(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6286065号
(P6286065)

(45)発行日　平成30年2月28日(2018.2.28)　(24)登録日　平成30年2月9日(2018.2.9)

(51)Int.Cl.			F I		
G06F	***9/38***	***(2006.01)***	G06F	9/38	310F
G06F	***9/34***	***(2006.01)***	G06F	9/38	310E
			G06F	9/34	350A

請求項の数 21　(全 24 頁)

(21)出願番号	特願2016-564427 (P2016-564427)
(86)(22)出願日	平成26年12月14日 (2014.12.14)
(65)公表番号	特表2017-503294 (P2017-503294A)
(43)公表日	平成29年1月26日 (2017.1.26)
(86)国際出願番号	PCT/IB2014/003170
(87)国際公開番号	WO2016/097791
(87)国際公開日	平成28年6月23日 (2016.6.23)
審査請求日	平成28年1月12日 (2016.1.12)

(73)特許権者　515324257
ヴィア　アライアンス　セミコンダクター　カンパニー　リミテッド
中華人民共和国　２０１２０３　シャンハイ，ザンジアン　ハイテク　パーク，ジンク・ロード　２５３７，ルーム　３０１
(74)代理人　100107766
弁理士　伊東　忠重
(74)代理人　100070150
弁理士　伊東　忠彦
(74)代理人　100091214
弁理士　大貫　進介

最終頁に続く

(54)【発明の名称】アウトオブオーダープロセッサの書き込み結合メモリ領域アクセスに依存するロードリプレイを除外する装置及び方法

(57)【特許請求の範囲】

【請求項１】

アウトオブオーダープロセッサでのリプレイを低減する装置であって、

第１のロードマイクロ命令をディスパッチするように構成された第１のリザベーションステーションであり、前記第１のロードマイクロ命令が複数の非コアリソースの１つに向けられた複数の指定されたロード命令の１つであるかを検出して保持バス上で示すように構成された、第１のリザベーションステーションと、

前記保持バスに接続され、前記第１のロードマイクロ命令に依存する１つ又は複数の新しいマイクロ命令を、前記第１のロードマイクロ命令がディスパッチされてから第１のクロックサイクル数の後に、実行のためにディスパッチするように構成された第２のリザベーションステーションであり、前記保持バス上で前記第１のロードマイクロ命令が複数の指定されたロード命令の前記１つであることが示されている場合に、前記１つ又は複数の新しいマイクロ命令のディスパッチを、前記第１のロードマイクロ命令がオペランドを取得するまでストールさせるように構成された、第２のリザベーションステーションと、を備え、

前記複数の非コアリソースは、

前記アウトオブオーダープロセッサに対する前記複数の指定されたロード命令が、ジョイントテストアクショングループインターフェイスを通じてプログラムされたランダムアクセスメモリ、を含み、

初期化時に、前記アウトオブオーダープロセッサは、前記複数の指定されたロード命令

を読み込むために前記ランダムアクセスメモリにアクセスする、
装置。
【請求項２】
前記アウトオブオーダープロセッサがマルチコアプロセッサを含み、前記マルチコアプロセッサ内の各コアが、前記第１のリザベーションステーションと前記第２のリザベーションステーションとを含む、
請求項１に記載の装置。
【請求項３】
前記複数の非コアリソースの１つが、前記ランダムアクセスメモリを含み、
前記ランダムアクセスメモリが、前記各コアと同じダイに配置され、かつ、前記各コアの外部に配置された、
請求項２に記載の装置。
【請求項４】
前記複数の非コアリソースの１つが、前記マルチコアプロセッサと同じダイに配置されておらず、
前記複数の非コアリソースの１つが、各コアと同じダイに配置され、かつ、前記各コアの外部に配置されたバスユニットを通じてアクセスされる、
請求項２に記載の装置。
【請求項５】
前記第１のリザベーションステーションに接続され、前記第１のロードマイクロ命令を受け取って実行するように構成されたロード実行ロジックであり、実行するマイクロ命令を受け取らない場合に節電状態に入るように構成された、ロード実行ロジックを、
さらに含む、請求項１に記載の装置。
【請求項６】
前記第１のロードマイクロ命令が前記指定されたロードマイクロ命令でない場合、前記ロード実行ロジックが、ミスバス上で、前記第１のロードマイクロ命令が前記第１のクロックサイクル数で正常な実行に失敗するか否かを示し、
前記１つ又は複数の新しいマイクロ命令のリプレイを開始する、
請求項５に記載の装置。
【請求項７】
前記第１のロードマイクロ命令が前記指定されたロードマイクロ命令である場合、前記ロード実行ロジックが、前記第１のロードマイクロ命令が正常な実行のために前記第１のクロックサイクル数よりも多くを必要とする場合に正常な実行に失敗することを示さず、
前記１つ又は複数の新しいマイクロ命令のリプレイを除外する、
請求項６に記載の装置。
【請求項８】
リプレイを低減する装置であって、
前記装置は、複数のコアを含むマルチコアプロセッサを含み、
前記複数のコアそれぞれが、
第１のロードマイクロ命令をディスパッチするように構成された第１のリザベーションステーションであり、前記第１のロードマイクロ命令が複数の非コアリソースの１つに向けられた複数の指定されたロード命令の１つであるかを検出して保持バス上で示すように構成された、第１のリザベーションステーションと、
前記保持バスに接続され、前記第１のロードマイクロ命令に依存する１つ又は複数の新しいマイクロ命令を、前記第１のロードマイクロ命令がディスパッチされてから第１のクロックサイクル数の後に、実行のためにディスパッチするように構成された第２のリザベーションステーションであり、前記保持バス上で前記第１のロードマイクロ命令が複数の指定されたロード命令の前記１つであることが示されている場合に、前記１つ又は複数の新しいマイクロ命令のディスパッチを、前記第１のロードマイクロ命令がオペランドを取得するまでストールさせるように構成された、第２のリザベーションステーションと、を

備え、
前記複数の非コアリソースは、
前記マルチコアプロセッサに対する前記複数の指定されたロード命令が、ジョイントテストアクショングループインターフェイスを通じてプログラムされたランダムアクセスメモリ、を含み、
初期化時に、前記マルチコアプロセッサは、前記複数の指定されたロード命令を読み込むために前記ランダムアクセスメモリにアクセスする、
装置。
【請求項９】
前記マルチコアプロセッサが、ｘ８６互換のマルチコアプロセッサを含む、
請求項８に記載の装置。
【請求項１０】
前記複数の非コアリソースの１つが、前記ランダムアクセスメモリを含み、
前記ランダムアクセスメモリが、前記マルチコアプロセッサと同じダイに配置され、かつ、前記複数のコアそれぞれの外部に配置された、
請求項８に記載の装置。
【請求項１１】
前記複数の非コアリソースの１つが、前記マルチコアプロセッサと同じダイに配置されておらず、
前記複数の非コアリソースの１つが、前記複数のコアのそれぞれと同じダイに配置され、かつ、前記複数のコアそれぞれの外部に配置されたバスユニットを通じてアクセスされる、
請求項８に記載の装置。
【請求項１２】
前記複数のコアそれぞれが、
前記第１のリザベーションステーションに接続され、前記第１のロードマイクロ命令を受け取って実行するように構成されたロード実行ロジックであり、実行するマイクロ命令を受け取らない場合に節電状態に入るように構成された、ロード実行ロジックを、
さらに含む請求項８に記載の装置。
【請求項１３】
前記第１のロードマイクロ命令が前記指定されたロードマイクロ命令でない場合、前記ロード実行ロジックが、ミスバス上で、前記第１のロードマイクロ命令が正常な実行のために前記第１のクロックサイクル数よりも多くを必要とするときに正常な実行に失敗することを示し、
前記１つ又は複数の新しいマイクロ命令のリプレイを開始する、
請求項１２に記載の装置。
【請求項１４】
前記第１のロードマイクロ命令が前記指定されたロードマイクロ命令である場合、前記ロード実行ロジックが、前記第１のロードマイクロ命令が正常な実行のために前記第１のクロックサイクル数よりも多くを必要とするときに正常な実行に失敗することを示さず、
前記１つ又は複数の新しいマイクロ命令のリプレイを除外する、
請求項１３に記載の装置。
【請求項１５】
アウトオブオーダープロセッサでのリプレイを低減する方法であって、
前記アウトオブオーダープロセッサに対する複数の指定されたロード命令が、ジョイントテストアクショングループインターフェイスを通じてプログラムされたランダムアクセスメモリを含む複数の非コアリソースを配置するステップと、
初期化時に、前記複数の指定されたロード命令を読み込むために前記ランダムアクセスメモリにアクセスするステップと、
第１のリザベーションステーションを通じて、第１のロードマイクロ命令をディスパッ

チし、前記第１のロードマイクロ命令が前記複数の非コアリソースの１つに向けられた前記複数の指定されたロード命令の１つであるか否かを検出して保持バス上で示すステップと、

前記保持バスに接続された第２のリザベーションステーションを通じて、前記第１のロードマイクロ命令に依存する１つ又は複数の新しいマイクロ命令を、前記第１のロードマイクロ命令がディスパッチされてから第１のクロックサイクル数の後に、実行のためにディスパッチし、前記保持バス上で前記第１のロードマイクロ命令が前記複数の指定されたロード命令の１つであることが示されている場合は、前記１つ又は複数の新しいマイクロ命令のディスパッチを、前記第１のロードマイクロ命令がオペランドを取得するまでストールさせるステップと、

を含む方法。

【請求項１６】

前記アウトオブオーダープロセッサがマルチコアプロセッサを含み、

前記マルチコアプロセッサ内の各コアが、前記第１のリザベーションステーションと前記第２のリザベーションステーションとを含む、

請求項１５に記載の方法。

【請求項１７】

前記複数の非コアリソースの１つが、前記ランダムアクセスメモリを含み、

前記ランダムアクセスメモリが、前記各コアと同じダイに配置され、かつ、前記各コアの外部に配置された、

請求項１６に記載の方法。

【請求項１８】

前記複数の非コアリソースの１つが、前記マルチコアプロセッサと同じダイに配置されておらず、

複数の非コアリソースの前記１つが、各コアと同じダイに配置され、かつ、前記各コアの外部に配置されたバスユニットを通じてアクセスされる、

請求項１６に記載の方法。

【請求項１９】

前記第１のリザベーションステーションに接続されたロード実行ロジックを通じて、前記第１のロードマイクロ命令を受け取って実行し、実行するマイクロ命令を受け取らない場合は節電状態に入るステップ、をさらに含む、

請求項１５に記載の方法。

【請求項２０】

前記第１のロードマイクロ命令が前記指定されたロードマイクロ命令でない場合、ミスバス上で、前記第１のロードマイクロ命令が正常な実行のために前記第１のクロックサイクル数よりも多くを必要とするときに正常な実行に失敗することを示し、

前記１つ又は複数の新しいマイクロ命令のリプレイを開始する、

請求項１９に記載の方法。

【請求項２１】

前記第１のロードマイクロ命令が前記指定されたロードマイクロ命令である場合、前記第１のロードマイクロ命令が正常な実行のために前記第１のクロックサイクル数よりも多くを必要とするときに正常な実行に失敗することを示さず、

前記１つ又は複数の新しいマイクロ命令のリプレイを除外する、

請求項２０に記載の方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、一般的にはマイクロエレクトロニクスの分野に関し、より詳細には、アウトオブオーダープロセッサのロードリプレイを軽減する節電メカニズムに関する。

【背景技術】

【０００２】
集積デバイス技術は、過去４０年間で飛躍的に進歩した。マイクロプロセッサ分野について詳細には、４ビット単一命令の１０マイクロメートルのデバイスから始まったものが、半導体製造技術の進歩により、アーキテクチャ及び密度の点でますます複雑なデバイスを設計することが可能となった。８０年代及び９０年代には、単一のダイに搭載された数百万個のトランジスタを含む、いわゆるパイプラインマイクロプロセッサやスーパースカラマイクロプロセッサが開発された。それから２０年を経た現在、単一のダイに搭載された数十億のトランジスタを備え、データ処理用の複数のマイクロプロセッサコアを含む、３２ナノメートルのデバイスが製造されている。

【０００３】
今日のマルチコアプロセッサにおける命令並列性の採用に加えて、アウトオブオーダー実行（ｏｕｔ－ｏｆ－ｏｒｄｅｒ　ｅｘｅｃｕｔｉｏｎ）メカニズムもまた普及している。アウトオブオーダー実行の原理によると、命令は、実行ユニットによる実行のためにリザベーションステーションにキューイングされ、古い命令の実行結果であるオペランドを待機している命令のみがリザベーションステーションで保持される。オペランドを待機していない命令は、実行のためにディスパッチされる。実行に続いて、結果がキューイングされ、典型的にはリタイア状態と呼ばれるプロセッサステージで、レジスタに正しい順序で戻される。よって、命令は元のプログラム順と異なる順序で実行される。

【０００４】
アウトオブオーダー実行はスループットを大幅に向上させる。これは、本来であればアイドル状態となる実行ユニットを利用して、古い命令がオペランドを待機している間に新しい命令を実行するからである。ただし、当業者が理解するように、命令は必ずしも正常に実行されるとは限らない。結果として、ある命令が正常に実行されなかった場合、その命令と、その命令よりも新しいすべての命令とを、再び実行しなければならない。この概念は「リプレイ（”ｒｅｐｌａｙ”）」と呼ばれている。これは、今日のプロセッサのメカニズムが、現在の実行を基本的に停止し、命令が異常実行された直前のポイントまでマシン状態を後退させ、異常実行された命令と、その命令よりも新しいすべての命令とをリプレイするからである。新しい命令は、異常実行された命令のディスパッチよりも前にディスパッチされている場合と、そうでない場合とがある。

【０００５】
ただし、リプレイは例外的なケースであり、リプレイがパフォーマンスに与える影響は、多くの場合は無視できる。しかし、オペランドが利用できるまで命令をリザベーションステーションで保持することのパフォーマンス面での影響は甚大であり、マイクロプロセッサの設計者は、特定の命令を、そのオペランドが実行の直前に利用可能となる可能性が高い場合にディスパッチできるようにするアクセラレーション手法を開発した。これらの特定の命令がディスパッチされるだけでなく、これらの特定の命令に必要なオペランドを適時に提供するメカニズムも設けられる。

【０００６】
本願は、そのようなアクセラレーション手法の１つに対処する。このアクセラレーション手法では、オンコアの（ｏｎ－ｃｏｒｅ）キャッシュメモリに存在する可能性が高いと予想されるオペランドを必要とする新しい命令が、実行されることでそのオペランドのキャッシュからの取得を可能にするロード命令がディスパッチされてから、指定されたクロックサイクル数の後にディスパッチされる。よって、ロード命令がディスパッチされたとき、そのオペランドを待機している新しい命令は、指定されたクロックサイクル数が発生するまで、それぞれのリザベーションステーションでストール（ｓｔａｌｌ）される。その後、必要なオペランドが利用可能になるという高い確実性を持って、それらの新しい命令が実行のためにディスパッチされる。

【０００７】
上述したアクセラレーション手法を利用することによるパフォーマンスの向上は非常に大きいため、マイクロプロセッサのアーキテクトは、通常、ほとんどのロード命令（Ｉ／

Ｏからのロード、キャッシュ不可のロード、割り込みレジスタからのロード、特殊ロードなど）にこの手法を適用する。ただし、オペランドを取得するために指定されたサイクル数よりも多くを要し、よってオペランドが利用可能になるという見込みでディスパッチされた新しい命令のすべてをリプレイすることが必要となるロード命令が存在することも確かである。このロードアクセラレーション手法によるパフォーマンスの向上は、稀に生じるリプレイによるパフォーマンスの低下を補って余りある。

【０００８】

しかし、マルチコアプロセッサ技術が進化し続ける中、設計者は、レベル２（Ｌ２）キャッシュ、割り込みコントローラ、ヒューズアレイなどの頻繁にアクセスされない特定のプロセッサリソースを、各コア内で複製するのではなく、マルチコアプロセッサダイの共通領域に配置するほうがよいと考えている。つまり、これら上述のリソースは、プロセッサコアによって共有される。当業者が理解するように、オペランドをオフコアの（ｏｆｆ－ｃｏｒｅ）リソース（たとえば、ヒューズアレイ）からロードするには、オンコアのリソース（たとえば、Ｌ１キャッシュ）からのロードに必要な時間よりもかなり長い時間がかかる。そして上述したアクセラレーション手法の下でディスパッチされた新しい命令をリプレイしなければならないことにより生じるパフォーマンス低下は甚大ではないが、電力利用への影響は大きく、かなりの数の命令が、リプレイを余儀なくされることが事実上確実な条件の下で実行されていることが、本発明者により確認された。また、これらの命令の最初の実行で電力が必然的に浪費されるため、電池寿命、熱プロファイル、及び信頼性の観点で不利である。

【０００９】

そのため、必要なリプレイの数を減らすことにより、プロセッサで電力を節約することを可能にする装置及び方法が求められている。

【００１０】

加えて、アウトオブオーダープロセッサの節電を実現する、同プロセッサのロードリプレイ低減メカニズムが求められている。

【発明の概要】

【発明が解決しようとする課題】

【００１１】

本発明は、数ある出願の中でも特に、上述した問題を解決すると共に、先行技術の他の問題、欠点、及び制限に対処することを目的としている。

【課題を解決するための手段】

【００１２】

本発明の一側面は、アウトオブオーダープロセッサでのリプレイを低減する装置に関する。この装置は、第１のリザベーションステーションと、第２のリザベーションステーションとを備える。第１のリザベーションステーションは、第１のロードマイクロ命令をディスパッチするように構成され、かつ第１のロードマイクロ命令がオンコアのキャッシュメモリ以外の規定のリソースからオペランドを取得するようになされた指定されたロードマイクロ命令であるかを検出して保持バス上で示すように構成される。第２のリザベーションステーションは、保持バスに接続され、第１のロードマイクロ命令に依存する１つ又は複数の新しいマイクロ命令を、第１のロードマイクロ命令がディスパッチされてから第１のクロックサイクル数の後に、実行のためにディスパッチするように構成され、保持バス上で第１のロードマイクロ命令が指定されたロードマイクロ命令であることが示されている場合に、１つ又は複数の新しいマイクロ命令のディスパッチを、第１のロードマイクロ命令がオペランドを取得するまでストールさせるように構成される。複数の非コア（ｎｏｎ－ｃｏｒｅ）リソースは、ジョイントテストアクショングループインターフェイスを通じてアウトオブオーダープロセッサに対応する複数の指定されたロード命令を用いてプログラムされたランダムアクセスメモリを含み、アウトオブオーダープロセッサは、初期化時に、複数の指定されたロード命令を判断するためにランダムアクセスメモリにアクセスする。

【００１３】
本発明の別の側面は、リプレイを低減する装置に関する。装置は、複数のコアを含むマルチコアプロセッサを含む。複数のコアのそれぞれは、第１のリザベーションステーションと、第２のリザベーションステーションとを備える。第１のリザベーションステーションは、第１のロードマイクロ命令をディスパッチするように構成され、かつ第１のロードマイクロ命令がオンコアのキャッシュメモリ以外の規定のリソースからオペランドを取得するようになされた指定されたロードマイクロ命令であるかを検出して保持バス上で示すように構成される。第２のリザベーションステーションは、保持バスに接続され、第１のロードマイクロ命令に依存する１つ又は複数の新しいマイクロ命令を、第１のロードマイクロ命令がディスパッチされてから第１のクロックサイクル数の後に、実行のためにディスパッチするように構成され、かつ保持バス上で第１のロードマイクロ命令が指定されたロードマイクロ命令であることが示されている場合に、１つ又は複数の新しいマイクロ命令のディスパッチを、第１のロードマイクロ命令がオペランドを取得するまでストールさせるように構成される。複数の非コアリソースは、ジョイントテストアクショングループインターフェイスを通じてアウトオブオーダープロセッサに対応する複数の指定されたロード命令を用いてプログラムされたランダムアクセスメモリを含み、マルチコアプロセッサは、初期化時に、複数の指定されたロード命令を判断するためにランダムアクセスメモリにアクセスする。
【００１４】
本発明のさらなる側面は、アウトオブオーダープロセッサでのリプレイを低減する方法に関する。方法は、ジョイントテストアクショングループインターフェイスを通じてアウトオブオーダープロセッサに対応する複数の指定されたロード命令を用いてプログラムされたランダムアクセスメモリを含む複数の非コアリソースを配置するステップと、初期化時に、複数の指定されたロード命令を判断するためにランダムアクセスメモリにアクセスするステップと、第１のリザベーションステーションを通じて、第１のロードマイクロ命令をディスパッチし、第１のロードマイクロ命令がオンコアのキャッシュメモリ以外の規定のリソースからオペランドを取得するようになされｘ８６特殊バスサイクルの実行により得られるロード命令を含む指定されたロードマイクロ命令であるか否かを検出して保持バス上で示すステップと、保持バスに接続された第２のリザベーションステーションを通じて、第１のロードマイクロ命令に依存する１つ又は複数の新しいマイクロ命令を、第１のロードマイクロ命令がディスパッチされてから第１のクロックサイクル数後に実行のためにディスパッチし、保持バス上で第１のロードマイクロ命令が指定されたロードマイクロ命令であることが示されている場合は、１つ又は複数の新しいマイクロ命令のディスパッチを、第１のロードマイクロ命令がオペランドを取得するまでストールさせるステップとを含む。
【００１５】
産業上の利用可能性に関し、本発明は、汎用又は専用のコンピューティングデバイスで使用され得るマイクロプロセッサ内で実装される。
【００１６】
本発明の上記及び他の目的、特徴、及び利点は、以下の説明及び添付の図面を参照することで、よりよく理解される。
【図面の簡単な説明】
【００１７】
【図１】各コアの外部に配置された共通リソースを利用する今日のマルチコアプロセッサを示すブロック図である。
【図２】図１に示す今日の各コアの例示的なコアステージを示すブロック図である。
【図３】非コアリソースからのロードのための節電メカニズムを備えた本発明のマルチコアプロセッサを示すブロック図である。
【図４】図３に示す各コアの例示的なコアステージを示すブロック図である。
【図５】図４に示すアンコア（ｕｎｃｏｒｅ）ストール要素の詳細を示すブロック図であ

る。
【図６】図４に示す各リザベーションステーションの詳細を示すブロック図である。
【図７】図４に示すアンコアミス要素の詳細を示すブロック図である。
【発明を実施するための形態】
【００１８】
本発明の例示的かつ例証的な実施形態について以下に説明する。明瞭性を考慮して、本明細書では実際の実装のすべての特徴については説明しない。当業者は、そのような実際の実施形態を発展させる過程で、具体的な目的を達成するために、実装に固有のさまざまな意思決定がなされることを理解する。そうした意思決定には、システム関連及びビジネス関連の制約への準拠など、実装ごとに異なるものが含まれる。さらに、そのような開発努力は複雑で時間がかかるものとなる可能性があるが、それでも本開示の利益を受ける当業者にとってはありふれた作業であろうことが理解される。好ましい実施形態に対する多様な変更は当業者にとって明白であり、本明細書で定義された一般原理は他の実施形態にも適用され得る。よって、本発明は本明細書で図示及び説明された特定の実施形態に限定されることを意図したものではなく、本明細書で開示される原理及び新規な特徴に一致する最も広い範囲が与えられる。
【００１９】
本発明について、添付の図面を参照しながら以下に説明する。さまざまな構造、システム、及びデバイスが、図面に概略的に示されている。これらは説明のみを目的としており、また当業者にとって既知である詳細事項によって本発明が不明瞭にならないようにしている。しかしながら添付の図面は、本発明の例証的な例について記述及び説明するために用意されている。本明細書で使用される用語及びフレーズは、それらの用語及びフレーズの関連分野における当業者による理解と一致する意味を持つものとして理解及び解釈されるべきである。用語又はフレーズの特殊な定義（即ち、当業者により理解される通常かつ慣例的な意味と異なる定義）を、その用語又はフレーズの本明細書での一貫した使用により暗示することは意図されていない。用語又はフレーズが特殊な意味（即ち、当業者による理解と異なる意味）を持つことが意図されている場合、そのような特殊な定義は、それらの用語又はフレーズの特殊な定義を直接かつ明白に提供する定義の態様で、本明細書において明示的に説明される。
【００２０】
定義
集積回路（ＩＣ）：典型的にはシリコンである半導体材料の小片の上に組み立てられた、一群の電子回路。よってＩＣは、チップ、マイクロチップ、又はダイとも呼ばれる。
【００２１】
中央処理装置（ＣＰＵ）：算術演算、論理演算、及び入力／出力演算を含むデータ操作を実行することで、コンピュータプログラム（「コンピュータアプリケーション」又は「アプリケーション」とも呼ばれる）の命令を実行する電子回路（即ち、「ハードウェア」）。
【００２２】
マイクロプロセッサ：単一の集積回路でＣＰＵとして機能する電子デバイス。マイクロプロセッサは、デジタルデータを入力として受け取り、メモリ（オンダイ又はオフダイ）からフェッチした命令に応じてデータを処理し、命令により規定された操作の結果を出力として生成する。汎用マイクロプロセッサは、デスクトップコンピュータ、モバイルコンピュータ、又はタブレットコンピュータで利用されることがあり、計算、テキスト編集、マルチメディア表示、インターネット閲覧などの用途に利用される。またマイクロプロセッサは、組み込みシステムに配置されて、アプライアンス、携帯電話、スマートフォン、産業用制御デバイスなどの幅広いデバイスを制御し得る。
【００２３】
マルチコアプロセッサ：マルチコアマイクロプロセッサとも呼ばれるマルチコアプロセッサは、単一の集積回路上に組み立てられた複数のＣＰＵ（「コア」）を備えるマイクロ

プロセッサである。
【００２４】
命令セットアーキテクチャ（ＩＳＡ）又は命令セット：データ型、命令、レジスタ、アドレッシングモード、メモリアーキテクチャ、割り込み及び例外処理、並びに入力／出力を含む、コンピュータアーキテクチャのプログラミングに関連する部分。ＩＳＡは、一群のオペコード（即ち、機械語命令）の仕様と、特定のＣＰＵにより実装されるネイティブ命令とを含む。
【００２５】
ｘ８６互換マイクロプロセッサ：ｘ８６　ＩＳＡに基づいてプログラムされたコンピュータアプリケーションを実行できるマイクロプロセッサ。
【００２６】
マイクロコード：複数のマイクロ命令を示すために利用される用語。マイクロ命令（「ネイティブ命令」とも呼ばれる）は、マイクロプロセッサのサブユニットが実行するレベルでの命令である。例示的なサブユニットには、整数ユニット、浮動小数点ユニット、マルチメディア（ＭＭＸ）ユニット、及びロード／ストアユニットがある。たとえば、マイクロ命令は、縮小命令セットコンピュータ（ＲＩＳＣ）マイクロプロセッサにより直接実行される。ｘ８６互換マイクロプロセッサなどの複合命令セットコンピュータ（ＣＩＳＣ）マイクロプロセッサの場合、ｘ８６命令が関連するマイクロ命令に変換され、関連するマイクロ命令がＣＩＳＣマイクロプロセッサ内のサブユニットにより直接実行される。
【００２７】
ヒューズ：典型的にはフィラメントとして配置される導電性構造であり、フィラメントの全体に電圧を加えることにより、及び／又はフィラメントを通じて電流を流すことにより、選択位置で破損させることができる。ヒューズは、すべての潜在的なプログラム可能領域にフィラメントを作成するために、既知の組み立て手法を使用して、ダイ微細構成（ｄｉｅ　ｔｏｐｏｇｒａｐｈｙ）全域の指定された領域に配置され得る。ヒューズ構造は、ダイに配置された対応するデバイスの所望のプログラミング性を実現するために、組み立て後に飛ばされる（又は飛ばされない）。
【００２８】
マルチコアプロセッサのロードメカニズム及び非コアリソースからのロード操作を実行するために今日のマルチコアプロセッサ内で利用されている関連手法の上述した背景説明に鑑みて、それらの今日の手法の制限及び欠点を、図１及び図２を参照しながら説明する。それに続いて、本発明について図３～図７を参照しながら説明する。
【００２９】
図１を参照すると、各コア１０１の外部に配置された共通リソースを利用する今日のマルチコアプロセッサを示すブロック図１００が提示されている。ブロック図１００は、４つのプロセッサコア１０１が配置されたデバイスダイ１１０を示している。本発明者は、明瞭性を目的として、本明細書ではクワッドコア（即ち、４つのコア１０１）のマルチコアマイクロプロセッサについて説明することを指摘する。しかし、本発明の原理及び特徴は、異なる数のコア１０１を備えたマルチコアマイクロプロセッサにも適用することができる。
【００３０】
当業者が理解するように、設計及び／又はビジネスに関連する理由により、アーキテクトは特定のプロセッサリソースをコア１０１の間で共有することを選択する場合がある。パフォーマンス上の動機により、これらの共有リソースは、典型的にはコア１０１と同じダイ１１０に配置され、各コア１０１から高速バス１１１～１１４を通じてアクセスされる。したがってブロック図１００は、Ｌ２キャッシュ１０３、ランダムアクセスメモリ（ＲＡＭ）１０４、バスユニット１０５、入力／出力ユニット１０６、高度プログラム可能割り込みコントローラ（ＡＰＩＣ）１０７、ヒューズアレイ１０８などの例示的な共有リソースを示している。コア１０１と同じダイ１１０に配置され、かつコア１０１の外部に位置する、これらの共有リソース１０３～１０８の集まりを、以下ではアンコア（ｕｎｃ

ｏｒｅ）リソース１０２と呼ぶ。よって、バスＢ１　１１１は、コア１　１０１がアンコア１０２にアクセスすることを可能にする。バスＢ２　１１２は、コア２　１０１がアンコア１０２にアクセスすることを可能にする。バスＢ３　１１３は、コア３　１０１がアンコア１０２にアクセスすることを可能にする。バスＢ４　１１４は、コア４　１０１がアンコア１０２にアクセスすることを可能にする。典型的な構成では、マルチコアプロセッサは、デバイスダイ１１０の外部にある、システムメモリ（メモリバスＭＥＭを通じてアクセス）、入力／出力要素（バスＩ／Ｏを通じてアクセス）、システム制御要素（バスＣＴＲＬを通じてアクセス）などの他の要素に接続される。

【００３１】

動作中、オペレーティングシステムの制御下にある各コア１０１は、システムメモリからフェッチされた関連する命令を実行することができ、対象の用途に対応するオペランドを操作する。１つ又は複数のコア１０１は、１つ又は複数のアンコアリソース１０２にアクセスすることが必要な場合があり、対応するバスＢ１～Ｂ４を通じて、制御された態様で、それら１つ又は複数のアンコアリソース１０２にアクセスする。たとえば、電源初期化時に、１つ又は複数のコア１０１は、ヒューズアレイ１０８からのロード操作を実行して構成パラメータを取得することができ、又はＲＡＭ１０４からのロードを実行してパッチ情報を取得することができる。通常の動作時に、コア１０１はＬ２キャッシュ１０３にアクセスして、オンコアキャッシュ（たとえば、Ｌ１キャッシュ）に存在しないメモリオペランドを読み取る／書き込むことができる。コア１０１は、バスユニット１０５にアクセスしてシステムメモリとの間で読み取り／書き込みを実行することができ、又はＩ／Ｏユニット１０６を利用してＩ／Ｏバスを通じてＩ／Ｏ操作を実行することができる。コア１０１は、さらにＡＰＩＣ１０７にアクセスして割り込み操作を実行することができる。

【００３２】

図２を参照すると、図１に示す今日の各コア１０１の例示的なコアステージを示すブロック図が示されている。この図は、ダイ１１０に配置されたプロセッサコア２０１を示している。コア２０１はフェッチステージ２１１を含み、このフェッチステージがトランスレータステージ２１２にバス２４１を通じて接続されている。トランスレータステージ２１２は、リネームステージ２１３にバス２４２を通じて接続されている。リネームステージ２１３は、リプレイマルチプレクサステージ２１４にバス２４３を通じて接続されている。リプレイマルチプレクサ２１４は、複数のリザベーションステーションＲＳ１～ＲＳＮ　２２１．１～２２１．Ｎと、ロードリザベーションステーションＲＳＬ２２１．Ｌとに、リザベーションバス２４４を通じて接続されている。各リザベーションステーションＲＳ１～ＲＳＮ、ＲＳＬは、対応する実行ユニットＥＵ１～ＥＵＮ　２２２．１～２２２．Ｎ、ＥＵＬ２２２．Ｌに、対応するディスパッチバス２５１．１～２５１．Ｎ、２５１．Ｌを通じて接続されている。リザベーションステーションＲＳ１～ＲＳＮ、ＲＳＬは、レジスタファイル２２６にレジスタバス２４５を通じて接続されている。

【００３３】

本発明者は、実行ユニットＥＵＬの外部で、他の実行ユニットＥＵ１～ＥＵＮが、整数ユニット、浮動小数点ユニット、マルチメディアユニット、ストアユニットなど、今日のスーパースカラプロセッサに典型的なユニットを含み得ることを指摘する。本願で特に興味深いのは、実行ユニットＥＵＬである。このユニットは、ロードユニット２２２．Ｌとして図示されており、その主な機能は、図１を参照しながら上述したシステムメモリ、システムＩ／Ｏ、アンコアリソース２３０などの多様なリソースからオペランドをロードすることである。

【００３４】

よって、ロードユニットＥＵＬは、Ｌ１キャッシュ２２３にバス２５４を通じて接続され、アンコア２３０にバス２５６を通じて接続されている。ほとんどのメモリオペランドについて、ロードユニット２２２．ＬはまずＬ１キャッシュ２２３にアクセスする。ロードがＬ１キャッシュ２２３でミスした場合、ロードユニット２２２．Ｌはアンコア２３０のＬ２キャッシュにアクセスしなければならない。また実行ユニットＥＵ１～ＥＵＮ、Ｅ

ＵＬは、リオーダバッファ２２４にバス２５２を通じて接続されている。さらに、ロードユニットＥＵＬは、リオーダバッファ２２４にバスＭＩＳＳ　２５３を通じて接続されている。リオーダバッファ２２４は、リプレイマルチプレクサ２１４にバスＲＥＰＬＡＹ２５８を通じて接続され、リタイアユニット２２５にバス２５７を通じて接続されている。リタイアユニット２２５は、レジスタファイル２２６に書き戻しＷＢバス２５５を通じて接続されている。

【００３５】

図２に示されたコアステージは、今日のスーパースカラ、即ち「アウトオブオーダー」プロセッサコア２０１の単なる例示であり、本発明を明確に教示する目的で提示されていることが指摘される。当業者が理解するように、プロセッサコアステージは、アーキテクチャ及び対象の用途によって異なり得る。

【００３６】

動作中、プログラム命令（図示せず）がフェッチユニット２１１によってメモリからフェッチされる。ｘ８６互換プロセッサコア２０１の場合、これらのプログラム命令は、ｘ８６　ＩＳＡに適合する。プログラム命令は、トランスレータ２１２にバス２４１上で順番に提供される。トランスレータ２１２は、プログラム命令により指定された操作を実行するために、対応する実行ユニットＥＵ１～ＥＵＮ、ＥＵＬにサブ操作を指示する１つ又は複数のマイクロ命令にプログラム命令を変換する。マイクロ命令は、リネームユニット２１３にバス２４２上で提供される。リネームユニット２１３において、一部のマイクロ命令で指定されているアーキテクチャレジスタ（即ち、オペランドのレジスタ位置）が、プロセッサコア２０１のハードウェアレジスタ（図示せず）に再マップされる。これは、独立したマイクロ命令ストリームの実行並列性を高めるためである。またリネームユニット２１３は、連続的なプログラム順に応じて、各マイクロ命令にタグ付けする。マイクロ命令のソースオペランドフィールド及びデスティネーションオペランドフィールドにも、１つ又は複数のオペランドが依存する新しいマイクロ命令のタグがタグ付けされる。リネーム済みマイクロ命令は、リプレイマルチプレクサ２１４にバス２４３上で提供される。

【００３７】

リプレイマルチプレクサ２１４は、複数の機能をアウトオブオーダープロセッサコア２０１で実行する。主として、マルチプレクサ２１４は、リネーム済み各マイクロ命令のオペコードを読み取って、実行のための適切な実行ユニットＥＵ１～ＥＵＮ、ＥＵＬを判断する。たとえば、リネームされた整数マイクロ命令はＥＵ１により実行され、浮動小数点マイクロ命令はＥＵ２により実行される、という具合である。そして本出願で特に興味深いのは、リネームされたロードマイクロ命令が、ロード実行ユニットＥＵＬにより実行され得ることである。よって、リプレイマルチプレクサ２１４は、１つ又は複数のリネーム済みマイクロ命令を、１つ又は複数のリザベーションステーションＲＳ１～ＲＳＮ、ＲＳＬに提供して、対応する実行ユニットＥＵ１～ＥＵＮ、ＥＵＬへのディスパッチを待機させる。

【００３８】

各リザベーションステーションＲＳ１～ＲＳＮ、ＲＳＬは、レジスタファイル２２６にアクセスして、キューイングされているリネーム済みマイクロ命令の操作に必要なオペランドを読み取る。古いリネーム済みマイクロ命令のタグがタグ付けされていないリネーム済みマイクロ命令（即ち、古いリネーム済みマイクロ命令に依存していないリネーム済みマイクロ命令）は、対応する実行ユニットＥＵ１～ＥＵＮ、ＥＵＬにすぐにディスパッチされて実行される。従属するリネーム済みマイクロ命令（即ち、まだ実行が完了していない古いリネーム済みマイクロ命令のタグを含む、リネーム済みマイクロ命令）は、通常はタグ付けされた従属オペランドが利用可能になるまで、リザベーションステーションＲＳ１～ＲＳＮ、ＲＳＬにより保持される。タグ付けされた従属オペランドが利用可能になると、それらのオペランドが従属するリネーム済みマイクロ命令に提供され、それらのマイクロ命令が対応する実行ユニットＥＵ１～ＥＵＮ、ＥＵＬにディスパッチされて実行される。実行ユニットＥＵ１～ＥＵＮ、ＥＵＬは、マイクロ命令を実行していないときに、節

電機能を実行することもできる。通常、実行ユニットＥＵ１～ＥＵＮ、ＥＵＬの内部のクロックは、マイクロ命令を実行していないときはシャットダウンされ、それによって電力が大幅に節約される。

【００３９】

リネーム済みマイクロ命令及びその結果は、リオーダバッファ２２４にバス２５２を通じて提供される。リオーダバッファは、リネーム済みマイクロ命令のアウトオブオーダー実行のすべての結果を、プログラム順に戻す。つまり、リネームされたプログラムレジスタからの結果は、対応するアーキテクチャレジスタに再マップされ、指定されたプログラム実行順序に応じてアーキテクチャレジスタに入れるためにキューイングされる。実行が正常に完了して適切な結果が得られたマイクロ命令は、リタイアユニット２２５にバス２５７上で提供される。これらのリタイア済みマイクロ命令の結果は、レジスタファイル２２６にＷＢバス２５５上で書き戻される。

【００４０】

当業者が理解するように、リネーム済みマイクロ命令の実行を失敗させ得る条件はいくつかある。たとえば、プログラムの例外、一般的な割り込み、Ｉ／Ｏ割り込み、分岐（ｂｒａｎｃｈ）例外などがあるが、これらに限定されない。リオーダバッファが、リネーム済みマイクロ命令が正常に実行されなかったと判断した場合、そのリネーム済みマイクロ命令を、実行のためにディスパッチされたすべての新しいリネーム済みマイクロ命令と共に再実行（「リプレイ」）しなければならない。よって、リオーダバッファ２２４は、正常に実行されなかったリネーム済みマイクロ命令のタグをリプレイバス２５８で提供することにより、リプレイイベントを開始する。

【００４１】

正常に実行されなかったリネーム済みマイクロ命令のタグがリプレイマルチプレクサ２１４に提供されると、それに応じてリプレイマルチプレクサ２１４は、ＲＥＰＬＡＹ２５８でタグが提供されたリネーム済みマイクロ命令から始まるリネーム済みマイクロ命令の実行に適合するように、マシン状態をバックアップする。

【００４２】

また当業者は、パフォーマンスを向上させるために、マイクロプロセッサの設計者が命令の実行方法に関する仮定をしばしば行うことを理解する。たとえば、技術分野において、かなりの割合のブランチが分岐しないことがよく知られている。よって、フェッチユニット２１１は、そのような仮定に基づいて実行用の命令をキューイングするように構成され得る。ブランチが分岐しない場合、全体的な実行速度が向上する。ブランチが分岐した場合、そのブランチよりも古いすべての命令を、分岐したプログラムパスの命令で置換する必要がある。

【００４３】

マイクロプロセッサの設計者が行うもう１つの仮定は、ロードマイクロ命令が指定されたクロックサイクル数でＬ１キャッシュ２２３にヒットするというものである。この仮定は、Ｌ１キャッシュのヒット統計、たとえば９０パーセントのヒット率と、設計上Ｌ１キャッシュ２２３にアクセスするために必要なクロックサイクル数とに基づく。本願の目的では、Ｌ１キャッシュ２２３へのアクセスに４クロックサイクルを要すると想定するが、そのような数は命令の目的に合わせて選択される。その他の数のクロックサイクルも考えられる。

【００４４】

よって、リザベーションステーションＲＳ１～ＲＳＮは、古いロードマイクロ命令に対応するタグを有するリネーム済みマイクロ命令を、その古いロード命令のディスパッチから４クロックサイクルまでストールさせ、その後リネーム済みマイクロ命令を対応する実行ユニットＥＵ１～ＥＵＮにディスパッチするロジックを含み得る。このとき、古いロードマイクロ命令が４クロックサイクル以内でＬ１キャッシュ２２３にヒットし、タグ付けされたオペランドが準備できたことが仮定されている。図２には示されていないが、さらに実行ユニットＥＵ１～ＥＵＮ、ＥＵＬは、ロード操作により利用可能となったオペラン

ドにアクセスし、それらのオペランドを現在実行中のマイクロ命令に提供し得ることが指摘される。Ｌ１キャッシュ２２３にヒットしたロードについては、ディスパッチされた新しい従属マイクロ命令にオペランドが提供され、それらの新しいマイクロ命令が、他の方法で提供されたときよりもはるかに高速に実行を完了する。しかし、Ｌ１キャッシュでミスしたロード（ヒット率が９０パーセントと仮定される場合は約１０パーセントの時間）については、ロードが正常に完了した後、ヒットするという仮定の下でディスパッチされたすべての新しい従属マイクロ命令をリプレイしなければならない。よって、Ｌ１キャッシュ２２３でミスした場合、ロード実行ユニットＥＵＬは、ミスしたロード命令のタグをバスＭＩＳＳ２５３で示すことによりその旨をリオーダバッファ２２４に通知して、新しい従属命令のリプレイを開始する。

【００４５】

そのようなスキームは、パフォーマンスの観点から見て非常に効果的である。これは、今日のキャッシュ２２３のほとんどが、極めて効率的だからである。よって、推定されるキャッシュアクセスのクロックサイクル数（たとえば、４クロックサイクル）に基づいて、ロードマイクロ命令に従属するすべてのマイクロ命令を、そのロード命令のディスパッチから複数のクロックサイクルにわたりストールさせるのが一般的である。従属マイクロ命令は、それぞれのリザベーションステーションＲＳ１～ＲＳＮでストールし、ロードマイクロ命令で指定されたオペランドがＬ１キャッシュ２２３から利用可能であると仮定されるときにディスパッチされる。通常、このアクセラレーションスキームは、すべてのロード命令に対して利用される。これには、Ｌ１キャッシュ２２３以外のリソースにアクセスするロード命令も含まれる。これらの種類のロード命令は、メモリロード命令と比べて稀にしか実行されないため、メモリ以外のリソースにアクセスするロード命令のリプレイに関連するパフォーマンスへの影響は、概ね許容される。よって、ロードマイクロ命令が正常に実行する（即ち、「解決する」）ために指定されたクロックサイクル数（この例では４クロックサイクル）よりも多くを要する場合、ロード実行ユニットＥＵＬは、バスＭＩＳＳでミスを宣言し、それによって新しい従属マイクロ命令をロードの完了後にリプレイさせる。

【００４６】

上述した手法は、過去数十年にわたり、スーパースカラプロセッサ２０１のパフォーマンスの向上に役立ってきた。しかし、本発明者は、このスキームを図１に示すようなマルチコアプロセッサ構成に適用する場合に、新たな課題が生じることに気付いた。より詳細には、そのようなスキームは、Ｌ１キャッシュ２２３へのアクセスが大半を占める構成では極めて効果的だが、アンコアリソース２３０へのアクセスが増えつつあるマルチコアプロセッサ構成に適用した場合には、あまり電力効率が良くない。なぜなら、アンコアリソース２３０へのアクセス時間は、今日のＬ１キャッシュ２２３へのアクセス時間と比べて、非常に低速だからである。

【００４７】

これは、ヒューズアレイ１０８、バスユニット１０５（キャッシュ不可のロードの場合）、ＡＰＩＣ１０７、Ｉ／Ｏユニット１０６、そしておそらくはＬ２キャッシュ１０３、ＲＡＭ１０４などのアンコアリソース２３０に具体的に向けられたすべてのロードマイクロ命令が、そうしたアンコアリソース２３０からのロードのタグを含む新しい従属マイクロ命令のリプレイを招くことを意味する。そして本発明者は、パフォーマンスへの影響はそれほどではないものの、これらの新しい従属マイクロ命令の無駄な初期実行により、電力が著しく消費されることに気付いた。これは、リプレイされることが確実なロードマイクロ命令が実行ユニットＥＵ１～ＥＵＮにディスパッチされ、本来であれば電源管理メカニズムにより節約される電力が利用されるからである。

【００４８】

本発明は、今日のロードメカニズムの上述した制限及びその他の制限を、新規なスキームに基づいてロードリプレイの回数を低減することによりアウトオブオーダーマルチコアプロセッサで電力を節約する装置及び方法を提供することにより克服する。これについて

、図３～図７を参照しながら以下に説明する。

【００４９】

図３を参照すると、非コアリソースからのロードのための節電メカニズムを備えた本発明のマルチコアプロセッサを示すブロック図３００が提示されている。ブロック図３００は、４つのプロセッサコア３０１が配置されたデバイスダイ３１０を示している。４つのコア３０１は、本発明を明瞭に教示する目的でのみ示されており、以下に説明する原理及び詳細は、非コアリソースからの特定のロード操作のアクセス時間がオンコアキャッシュのアクセス時間よりも長い、任意の数のコア３０１を備えるプロセッサに適用できることが指摘される。

【００５０】

図１のマルチコアプロセッサと同様に、本発明のマルチコアプロセッサは、典型的にはコア３０１と同じダイ３１０に配置されるアンコアリソース３０２を含み得る。アンコアリソース３０２は、各コア３０１により高速バス３１１～３１４を通じてアクセスされる。したがって図３００は、Ｌ２キャッシュ３０３、ランダムアクセスメモリ（ＲＡＭ）３０４、バスユニット３０５、入力／出力ユニット３０６、ＡＰＩＣ３０７、ヒューズアレイ３０８などであるがこれらに限定されない例示的な共有リソースを示す。よって、バスＢ１　３１１は、ＣＯＲＥ１　３０１がアンコア３０２にアクセスすることを可能にする。バスＢ２　３１２は、ＣＯＲＥ２　３０１がアンコア３０２にアクセスすることを可能にする。バスＢ３　３１３は、ＣＯＲＥ３　３０１がアンコア３０２にアクセスすることを可能にする。バスＢ４　３１４は、ＣＯＲＥ４　３０１がアンコア３０２にアクセスすることを可能にする。典型的な構成では、マルチコアプロセッサは、デバイスダイ３１０の外部にある、システムメモリ（メモリバスＭＥＭを通じてアクセス）、入力／出力要素（バスＩ／Ｏを通じてアクセス）、システム制御要素（バスＣＴＲＬを通じてアクセス）などであるがこれらに限定されない他の要素（図示せず）に接続される。制御要素は、周辺構成要素相互接続エクスプレス（ＰＣＩ－ｅ）要素、周辺構成要素相互接続（ＰＣＩ）要素、ユニバーサルシリアルバス（ＵＳＢ）要素、グラフィックアダプタ、コプロセッサ、及びプロセッサ間通信要素を含み得るが、これらに限定されない。

【００５１】

図１のマルチコアプロセッサとは対照的に、本発明のマルチコアプロセッサは、リプレイ低減装置要素３２０を各コア３０１内に備える。一実施形態では、リプレイ低減装置３２０は、オンコアキャッシュ（図示せず）以外のリソースに向けられたロードを検出し、それらのロードが解決するまで、すべての新しい従属マイクロ命令のディスパッチをストールさせ、本来であればリプレイイベントを発生させるすべての指示のアサーションを除外するように構成されている。よって、コア３０１内の１つ又は複数の実行ユニット（図示せず）は、新しい従属マイクロ命令のディスパッチをストールさせることにより電力管理モードに入り、それによって本来であればダイ３１０で浪費される電力を節約することができる。

【００５２】

動作中、オペレーティングシステムの制御下にある各コア３０１は、システムメモリからフェッチされた関連する命令を実行することができ、対象の用途に対応するオペランドを操作する。１つ又は複数のコア３０１は、１つ又は複数のアンコアリソース３０２にアクセスすることが必要な場合があり、対応するバスＢ１～Ｂ４を通じて、制御された態様で、それら１つ又は複数のアンコアリソース３０２にアクセスする。たとえば、電源初期化時に、１つ又は複数のコア３０１は、ヒューズアレイ３０８からのロード操作を実行して構成パラメータを取得することができ、又はＲＡＭ３０４からのロードを実行してマイクロコードのパッチ及び／又は他の構成情報を取得することができる。通常の動作時に、コア３０１はＬ２キャッシュ３０３にアクセスして、システムメモリからキャッシュされた可能性がありオンコアキャッシュ（たとえば、Ｌ１キャッシュ）に存在していないメモリオペランドを読み取る／書き込むことができる。コア３０１は、バスユニット３０５にアクセスしてシステムメモリとの間で読み取り／書き込みを実行することができ、又はＩ

／Ｏユニット３０６を利用してＩ／Ｏバスを通じてＩ／Ｏ操作を実行することができる。コア３０１は、バスユニット３０５にアクセスして制御要素との間で制御データの読み取り／書き込みを行うことができる。コア３０１は、さらにＡＰＩＣ３０７にアクセスして割り込み操作を実行することができる。

【００５３】

これらのアンコア３０２からのロードの結果としてミスを自動的に宣言して新しい従属マイクロ命令ストリームをリプレイさせる代わりに、リプレイ低減装置３２０は、ロードが解決するまで新しい従属マイクロ命令ストリームの実行をストールさせ、それによって実行ユニットの電力管理機能を利用できるようにする。一実施形態では、リプレイ低減装置３２０は、具体的にアンコアリソース３０２に向けられているわけではないが、ミスの指示を招くことが確実である他の種類のロードをも検出し得る。これらの他の種類のロードは、Ｉ／Ｏロード、指定されたサイクル数を必要とするロード、第２レベルアドレス変換に関連するロード（即ち、ネステッドページング、ｘ８６拡張ページテーブルロード）などページテーブルウォークを必要とすることが既知であるロード、ｘ８６特殊バスサイクル（たとえば、シャットダウン、停止、フラッシュなど）の実行により生じるロード、及びキャッシュ不可のメモリ領域又は書き込み結合領域に解決されることが既知であるロードを含むが、これらに限定されない。他の実施形態では、完了するために指定されたサイクル数よりも多くを必要とする可能性が極めて高い任意の種類のロード操作を検出することを意図する。

【００５４】

図４を参照すると、図３に示す各コア３０１の例示的なコアステージを示すブロック図４００が提示されている。このブロック図は、ダイ３１０に配置されたプロセッサコア４０１を示している。コア４０１はフェッチステージ４１１を含み、このフェッチステージがトランスレータステージ４１２にバス４４１を通じて接続されている。トランスレータステージ４１２は、リネームステージ４１３にバス４４２を通じて接続されている。リネームステージ４１３は、リプレイマルチプレクサステージ４１４にバス４４３を通じて接続されている。リプレイマルチプレクサ４１４は、複数のリザベーションステーションＲＳ１～ＲＳＮ　４２１．１～４２１．Ｎと、拡張ロードリザベーションステーションＥＲＳＬ４２１．Ｌとに、リザベーション及び保持バスＨＯＬＤＹ４４４を通じて接続されている。リザベーションステーションＥＲＳＬは、アンコアストール要素４６１を含む。各リザベーションステーションＲＳ１～ＲＳＮ、ＥＲＳＬは、対応する実行ユニットＥＵ１～ＥＵＮ　４２２．１～４２２．Ｎ、ＥＵＬ４２２．Ｌに、対応するディスパッチバス４５１．１～４５１．Ｎ、４５１．Ｌを通じて接続されている。リザベーションステーションＲＳ１～ＲＳＮ、ＥＲＳＬは、レジスタファイル４２６にレジスタバス４４５を通じて接続されている。

【００５５】

実行ユニットＥＵＬを除き、他の実行ユニットＥＵ１～ＥＵＮは、整数ユニット、浮動小数点ユニット、マルチメディアユニット、ストアユニットなど、今日のスーパースカラプロセッサに典型的なユニットを含み得る。実行ユニットＥＵＬは、ロードユニット４２２．Ｌであり、その主な機能は、図３を参照しながら上述したシステムメモリ、システムＩ／Ｏ、及びアンコアリソース４３０を含むがこれらに限定されない多様なリソースからオペランドをロードすることである。実行ユニットＥＵＬは、アンミス要素ＵＭＩＳＳ４６２をさらに含む。

【００５６】

よって、ロードユニットＥＵＬは、Ｌ１キャッシュ４２３にバス４５４を通じて接続され、アンコア４３０にバス４５６を通じて接続されている。メモリオペランドについて、ロードユニット４２２．ＬはまずＬ１キャッシュ４２３にアクセスする。ロードがＬ１キャッシュ４２３でミスした場合、ロードユニット４２２．Ｌはアンコア４３０のＬ２キャッシュ（図示せず）にアクセスしなければならない。また実行ユニットＥＵ１～ＥＵＮ、ＥＵＬは、リオーダバッファ４２４にバス４５２を通じて接続されている。さらに、ロー

ドユニットＥＵＬは、リオーダバッファ４２４にバスＭＩＳＳ４５３を通じて接続されている。リオーダバッファ４２４は、リプレイマルチプレクサ４１４にバスＲＥＰＬＡＹ４５８を通じて接続され、リタイアユニット４２５にバス４５７を通じて接続され、リザベーションステーションＲＳ１～ＲＳＮ、ＥＲＳＬにＨＯＬＤＹバス４４４を通じて接続されている。リタイアユニット４２５は、レジスタファイル４２６に書き戻しＷＢバス４５５を通じて接続されている。

【００５７】

図４に示されたコアステージは、本発明の側面を明瞭に教示するための例として提示されていることが指摘される。なぜなら、これらのコアステージは、今日のアウトオブオーダープロセッサコアの例示であるからである。ただし、当業者は本明細書に記載された本発明の側面及び特徴を、アーキテクチャ及び対象の用途に応じて要求され得る他のプロセッサコアステージ構成に適用できることが指摘される。

【００５８】

動作中、プログラム命令（図示せず）がフェッチユニット４１１によってメモリ（図示せず）からフェッチされる。ｘ８６互換プロセッサコア４０１の場合、これらのプログラム命令は、ｘ８６　ＩＳＡに適合する。プログラム命令は、トランスレータ４１２にバス４４１上で順番に提供される。トランスレータ４１２は、プログラム命令により指定された操作を実行するために、対応する実行ユニットＥＵ１～ＥＵＮ、ＥＵＬにサブ操作を指示する１つ又は複数のマイクロ命令にプログラム命令を変換する。マイクロ命令は、リネームユニット４１３にバス４４２上で提供される。リネームユニット４１３において、一部のマイクロ命令で指定されているアーキテクチャレジスタ（即ち、オペランドのレジスタ位置）が、プロセッサコア４０１のハードウェアレジスタ（図示せず）に再マップされる。これは、独立したマイクロ命令ストリームの実行並列性を高めるためである。またリネームユニット４１３は、連続的なプログラム順に応じて、各マイクロ命令にタグ付けする。マイクロ命令のソースオペランドフィールド及びデスティネーションオペランドフィールドにも、１つ又は複数のオペランドが依存する新しいマイクロ命令のタグがタグ付けされる。リネーム済みマイクロ命令は、リプレイマルチプレクサ４１４にバス４４３上で提供される。

【００５９】

リプレイマルチプレクサ４１４は、リネームされた各マイクロ命令のオペコードを読み取って、実行のための適切な実行ユニットＥＵ１～ＥＵＮ、ＥＵＬを判断する。詳細には、リネームされたロードマイクロ命令は、ロード実行ユニットＥＵＬにより実行される。よって、リプレイマルチプレクサ４１４は、１つ又は複数のリネーム済みマイクロ命令を、１つ又は複数のリザベーションステーションＲＳ１～ＲＳＮ、ＥＲＳＬに提供して、対応する実行ユニットＥＵ１～ＥＵＮ、ＥＵＬへのディスパッチを待機させる。

【００６０】

各リザベーションステーションＲＳ１～ＲＳＮ、ＥＲＳＬは、レジスタファイル４２６にアクセスして、キューイングされているリネーム済みマイクロ命令の操作に必要なオペランドを読み取る。古いリネーム済みマイクロ命令のタグがタグ付けされていないリネーム済みマイクロ命令（即ち、古いリネーム済みマイクロ命令に依存していないリネーム済みマイクロ命令）は、対応する実行ユニットＥＵ１～ＥＵＮ、ＥＵＬにすぐにディスパッチされて実行される。後述する例外を除き、従属する新しいリネーム済みマイクロ命令（即ち、まだ実行が完了していない古いリネーム済みマイクロ命令のタグを含む、リネーム済みマイクロ命令）は、通常はタグ付けされた従属オペランドが利用可能になるまで、リザベーションステーションＲＳ１～ＲＳＮ、ＥＲＳＬにより保持される。タグ付けされた従属オペランドが利用可能になると、それらのオペランドが従属する新しいリネーム済みマイクロ命令に提供され、それらの新しいマイクロ命令が対応する実行ユニットＥＵ１～ＥＵＮ、ＥＵＬにディスパッチされて実行される。実行ユニットＥＵ１～ＥＵＮ、ＥＵＬは、マイクロ命令を実行していないときに、節電機能を実行することもできる。実行ユニットＥＵ１～ＥＵＮ、ＥＵＬの内部のクロックは、マイクロ命令を実行していないときは

シャットダウンされ、それによって電力が大幅に節約される。

【００６１】

リネーム済みマイクロ命令及びその結果は、リオーダバッファ４２４にバス４５２を通じて提供される。リオーダバッファ４２４は、リネーム済みマイクロ命令のアウトオブオーダー実行のすべての結果を、プログラム順に戻す。つまり、リネームされたプログラムレジスタからの結果は、対応するアーキテクチャレジスタに再マップされ、指定されたプログラム実行順序に応じてアーキテクチャレジスタに入れるためにキューイングされる。実行が正常に完了して適切な結果が得られたマイクロ命令は、リタイアユニット４２５にバス４５７上で提供される。これらのリタイア済みマイクロ命令の結果は、レジスタファイル４２６にＷＢバス４５５上で書き戻される。

【００６２】

リオーダバッファ４２４が、リネーム済みマイクロ命令が正常に実行されなかったと判断した場合、そのリネーム済みマイクロ命令を、実行用にディスパッチされたすべての新しい従属リネーム済みマイクロ命令と共にリプレイしなければならない。よって、リオーダバッファ４２４は、正常に実行されなかったリネーム済みマイクロ命令のタグをリプレイバス４５８で提供することにより、リプレイイベントを開始する。

【００６３】

正常に実行されなかったリネーム済みマイクロ命令のタグがリプレイマルチプレクサ４１４に提供されると、それに応じてリプレイマルチプレクサ４１４は、ＲＥＰＬＡＹ４５８でタグが提供されたリネーム済みマイクロ命令から始まるリネーム済みマイクロ命令の実行に適合するように、マシン状態をバックアップする。

【００６４】

後述する例外を除き、本発明は、新しいロードマイクロ命令に依存するマイクロ命令を、そのロードマイクロ命令が指定されたクロックサイクル数でＬ１キャッシュ４２３にヒットすると想定して、そのロードマイクロ命令がディスパッチされてから指定されたクロックサイクル数にわたりストールするように構成されたリザベーションステーションＲＳ１～ＲＳＮを含む。一実施形態では、指定されたクロックサイクル数は４クロックサイクルである。その他の数のクロックサイクルも考えられる。

【００６５】

よって、後述する例外を除き、リザベーションステーションＲＳ１～ＲＳＮは、古いロードマイクロ命令に対応するタグを有するリネーム済みマイクロ命令を、その古いロード命令のディスパッチから４クロックサイクルまでストールさせ、その後リネーム済みの新しいマイクロ命令を対応する実行ユニットＥＵ１～ＥＵＮにディスパッチするロジックを含む。このとき、古いロードマイクロ命令が４クロックサイクル以内でＬ１キャッシュ４２３にヒットし、タグ付けされたオペランドが準備できたことが仮定されている。図４には示されていないが、さらに実行ユニットＥＵ１～ＥＵＮ、ＥＵＬは、ロード操作により利用可能となったオペランドにアクセスし、それらのオペランドを現在実行中のマイクロ命令に提供し得ることが指摘される。Ｌ１キャッシュ４２３にヒットしたロードなど、指定されたサイクル数未満で完了したロードについては、ディスパッチされた新しい従属マイクロ命令にオペランドが提供され、それらの新しいマイクロ命令が、他の方法で提供されたときよりもはるかに高速に実行を完了する。Ｌ１キャッシュにおいてミスしたロードなど、指定されたクロックサイクルよりも多くを要するロードについては、ロードが正常に完了した後、ヒットするという仮定の下でディスパッチされたすべての新しい従属マイクロ命令をリプレイしなければならない。よって、Ｌ１キャッシュ４２３でミスした場合、ロード実行ユニットＥＵＬは、ミスしたロード命令のタグをバスＭＩＳＳ４５３で示すことによりその旨をリオーダバッファ４２４に通知して、新しい命令のリプレイを開始する。

【００６６】

しかし、本発明はまた、アンコアストールロジック４６１を拡張ロードリザベーションステーションＥＲＳＬ４２１．Ｌに含めることにより、上述したアクセラレーションスキ

ームの例外を提供する。アンコアストールロジック４６１は、１つ又は複数のロードマイクロ命令タイプを検出して、それら１つ又は複数のロードマイクロ命令タイプのマイクロ命令に依存する新しいマイクロ命令を、そのオペランドが利用可能になるまでストールさせることにより、１つ又は複数の実行ユニットＥＵ１～ＥＵＮで節電を実装する。またアンミスロジック４６２は、１つ又は複数のロードマイクロ命令タイプを検出して、それら１つ又は複数のロードマイクロ命令タイプのマイクロ命令がオペランドを取得するために指定されたクロックサイクル数よりも多くを必要とする場合に、バスＭＩＳＳ４５３でのミスのアサーションを除外する。これにより、１つ又は複数のロードマイクロ命令タイプのマイクロ命令が実行を完了することが可能となり、１つ又は複数のマイクロ命令タイプのマイクロ命令に依存する新しいマイクロ命令がリザベーションステーションＲＳ１～ＲＳＮでストールされているため、それらの新しいマイクロ命令のリプレイが不要になる。一実施形態では、リザベーションステーションＲＳ１～ＲＳＮ、ＥＲＳＬは、１つ又は複数のマイクロ命令タイプの検出されたマイクロ命令に関する情報（たとえば、タグ）を相互間で通信し、及びバスＨＯＬＤＹ４４４を通じてリオーダバッファ４２４に通信して、新しい従属マイクロ命令のストールを開始する。１つ又は複数のロードマイクロ命令タイプのマイクロ命令が実行を完了すると、リオーダバッファ４２４は、完了した１つ又は複数のロードマイクロ命令タイプのマイクロ命令のタグをＨＯＬＤＹ４４４で提供することにより、ストールされた新しい従属マイクロ命令を解放してディスパッチさせるようリザベーションステーションＲＳ１～ＲＳＮに指示する。

【００６７】

有利なことに、本発明は、システムメモリからキャッシュされたオペランドに対応するロードマイクロ命令に関するパフォーマンスを効率化し、１つ又は複数のマイクロ命令タイプのロードマイクロ命令に関連するリプレイの回数を実質的に低減する。それにより、実行ユニットＥＵ１～ＥＵＮが、実装された従属関係のストールによって空になったときに、節電モードに入れるようにする。

【００６８】

したがって、たとえば、ヒューズアレイ３０８、バスユニット３０５、ＡＰＩＣ３０７、Ｉ／Ｏユニット３０６、Ｌ２キャッシュ３０３、ＲＡＭ３０４などのアンコアリソース４３０に具体的に向けられたロードマイクロ命令により、アンコアリソース４３０からのそれらのロードのタグを有する新しい従属マイクロ命令がリプレイされることがなくなる。

【００６９】

一実施形態では、１つ又は複数のロードマイクロ命令タイプは、指定されたアンコアリソース４３０からのロードを、他のタイプのロードと共に含み得る。この他のタイプのロードは、Ｉ／Ｏロード、特定のサイクル数を必要とするロード、ページテーブルウォークを必要とすることが既知であるシステムメモリからのロード、ｘ８６特殊バスサイクル（たとえば、シャットダウン、停止、フラッシュなど）の実行により生じるロード、キャッシュ不可のメモリ領域に解決されることが既知であるロード、及び書き込み結合メモリ領域に解決されることが既知であるロードを含むが、これらに限定されない。他の実施形態では、完了するために指定されたサイクル数よりも多くを必要とする可能性が極めて高い任意の種類のロード操作を検出することを意図する。

【００７０】

一実施形態では、アンコアストール要素４６１とアンミス要素４６２とは、本発明のプロセッサコア４０１の初期化（たとえば、電源投入又はリセット）時に、規定のロードマイクロ命令タイプを検出するように構成され得る。規定のロードマイクロ命令タイプは、初期化時に、ヒューズアレイ３０８の指定された位置から読み込まれ得る。別の実施形態では、各コア４０１は、プログラミングを通じて、異なるタイプの規定のロードマイクロ命令をヒューズアレイ３０８で検出するように構成され得る。ここで、各コア４０１に関連するタイプは、ヒューズアレイ３０８の対応する位置に対してプログラムされ得る。さらなる実施形態では、規定のロードマイクロ命令タイプは、マルチコアデバイス３１０へ

のジョイントテストアクショングループ（ＪＴＡＧ）インターフェイス（図示せず）を通じて、電源投入時又はリセット時にＲＡＭ３０４に対してプログラムされ得る。規定のロードマイクロ命令タイプは、その後の初期化時にＲＡＭ３０４の指定された位置から読み込まれる。

【００７１】

図５を参照すると、図４のアンコアストール要素４６１の詳細を示すブロック図５００が提示されている。ストール要素４６１は、アンコアロードオペコード検出ロジック５０１に接続されたマイクロ命令レジスタ５１０を含む。マイクロ命令レジスタ５１０は、マイクロ命令タグフィールドＯＰ　ＴＡＧ５１１と、オペコードフィールドＭＩＣＲＯ　ＯＰ５１２と、ソースＡフィールドＳＲＣ　Ａ５１３と、タグＡフィールドＴＡＧ　Ａ５１４と、ソースＢフィールドＳＲＣ　Ｂ５１５と、タグＢフィールドＴＡＧ　Ｂ５１６と、ソースＣフィールドＳＲＣ　Ｃ５１７と、タグＣフィールドＴＡＧ　Ｃ５１８とを含む。検出ロジック５０１は、バス４４４に結び付けられたホールド信号ＨＯＬＤＹを生成する。

【００７２】

当業者が理解するように、ｘ８６　ＩＳＡなどの今日のＩＳＡは、直接（ｄｉｒｅｃｔ）、間接（ｉｎｄｉｒｅｃｔ）、イミディエイト（ｉｍｍｅｄｉａｔｅ）、及び相対（ｒｅｌａｔｉｖｅ）を含むが、これらに限定されない複数の異なるオペランドアドレッシングモードを提供する。結果として、ソースフィールドＳＲＣ　Ａ～Ｃの１つ又は複数はオペランドを含み得、１つ又は複数はオペランドの位置（結果のデスティネーションを含む）を指定し得る。そこで、ストール要素４６１の動作について、本願を多数のＩＳＡに幅広く適用できるよう、ソースフィールドＳＲＣ　Ａ～Ｃの内容に関しては総称的に触れながら説明する。

【００７３】

動作的に言うと、マイクロ命令がリプレイマルチプレクサ４１４により提供されると、ロードマイクロ命令がマイクロ命令レジスタ５１０に入力される。ＯＰ　ＴＡＧは、レジスタ５１０の現在のマイクロ命令のタグを含み、ＭＩＣＲＯ　ＯＰは、そのオペコードを含む。ＴＡＧ　Ａの内容は、ＳＲＣ　Ａの内容が依存する古いマイクロ命令のタグを含み得る。ＴＡＧ　Ｂの内容は、ＳＲＣ　Ｂの内容が依存する古いマイクロ命令のタグを含み得る。ＴＡＧ　Ｃの内容は、ＳＲＣ　Ｃの内容が依存する古いマイクロ命令のタグを含み得る。検出ロジック５０１は、ＭＩＣＲＯ　ＯＰの内容を読み取るように構成されている。ＭＩＣＲＯ　ＯＰが、レジスタ５１０内の現在のマイクロ命令に依存する他のリザベーションステーションＲＳ１～ＲＳＮ内の新しいマイクロ命令をストールさせる上述した規定のロードオペコードのいずれかを含んでいない場合、検出ロジック５０１はＨＯＬＤＹをデアサートし、それによってＲＳ１～ＲＳＮに対してそれらの新しいマイクロ命令をいずれディスパッチしてもよいことを示す。しかし、ＭＩＣＲＯ　ＯＰが、レジスタ５１０内の現在のマイクロ命令に依存する他のリザベーションステーションＲＳ１～ＲＳＮ内の新しいマイクロ命令をストールさせる上述した規定のロードオペコードのいずれかを含んでいる場合、検出ロジック５０１はＨＯＬＤＹをアサートし、ＯＰ　ＴＡＧの内容をバス４４４に置き、それによってＲＳ１～ＲＳＮに対して、レジスタ５１０内の現在のマイクロ命令により規定されるロードが完了し、ロードの結果が新しい従属マイクロ命令に提供されるまで、それらの新しいマイクロ命令をストールさせなければならないことを示す。ロードが完了すると、リオーダバッファ４２４はＨＯＬＤＹをデアサートし、それによってストールを解放する。

【００７４】

図６を参照すると、図４のリザベーションステーションＲＳ１～ＲＳＮのそれぞれの詳細を示すブロック図が提示されている。リザベーションステーションは、従属確認ロジック６０１に接続されたマイクロ命令レジスタ６１０を含む。マイクロ命令レジスタ６１０は、マイクロ命令タグフィールドＯＰ　ＴＡＧ６１１と、オペコードフィールドＭＩＣＲＯ　ＯＰ６１２と、ソースＡフィールドＳＲＣ　Ａ６１３と、タグＡフィールドＴＡＧ

Ａ６１４と、ソースＢフィールドＳＲＣ　Ｂ６１５と、タグＢフィールドＴＡＧ　Ｂ６１６と、ソースＣフィールドＳＲＣ　Ｃ６１７と、タグＣフィールドＴＡＧ　Ｃ６１８とを含む。従属確認ロジック６０１は、レディ信号ＲＥＡＤＹを生成し、バス４４４に結び付けられたホールド信号ＨＯＬＤＹを監視する。

【００７５】

レジスタ６１０のフィールド６１１～６１８の内容は、図５を参照して上述した同様の名前のフィールドと同じである。確認ロジック６０１は、ソースタグフィールドＴＡＧ　Ａ～Ｃの内容を読み取るようにさらに構成される。タグフィールドＴＡＧ　Ａ～Ｃのいずれかの内容がアサート時にＨＯＬＤＹのタグと一致する場合、レジスタ６１０内のマイクロ命令は、レジスタ６１０内のマイクロ命令が依存するロードマイクロ命令が完了し、ロードを通じて得られたオペランドが対応するソースフィールドＳＲＣ　Ａ～Ｃに提供され、リオーダバッファ４２４がＨＯＬＤＹをデアサートするまで、ストールさせられる。ＨＯＬＤＹがデアサートされると、確認ロジック６０１がＲＥＡＤＹをアサートして、レジスタ６１０内のマイクロ命令を対応する実行ユニットＥＵ１～ＥＵＮにディスパッチする準備ができたことを示す。

【００７６】

タグフィールドＴＡＧ　Ａ～Ｃのいずれかの内容がアサート時にＨＯＬＤＹのタグと一致しない場合、確認ロジック６０１はＲＥＡＤＹをアサートして、レジスタ６１０内のマイクロ命令を対応する実行ユニットＥＵ１～ＥＵＮにディスパッチする準備ができたことを示す。

【００７７】

図７を参照すると、図４のアンコアミス要素４６２の詳細を示すブロック図７００が提示されている。アンコアミス要素４６２は、ロードミス除外ロジック７０１に接続されたマイクロ命令レジスタ７１０を含む。マイクロ命令レジスタ７１０は、マイクロ命令タグフィールドＯＰ　ＴＡＧ７１１と、オペコードフィールドＭＩＣＲＯ　ＯＰ７１２と、ソースＡフィールドＳＲＣ　Ａ７１３と、タグＡフィールドＴＡＧ　Ａ７１４と、ソースＢフィールドＳＲＣ　Ｂ７１５と、タグＢフィールドＴＡＧ　Ｂ７１６と、ソースＣフィールドＳＲＣ　Ｃ７１７と、タグＣフィールドＴＡＧ　Ｃ７１８とを含む。ミス除外ロジック７０１は、ノーミス信号ＮＯＭＩＳＳを生成する。

【００７８】

レジスタ７１０のフィールド７１１～７１８の内容は、図５及び図６を参照して上述した同様の名前のフィールドと同じである。除外ロジック７０１は、ＭＩＣＲＯ　ＯＰの内容を読み取るように構成されている。ＭＩＣＲＯ　ＯＰが、新しい従属マイクロ命令をストールさせる上述した規定のロードオペコードのいずれかを含んでいない場合、ロードミス除外ロジック７０１は、信号ＮＯＭＩＳＳをデアサートし、それによって対応するロード実行ユニットＥＵＬ４２２．Ｌに、信号ＭＩＳＳの状態を通常のロード命令実行手続きに従って管理するよう通知する。ＭＩＣＲＯ　ＯＰが規定のオペコードのいずれかを含んでいる場合、除外ロジック７０１はＮＯＭＩＳＳをアサートし、それによって対応するロード実行ユニットＥＵＬ４２２．Ｌに、レジスタ７１０のマイクロ命令の実行時にＭＩＳＳのアサーションを除外するよう通知する。

【００７９】

本発明の上述した要素は、本明細書に記載された機能及び操作を実行するように構成される。本発明の要素は、ロジック、回路、デバイス、若しくはマイクロコード（即ち、マイクロ命令若しくはネイティブ命令）、又はロジック、回路、デバイス、若しくはマイクロコードの組合せ、又は本発明の上述した機能及び操作を実行するために利用される等価の要素を含む。これらの操作及び機能を達成するために利用される要素は、マルチコアマイクロプロセッサ内の他の機能及び／又は操作を実行するために利用される他の回路、マイクロコードなどと共有され得る。

【００８０】

本発明及び対応する詳細な説明の各部分は、ソフトウェア、即ちコンピュータメモリ内

でのデータビットの操作のアルゴリズム及び記号表現で表される。これらの説明及び表現は、当業者がその作業の実体を他の当業者に効果的に伝えるためのものである。本明細書で使用される意味、及び一般的に使用される意味でのアルゴリズムとは、所望の目的に至る一連の自己矛盾のないステップと理解される。これらのステップは、物理量を物理的に操作することを必要とする。通常、必須ではないものの、これらの物理量は、格納し、移動し、組合せ、比較し、及びその他の方法で操作することができる、光学的、電気的、又は磁気的な信号の形式をとる。これらの信号をビット、値、要素、記号、文字、用語、数字などとして表すことは、特に共通使用のために、往々にして便利であることが立証されている。

【００８１】

ただし、これらの用語及び同様の用語のすべては適切な物理量と関連付けられるものであり、これらの物理量に適用される便利なラベルに過ぎないことに留意すべきである。特に明記されない限り、又は記載から明らかであるように、「処理」、「演算」、「計算」、「判断」、「表示」などの用語は、コンピュータシステムのレジスタ及びメモリ内で物理的かつ電子的な量として表されたデータを操作して、コンピュータシステムのメモリやレジスタなどの情報ストレージ、送信デバイス、又はディスプレイデバイス内で物理量として同様に表された他のデータに変換する、コンピュータシステム、マイクロプロセッサ、中央処理装置、又は同様の電子コンピューティングデバイスの動作及びプロセスを意味する。

【００８２】

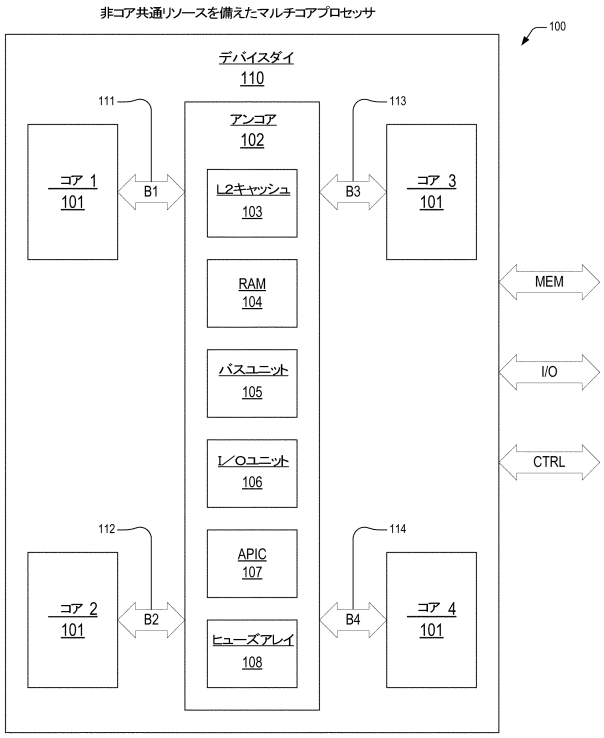
また、本発明のソフトウェアにより実装される側面は、典型的には何らかの形式のプログラムストレージ媒体で符号化され、又は何らかの種類の伝送媒体で実装されることに留意されたい。プログラムストレージ媒体は、電子的（たとえば、読み取り専用メモリ、読み取り専用フラッシュメモリ、電子的プログラム可能読み取り専用メモリ、ランダムアクセスメモリ）、磁気的（たとえば、フロッピー（登録商標）ディスク又はハードドライブ）、又は光学的（たとえば、コンパクトディスク読み取り専用メモリ、即ち「ＣＤ－ＲＯＭ」）であり得、読み取り専用又はランダムアクセス可能であり得る。同様に、伝送媒体は、金属トレース、ツイストペア（ｔｗｉｓｔｅｄ　ｗｉｒｅ　ｐａｉｒ）、同軸ケーブル、光ファイバ、又は技術分野で知られるその他の何らかの適切な伝送媒体であり得る。本発明は、特定の実装のこれらの側面によって限定されない。

【００８３】

上記で開示された特定の実施形態は、あくまでも例示のためのものである。当業者は、開示された概念及び特定の実施形態を基盤として容易に使用して、本発明と同じ目的を達成する他の構造を設計又は修正できること、並びに添付の特許請求の範囲に記載された本発明の範囲から逸脱せずに、多様な変更、交換、及び代替を加えられることを理解する。

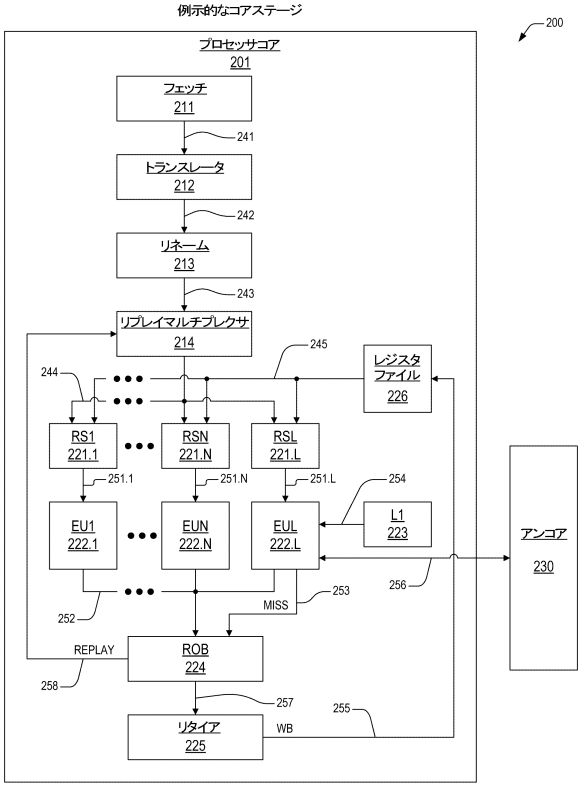
【図１】

（先行技術）

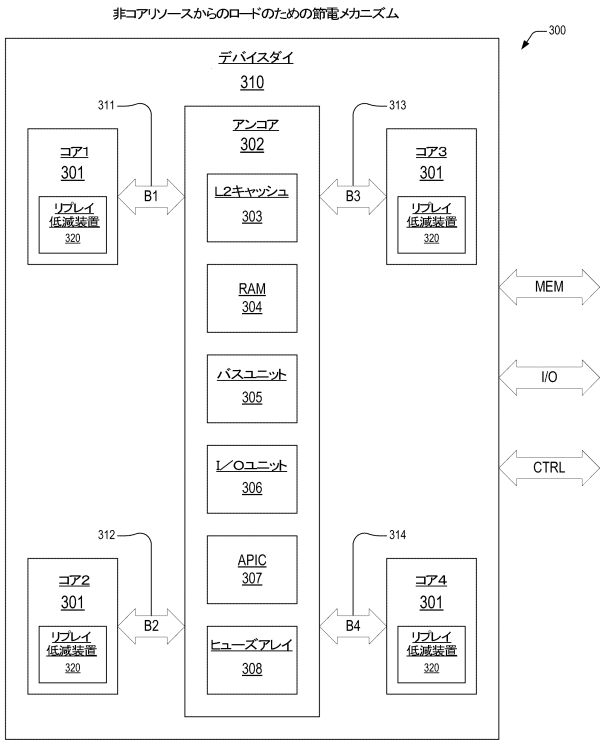


【図２】

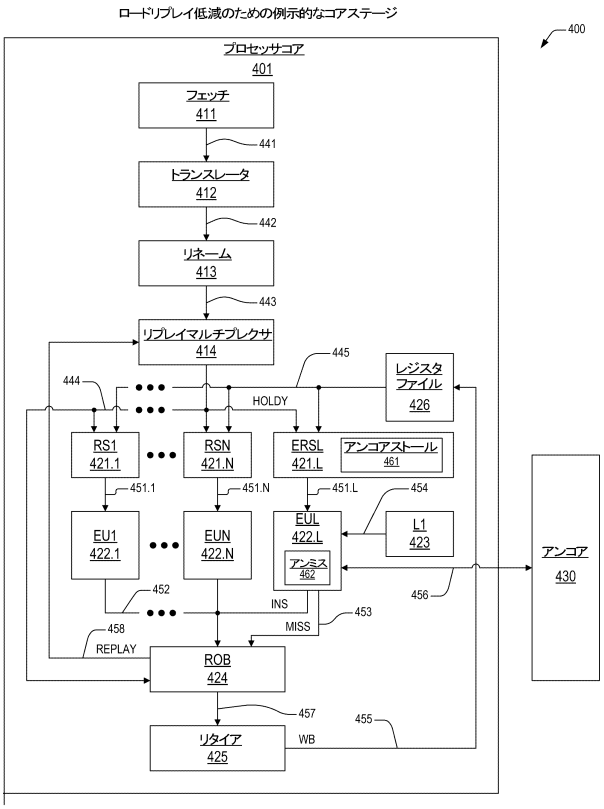
（先行技術）



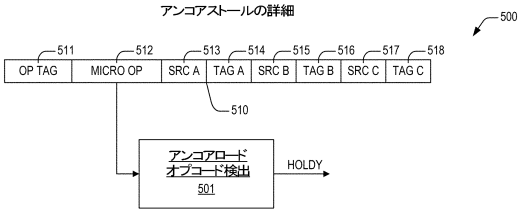
【図３】



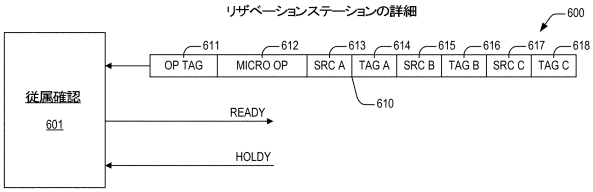
【図４】



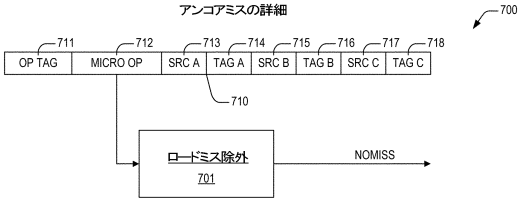
【図５】


アンコアストールの詳細
500
511
512
513
514
515
516
517
518
OP TAG
MICRO OP
SRC A
TAG A
SRC B
TAG B
SRC C
TAG C
510
アンコアロード
オプコード検出
501
HOLDY

【図６】


リザベーションステーションの詳細
600
611
612
613
614
615
616
617
618
OP TAG
MICRO OP
SRC A
TAG A
SRC B
TAG B
SRC C
TAG C
610
従属確認
601
READY
HOLDY

【図７】


アンコアミスの詳細
700
711
712
713
714
715
716
717
718
OP TAG
MICRO OP
SRC A
TAG A
SRC B
TAG B
SRC C
TAG C
710
ロードミス除外
701
NOMISS

フロントページの続き

(72)発明者　コル，ジェラード，エム
アメリカ合衆国　７８７２６　テキサス州，オースティン，コンチョス・トレイル　１１００８
(72)発明者　エディー，コリン
アメリカ合衆国　７８７０１　テキサス州，オースティン，ヌエイシス・ストリート　３６０，ユニット　４１０７
(72)発明者　ヘンリー，ジー，グレン
アメリカ合衆国　７８７４６　テキサス州，オースティン，レイク・クリフ・トレイル　４１１

審査官　三坂　敏夫

(56)参考文献　特開２００６－３１８０５１（ＪＰ，Ａ）
特開２００６－１４６９５３（ＪＰ，Ａ）
特開２０１２－１９８８０３（ＪＰ，Ａ）
特開２００３－２８０８９６（ＪＰ，Ａ）
特開２０１０－２１８３６７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　９／３０
９／３０５－９／３０８
９／３１５－９／３４
９／３５－　９／４２