキャッシュミスを検出した場合、前記命令語の実行サイクル数見積り情報を記憶している第 2 または第 3 の記憶手段 40, 42 における命令語のエントリ位置に、キャッシュミス検出情報を送り、キャッシュミスを

50

検出しない場合、そのエントリ位置に、キャッシュミス非検出情報を送る。第2または第3の記憶手段40, 42は、キャッシュミス検出情報を受け取ると、その命令語の実行サイクル数見積り情報を増加または最大値に飽和させ、逆に、キャッシュミス非検出情報を受け取ると、その命令語の実行サイクル数見積り情報を減少または最小値に飽和させる回路を備える。これにより、その命令語の実行サイクル数見積り情報は、キャッシュミス発生状況(ミス率)に応じてキャッシュミスが生じない最小値に動的に変更される。したがって、命令スケジューリング機構26, 28は、後続命令をスケジューリングする位置を、記憶手段40, 42における実行サイクル数見積り情報に基づいて、キャッシュミスが生じないと予測される適当な位置に動的に変更できる。これにより、記憶手段40, 42に記憶される実行サイクル数見積り情報の値を、動的に最適な値に調整できる。

10

【0021】

以上のようにキャッシュミスを予測して命令語のスケジューリングを行うことにより命令語発行の際のキャッシュミスを防止するが、予測ミスによりキャッシュミスが起こることがある。特に、同時に実行する複数のプログラムの中に最低性能の保障(QoS保障)を行うプログラム(たとえばマルチメディアアプリケーション)がある場合、最低性能を保障するため、そのプログラムの実行の際にキャッシュミスが頻発しないようにしなければならない。そこで、図2に示す第3の実施形態のプロセッサでは、命令キュー30において、キューイングされる命令語50は、その命令語がどのプログラムに属するかを示す種別フラグ52とともに記憶しておく。さらに、優先プログラムに属しない命令語が一定時間内に実行された数を測定する第1の測定手段48を備える。第1の測定手段48により測定された実行数が増加または減少するにつれ、非優先プログラムに対応する記憶手段38、40は、その内部での命令語の実行サイクル数見積り情報を増加または減少する。測定された実行数に対応する実行サイクル数見積り情報の増減の関係はあらかじめ設定されていて、実行サイクル数見積り情報は、これを用いて増減される。

20

【0022】

上述のキャッシュミス予測にもかかわらず予測ミスが頻発する状況下では最低性能の維持が困難となる。そこで、図3に示す第4の実施形態のプロセッサでは、最低性能保障を効率的に行うため、優先プログラムに属さない命令語の実行においてキャッシュミスが起こると、優先でないプログラム(非優先プログラム)に属する、その命令語以降の命令流を命令キュー30の中で選択的にフラッシュ(消去)して、キャッシュミスを起こしたロード命令に依存している命令が原因で発行停止(ストール)が発生しないようにする。この選択的フラッシュにより、優先プログラムへの干渉を排除して、優先プログラムの命令流の発行停止要因を取り除く。

30

【0023】

具体的には、図3に示すように、命令キュー30において、キューイングされる命令語50が、その命令語がどのプログラムに属するかを示す種別フラグ52とともに記憶される。そして、非優先プログラムに属する命令語の実行においてキャッシュミスが起こると、優先フラグ38と種別フラグ52とが不一致の命令(すなわち非優先プログラムの命令語)のみを選択的にフラッシュする。すなわち、キャッシュ制御機構32は、優先フラグ38により指定された優先プログラムに属さない命令語を実行する際にキャッシュミスを検知した場合、キャッシュミス検知情報を命令スケジューリング機構26, 28に通知する。命令スケジューリング機構26, 28には、選択的フラッシュ機構が設けられている。選択的フラッシュ機構は、キャッシュミス検知情報を受け取ると、命令キュー30において、キャッシュミスを起こした命令語が属するプログラムにおいて、その命令語の実行結果を必要とする命令語を含む後続命令を特定し、すべて消去する。好ましくは、命令スケジューリング機構26, 28は、キャッシュミスが解消されたとの情報をキャッシュ制御機構30から通知されると、命令キュー30に対して、消去した後続命令のキューイングを再開する。

40

【0024】

第5の実施形態では、図3に示すプロセッサにおいて、たとえば、キャッシュ制御機構

50

30が優先プログラムに属さない命令語である第1の命令語を実行し、かつ、キャッシュミスを検知した場合であっても、第1の命令語の実行結果を必要とする第2の命令語のスケジューリング可能な位置が命令キュー30に存在しない場合は、優先でないプログラムにおける後続命令のキューイングを保留する。すなわち、第1の命令語を処理した命令スケジュール機構26, 28は、キャッシュ制御機構30からキャッシュミスの検知を通知されたとき、記憶手段40, 42から読み出した第2の命令語の実行サイクル数見積み情報に従うと命令キュー30に第2の命令語のスケジューリング可能な位置が存在しないと判断した場合、第1の命令語が属するプログラムを構成する命令語のうち、第1の命令語の実行結果を必要とする命令語を含む後続命令のキューイングを保留する。こうして優先でないプログラムの実行を一時的に停止させることにより、優先プログラムの実行速度を一定に維持できる。

10

【0025】

第6の実施形態のプロセッサでは、図4に示すように、第1の実施形態で用いられるキャッシュミス予測機構に加えて、キャッシュ制御機構30からキャッシュミスの有無の通知を受け取る第2の測定手段60を備え、キャッシュミスの測定結果に応じて命令語のスケジューリングを行う。具体的には、第2の測定手段60は、たとえば一定の期間において、優先フラグ38によって指定された優先プログラムが走行中である場合における、優先フラグ38によって指定されずかつキャッシュミスの発生頻度がしきい値より低い非優先プログラムの走行時間とキャッシュミスの発生頻度がしきい値より高い非優先プログラムの走行時間の第1の比と、優先フラグ38によって指定された優先プログラムが走行中でない場合における、優先フラグ38によって指定されずかつキャッシュミスの発生頻度がしきい値より低い非優先プログラムの走行時間とキャッシュミスの発生頻度がしきい値より高い非優先プログラムの走行時間の第2の比を、測定する。第2の測定手段60は、ソフトウェアとして実装してもよい。この測定結果に対応して、命令スケジュール機構26, 28は、命令キュー30における命令語のスケジューリングにおいて、第1の比が第2の比よりも大きくなるようにする。これにより、優先フラグ38により指定されない非優先プログラムにおけるキャッシュミスの発生頻度に依存せずに、優先フラグ38により指定される優先プログラムの走行中における優先プログラムの実行速度を一定に維持する。好ましくは、第4の実施形態におけるように、非優先プログラムに属する命令語の実行においてキャッシュミスを起こると、命令スケジュール機構26, 28は、非優先プログラムにおいて、キャッシュミスを起こした命令語の実行結果を必要とする命令語を含む後続命令を特定し、すべて消去する。

20

30

【0026】

第7の実施形態のプロセッサでは、図5に示すように、第1の実施形態で用いられるキャッシュミス予測機構に加えて、優先フラグ38によって指定された優先プログラムに属する命令語が一定時間に実行された数を測定する第3の測定手段62を備える。そして、測定された実行数がしきい値より低い場合、当該事象をイベントにより通知する。この通知に対応する制御のための制御レジスタまたは命令セットをOSが用意していれば、OSのスケジューラ制御を併用できる。通知を受けたOSは、必要に応じて、スケジューラでのQoS維持に切り換える。好ましくは、さらに、第4の実施形態におけるように、非優先プログラムに属する命令語の実行においてキャッシュミスを起こると、非優先プログラムにおいて、キャッシュミスを起こした命令語の実行結果を必要とする命令語を含む後続命令を特定し、すべて消去する。

40

【0027】

第8の実施形態のプロセッサでは、図6に示すように、第1の実施形態で用いられるキャッシュミス予測機構に加えて、第3の測定手段62に加えて、さらに、命令読み出し機構10, 12は、優先フラグ38によって指定されない非優先プログラムに属する命令語の実行を一時的に停止させる停止手段64を備える。第3の測定手段62は、測定された実行数がしきい値より低いことを検出した場合には、停止手段64に、その旨を通知する。停止手段64は、この通知に対応して、非優先プログラムの命令スケジュールリングを一

50

時的に停止させる。これにより、優先プログラムの実行速度が一定に維持される。

【 0 0 2 8 】

第 9 の実施形態のプロセッサでは、第 4 の実施形態で説明した選択的フラッシュに加えて、さらに、図 7 に示すように、優先フラグ 3 8 によって指定された優先プログラムに属する命令語が一定時間に実行された数を測定する第 3 の測定手段 6 2 を備え、測定結果に対応して優先プログラムの命令語の選択的フラッシュをすることにより、非優先プログラムにおけるキャッシュミスの発生頻度にかかわらず、優先プログラムの実行速度を一定に保つ。具体的には、第 3 の測定手段 6 2 から、測定された実行数がしきい値より低いことが通知されると、命令スケジューリング機構 2 6 , 2 8 は、命令キュー 3 0 において、優先フラグ 3 8 と種別フラグ 5 2 とが不一致の命令語（すなわち非優先プログラムの命令語）のみを選択的にフラッシュする。

10

【 0 0 2 9 】

第 1 0 の実施形態のプロセッサでは、優先フラグ 3 8 によって指定されない複数の非優先プログラムが実行される。図 8 に示すように、第 1 の実施形態で用いられるキャッシュミス予測機構に加えて、非優先プログラムに属する命令語が一定時間に実行された数を測定する第 4 の測定手段 6 6 を備えるとともに、命令読み出し機構 1 0 , 1 2 は、非優先プログラムごとに、命令語のスケジューリングを一時的に停止させる停止手段 6 8 を備える。第 4 の測定手段 6 6 は、測定された実行数がしきい値より低いことを検出した場合には、停止手段 6 8 に通知し、停止手段 6 8 は、個々の非優先プログラムのスケジューリングを一時的に停止する。この一時的停止により、優先プログラムまたは複数の非優先プログラムの中の特定の非優先プログラムの実行速度を一定に維持する。

20

【 0 0 3 0 】

第 1 1 の実施形態のプロセッサでは、第 4 の実施形態で説明した選択的フラッシュに加えて、さらに、図 9 に示すように、優先フラグ 3 8 によって指定された優先プログラムに属する命令語が一定時間に実行された数を測定する第 3 の測定手段 6 2 を備える。第 3 の測定手段 6 2 は、測定された実行数がしきい値より低い場合、すなわち、優先プログラムの実行速度を一定に維持することが困難であることを検出した場合、当該事象をイベントにより通知する。これらの機構を制御するための制御レジスタまたは命令セットを OS が用意していれば、OS のスケジューラ制御を併用できる。すなわち、通知を受けた OS は必要に応じてスケジューラでの QoS 維持に切り換える。

30

【 0 0 3 1 】

次に、第 1 2 の実施形態のプロセッサについて説明する。このプロセッサは、第 1 の実施形態で用いられるキャッシュミス予測機構に加えて、図 1 0 に示すように、外部キャッシュリクエストキューを登録する記憶手段 7 4 を備える。以下では、記憶手段 7 0 に記憶された外部キャッシュキューも参照数字 7 4 で表す。外部リクエストキュー 7 4 では、キューイングされている外部キャッシュリクエスト 7 0 が、外部キャッシュリクエストがどのプログラムに属するかを示す情報である種別フラグ 7 2 とともに、キューイングされる。外部キャッシュは、この例では 2 次キャッシュ 2 2 である。外部キャッシュコントローラ 7 6 は、外部リクエストキュー 7 4 から外部キャッシュリクエストを実行する。しかし、優先フラグ 3 8 により指定された優先プログラムに属さないロード命令により外部キャッシュミスを検出した場合には、外部キャッシュリクエストキュー 7 4 の中から該当する外部キャッシュリクエストを特定し、後続の 2 次リクエストをフラッシュ（消去）する。非保障側のリクエストを選択的にフラッシュすることにより、非優先側の命令の 2 次キャッシュミスによる優先プログラムの実行への干渉を排除して、優先プログラムの QoS を維持する。

40

【 0 0 3 2 】

次に、第 1 3 の実施形態のプロセッサについて説明する。このプロセッサは、第 1 の実施形態で用いられるキャッシュミス予測機構に加えて、図 1 1 に示すように、入出力リクエストキューを登録する記憶手段 8 4 を備える。以下では、記憶手段 8 4 に記憶された入出力リクエストキューも参照数字 8 4 で表す。入出力リクエストキュー 8 4 には、外部の

50

入出力装置との入出力リクエスト 80 が、その入出力リクエストがどのプログラムに属するかを示す情報である種別フラグ 82 とともにキューイングされる。入出力コントローラ 86 は、入出力リクエストキュー 84 に対応して入出力リクエスト 80 を実行するが、優先フラグ 38 により指定された優先プログラムに属さない命令語により入出力待ちを検知した場合には、入出力リクエストキュー 84 の中で該当する入出力リクエストを特定し消去する。すなわち、性能低下の原因が外部入出力装置へのリクエストであった場合、非保障側のリクエストを選択的にキャンセルする。これにより、優先プログラムの QoS を維持する。

【0033】

好ましくは、第 12 と第 13 の実施形態において、優先フラグ 38 によって指定された優先プログラムに属する命令語が一定時間に実行された数を測定する第 3 の測定手段 62 を備える。第 3 の測定手段 62 は、測定された実行数がしきい値より低いことを検出したときに、当該事象をイベントにより通知する。これにより、これらの機構を制御するための制御レジスタまたは命令セットを OS が用意していれば、OS のスケジューラ制御を併用できる。すなわち、通知を受けた OS は、必要に応じてスケジューラでの QoS 維持に切り換える。

【0034】

以上に説明したように、本発明の上述の種々の実施形態において、(1) キャッシュミス予測機構による命令スケジューリング、(2) キャッシュミスによるストール発生時の選択的命 20 令フラッシュ、(3) 非優先プログラムの実行を一時的に停止させることによる優先プログラムの一定の実行速度の維持などの、互いに独立した処理が種々の形態で組み合わせられている。このような発明の種々の特徴は、いうまでもなく、個別に、また、種々の組合せとして、具体化できる。

【0035】

また、発明の実施形態では、命令キューとして、VLW キューを用いたが、これに限定されない。また、複数のプログラムとして、異種命令の場合について説明しているが、同種命令でもよい。同種命令でも優先側を判断する機構があれば、本発明の優先的な処理が適用できる。また、組み込み機器用プロセッサでないプロセッサでも、個々のコアを異種命令混在実行とする場合は、本発明が適用できる。

【図面の簡単な説明】

【0036】

【図 1】第 1 の実施形態のプロセッサのブロック図

【図 2】第 3 の実施形態のプロセッサのブロック図

【図 3】第 4 の実施形態のプロセッサのブロック図

【図 4】第 6 の実施形態のプロセッサのブロック図

【図 5】第 7 の実施形態のプロセッサのブロック図

【図 6】第 8 の実施形態のプロセッサのブロック図

【図 7】第 9 の実施形態のプロセッサのブロック図

【図 8】第 10 の実施形態のプロセッサのブロック図

【図 9】第 11 の実施形態のプロセッサのブロック図

【図 10】第 12 の実施形態のプロセッサの 1 部のブロック図

【図 11】第 13 の実施形態のプロセッサの 1 部のブロック図

【符号の説明】

【0037】

10、12 命令読み出し機構、14、16 プログラムカウンタ、18 命令 1 次キャッシュ、20 データ 1 次キャッシュ、22 2 次キャッシュ、24 主記憶装置、26、28 命令スケジューラ機構、30 命令キュー（第 1 の記憶手段）、32 キャッシュ制御機構、34、36 演算機構、38 優先フラグ、40 第 2 の記憶手段、42 第 3 の記憶手段、44、46 キャッシュミス予測器、50 命令語、52 種別フラグ、60 第 2 の

10

20

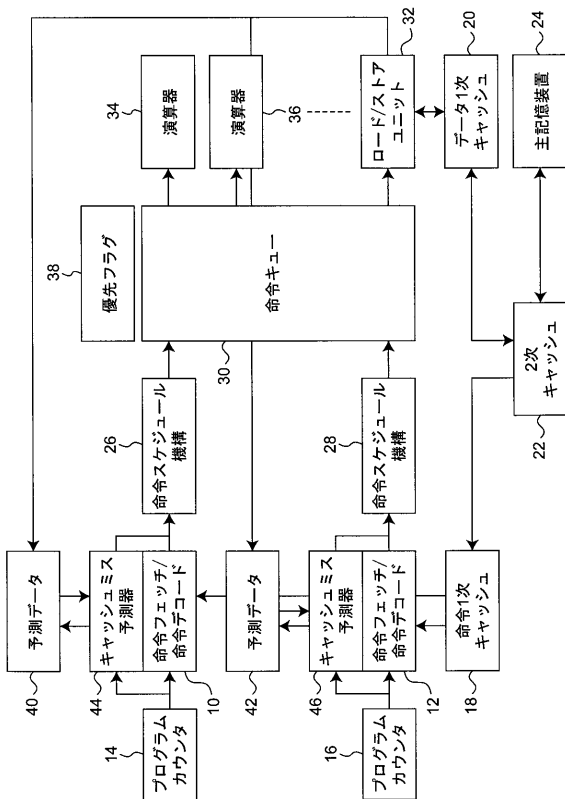
30

40

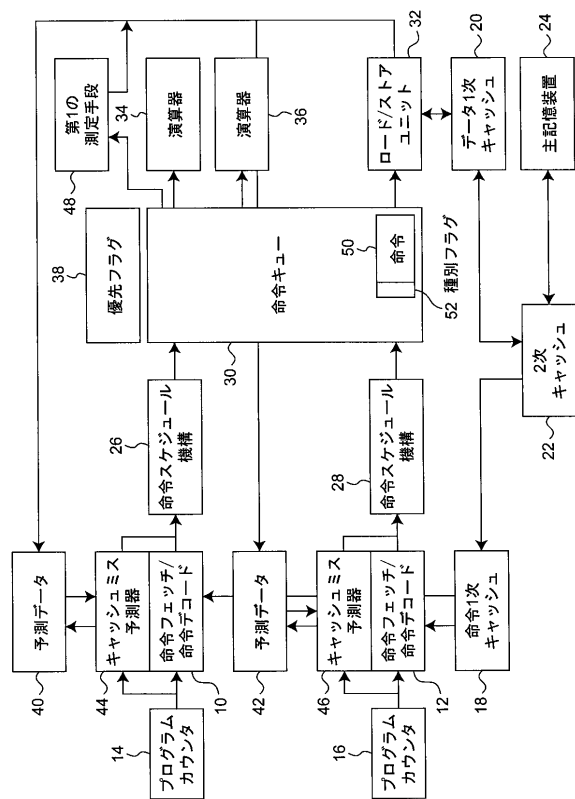
50

測定手段、 6 2 第 3 の測定手段、 6 4 停止手段、 6 6 第 4 の測定手段、
 6 8 停止手段、 7 0 外部キャッシュリクエスト、 7 2 種別フラグ、
 7 4 外部キャッシュリクエストキュー、 7 6 外部キャッシュコントローラ、
 8 0 入出力リクエスト、 8 2 種別フラグ、 8 4 入出力リクエストキュー、
 8 6 入出力コントローラ。

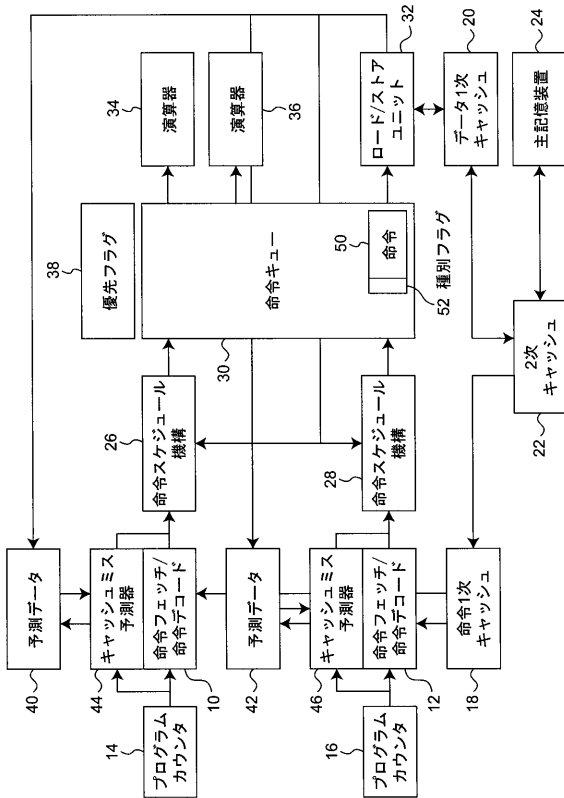
【図 1】



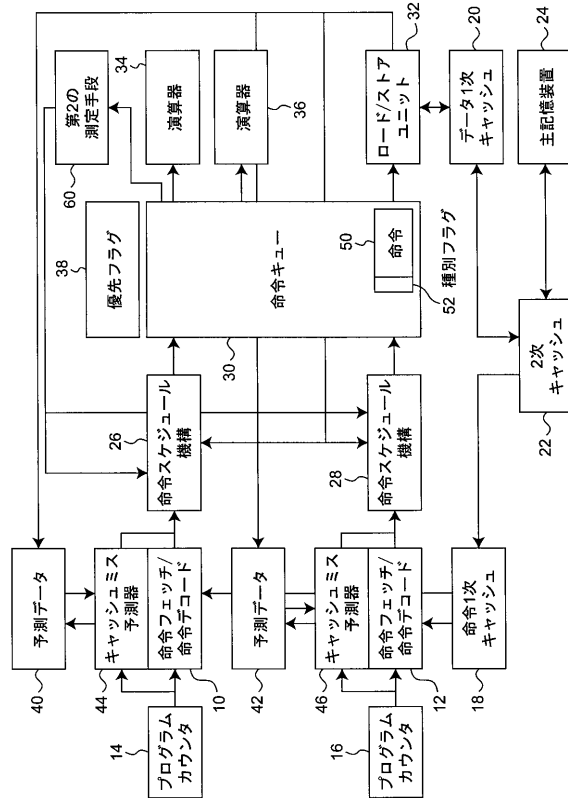
【図 2】



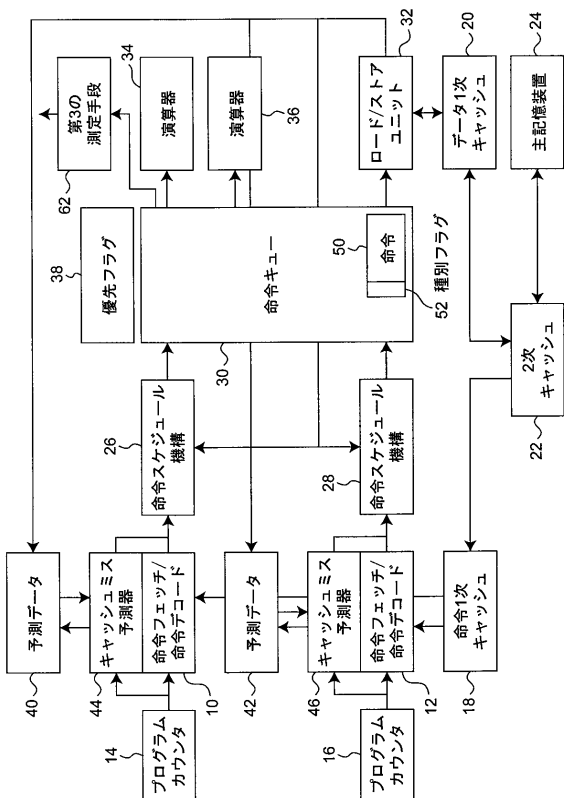
【 図 3 】



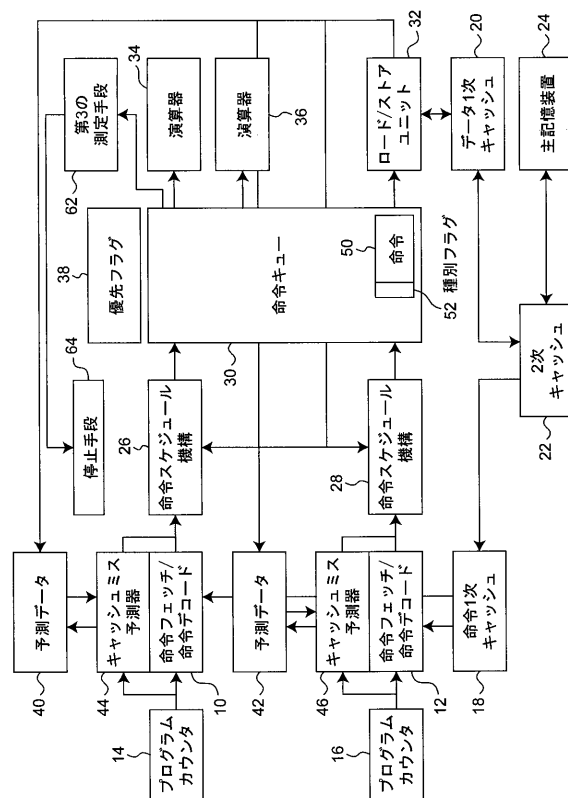
【 図 4 】



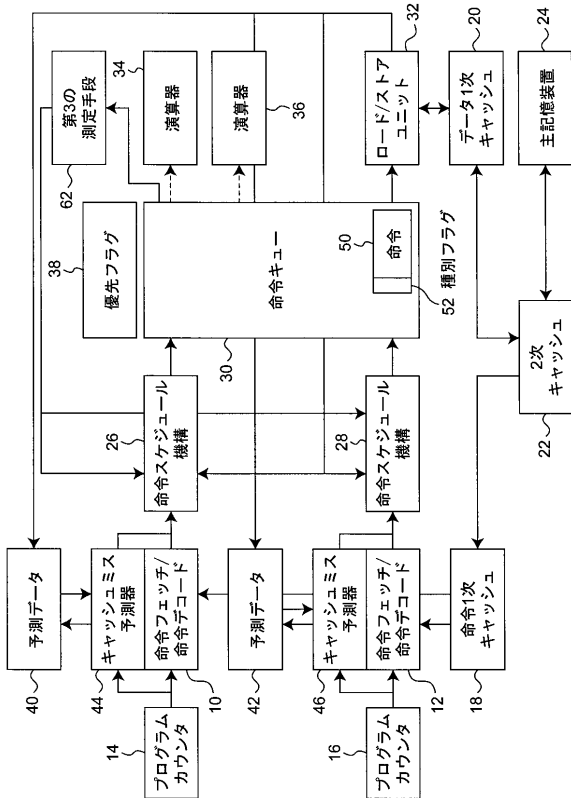
【 図 5 】



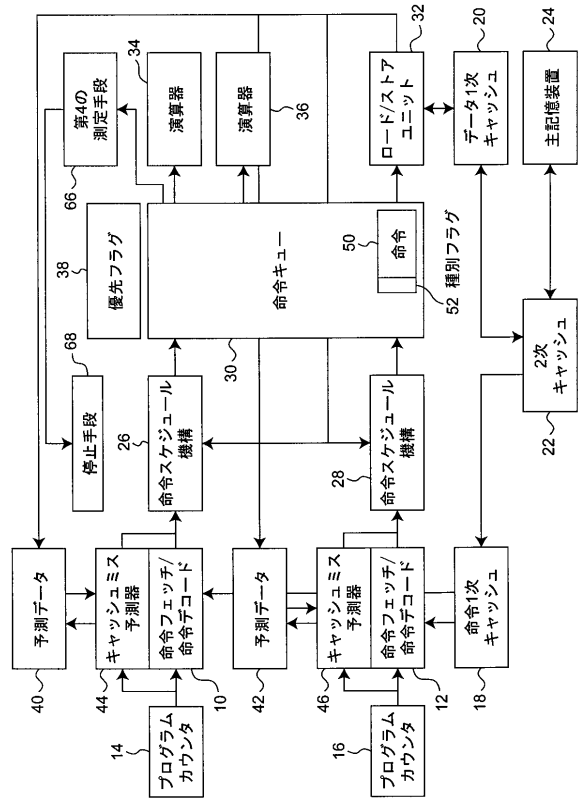
【 図 6 】



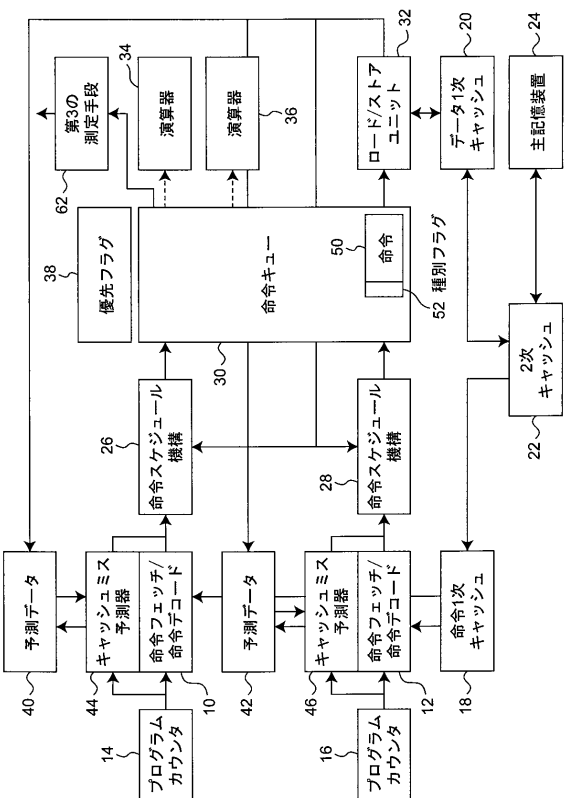
【 図 7 】



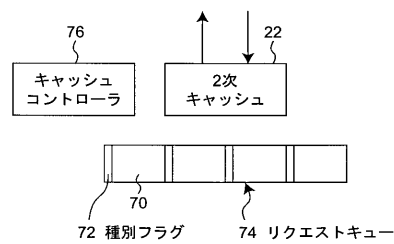
【 図 8 】



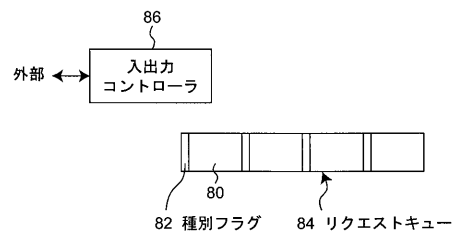
【 図 9 】



【 図 1 0 】



【 ㊦ 1 1 】



フロントページの続き

(74)代理人 100125874
弁理士 川端 純市

(72)発明者 中田 尚
奈良県生駒市高山町 8 9 1 6 - 5 大学宿舎 D - 1 0 1

(72)発明者 中島 康彦
京都府京都市左京区吉田上阿達町 3 7 - 3 - 2 0 3

(72)発明者 北村 俊明
広島県広島市安佐南区古市一丁目 5 - 1 5 - 7 0 1

(72)発明者 須賀 敦浩
埼玉県所沢市下富 1 1 9 0 - 9

(72)発明者 宮本 幸昌
神奈川県横浜市磯子区森 6 - 2 8 - 1 7

F ターム(参考) 5B005 KK22 LL01 MM02 NN22 SS12 VV03 VV04
5B013 AA05 AA11