

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4181500号  
(P4181500)

(45) 発行日 平成20年11月12日(2008.11.12)

(24) 登録日 平成20年9月5日(2008.9.5)

(51) Int. Cl.		F I			
<b>G06F 12/08</b>	<b>(2006.01)</b>	G06F 12/08	541D		
<b>G06F 12/16</b>	<b>(2006.01)</b>	G06F 12/08	565		
		G06F 12/08	507Z		
		G06F 12/16	320D		
		G06F 12/16	320F		

請求項の数 20 (全 11 頁)

(21) 出願番号	特願2003-544595 (P2003-544595)	(73) 特許権者	597004720
(86) (22) 出願日	平成14年11月6日(2002.11.6)		サン・マイクロシステムズ・インコーポレ ーテッド
(65) 公表番号	特表2006-502460 (P2006-502460A)		Sun Microsystems, In c.
(43) 公表日	平成18年1月19日(2006.1.19)		アメリカ合衆国 カリフォルニア州 95 054 サンタ クララ ネットワーク サークル 4150
(86) 国際出願番号	PCT/US2002/035681	(74) 代理人	100078282
(87) 国際公開番号	W02003/042828		弁理士 山本 秀策
(87) 国際公開日	平成15年5月22日(2003.5.22)	(74) 代理人	100062409
審査請求日	平成17年10月11日(2005.10.11)		弁理士 安村 高明
(31) 優先権主張番号	60/337,734	(74) 代理人	100113413
(32) 優先日	平成13年11月9日(2001.11.9)		弁理士 森下 夏樹
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 プロッキングなく、キャッシュ参照間に直面するビットエラーを修正する方法および装置

(57) 【特許請求の範囲】

【請求項1】

キャッシュメモリへの参照の間に直面するビットエラーを修正する方法であって、  
アプリケーションの実行中に、該キャッシュメモリへの参照を実行することであって、  
該参照を実行することは、該キャッシュメモリから、該参照のタグと該参照のデータラインとを含むデータアイテムおよび関連するエラー訂正コードを取り出すことを含む、ことと、

該取り出されたデータアイテムからエラー訂正コードを計算することと、  
該計算されたエラー訂正コードを該関連するエラー訂正コードと比較することと  
を包含し、

該計算されたエラー訂正コードが、該関連するエラー訂正コードと整合せず、ビットエラーが生じたことを示す場合、該方法は、

該キャッシュメモリと関連する1つ以上のレジスタのセットのあるレジスタに該参照の識別子を格納することをさらに包含し、

該ビットエラーが該タグ内で検出された場合、該方法は、

該アプリケーションの実行を継続することであって、該アプリケーションの実行を継続することは、該参照を行ったスレッドが、該参照が完了することを待ちつつ、該アプリケーションの他のスレッドの実行を継続することを含む、ことと、

該アプリケーションの継続された実行の後の時間に、該レジスタから該格納された識別子を読み出し、対応するデータアイテムを訂正することと

をさらに包含する、方法。

【請求項 2】

前記後の時間に前記キャッシュメモリの前記ビットエラーを修正することであって、  
前記レジスタから前記参照の識別子を読み出すことと、  
該識別子を利用して、前記データアイテムおよび前記関連するエラー訂正コードを取り出すことと、  
該関連するエラー訂正コードを利用して、訂正されたデータアイテムおよびエラー訂正コードを生成することと、  
該キャッシュメモリに該訂正されたデータアイテムおよびエラー訂正コードを格納することと

10

により行われることをさらに包含する、請求項 1 に記載の方法。

【請求項 3】

前記ビットエラーを修正することは、オペレーティングシステムのチックタイマーによって決定される周期的な間隔で該ビットエラーを修正することを含む、請求項 2 に記載の方法。

【請求項 4】

前記関連するエラー訂正コードとともに前記データアイテムを取り出すことは、  
前記キャッシュメモリから前記参照の前記タグを該タグの関連するエラー訂正コードとともに取り出すことと、  
該参照の前記データラインを該データラインの関連するエラー訂正コードとともに取り出すことと

20

を含む、請求項 1 ~ 3 のいずれか一項に記載の方法。

【請求項 5】

前記ビットエラーが前記タグ内で検出された場合、前記方法は、前記キャッシュメモリのミスバッファに前記参照を格納させるように、キャッシュミスを導き、それにより、該ビットエラーが修正された後の時間に、該参照がリトライされ得ることをさらに包含する、請求項 4 に記載の方法。

【請求項 6】

前記参照が前記ミスバッファに格納された場合、前記方法は、前記ビットエラーが修正された後に、該参照を該ミスバッファから起動させるようにキャッシュフィル動作を導くことをさらに包含する、請求項 5 に記載の方法。

30

【請求項 7】

前記ビットエラーが前記データライン内で検出された場合、前記方法は、前記キャッシュメモリとプロセッサとの間のデータパスに配置された回路を介して該ビットエラーを訂正することをさらに包含する、請求項 4 に記載の方法。

【請求項 8】

1 つ以上のレジスタの前記セットのレジスタは、エントリが上書きされることを可能にする環状バッファとして組織化される、請求項 1 ~ 7 のいずれか一項に記載の方法。

【請求項 9】

前記キャッシュメモリは、メモリ階層内のレベル 2 ( L 2 ) キャッシュまたはより下位のキャッシュである、請求項 1 ~ 8 のいずれか一項に記載の方法。

40

【請求項 10】

前記エラー訂正コードは、シングルビットエラーを訂正し、ダブルビットエラーを検出する、請求項 1 ~ 9 のいずれか一項に記載の方法。

【請求項 11】

キャッシュメモリへの参照の間に直面するビットエラーを修正する装置であって、  
該キャッシュメモリと、  
アプリケーションから参照を受け取るように構成された該キャッシュメモリの入力と、  
該キャッシュメモリから、該参照のタグと該参照のデータラインとを含む該参照のデータアイテムおよび関連するエラー訂正コードを取り出すように構成された取り出しメカニ

50

ズムと、

該取り出されたデータアイテムからエラー訂正コードを計算するように構成されたエラー訂正コード回路と、

該計算されたエラー訂正コードを該関連するエラー訂正コードと比較することによってビットエラーを検出するように構成された比較器と、

キャッシュ参照の間に検出されたエラーの識別子を格納する1つ以上のレジスタのセットと、

エラーロギングメカニズムであって、ビットエラーが生じた場合、該参照の識別子を該1つ以上のレジスタのセットのあるレジスタに格納するように構成されるエラーロギングメカニズムと、

ビットエラー修正メカニズムと

を備え、

該ビットエラー修正メカニズムは、該ビットエラーが該タグ内で検出された場合、

該アプリケーションの継続した実行を可能にすることであって、該アプリケーションの実行を継続することは、該参照を行ったスレッドが、該参照が完了することを待ちつつ、該アプリケーションの他のスレッドの実行を継続することを含む、ことと、

該継続された実行の後の時間に、該レジスタから該格納された識別子を読み出し、対応するデータアイテムを訂正することと

を実行するように構成されている、装置。

【請求項12】

前記ビットエラー修正メカニズムは、

前記レジスタから前記参照の識別子を読み出し、

該識別子を利用して、前記データアイテムおよび前記関連するエラー訂正コードを取り出し、

該関連するエラー訂正コードを利用して、訂正されたデータアイテムおよびエラー訂正コードを生成し、

前記キャッシュメモリに該訂正されたデータアイテムおよびエラー訂正コードを格納する

ように構成されている、請求項11に記載の装置。

【請求項13】

前記ビットエラー修正メカニズムは、オペレーティングシステムのチックタイマーによって決定される周期的な間隔でビットエラーを修正するように構成されている、請求項12に記載の装置。

【請求項14】

前記取り出しメカニズムは、

前記キャッシュメモリから前記参照の前記タグを該タグの関連するエラー訂正コードとともに取り出し、

前記参照の前記データラインを該データラインの関連するエラー訂正コードとともに取り出す

ように構成されている、請求項11～13のいずれか一項に記載の装置。

【請求項15】

前記ビットエラーが前記タグ内で検出された場合、前記エラーロギングメカニズムは、前記参照を前記キャッシュメモリのミスバッファに格納させるようにキャッシュミストリガーすることにより、該ビットエラーが修正された後の時間に該参照が取り出され得るように、構成されている、請求項14に記載の装置。

【請求項16】

前記参照が前記ミスバッファに格納された場合、前記装置は、前記タグの前記ビットエラーが修正された後に、該参照を該ミスバッファから起動させるようにキャッシュフィル動作を導くようにさらに構成される、請求項15に記載の装置。

【請求項17】

10

20

30

40

50

前記キャッシュメモリとプロセッサとの間のデータパスに配置された回路であって、データラインのビットエラーを訂正する回路をさらに備える、請求項 14 に記載の装置。

【請求項 18】

前記 1 つ以上のレジスタのセットのレジスタは、エントリが上書きされることを可能にする環状バッファとして組織化される、請求項 11 ~ 17 のいずれか一項に記載の装置。

【請求項 19】

前記キャッシュメモリは、メモリ階層内のレベル 2 ( L 2 ) キャッシュまたはより下位のキャッシュである、請求項 11 ~ 18 のいずれか一項に記載の装置。

【請求項 20】

前記エラー訂正コードは、シングルビットエラーを訂正し、ダブルビットエラーを検出する、請求項 11 ~ 19 のいずれか一項に記載の装置。

10

【発明の詳細な説明】

【技術分野】

【0001】

(発明の分野)

本発明は、コンピューティングシステム内のフォルトトレランスを提供するメカニズムに関する。より詳細には、本発明は、スレッドおよび/またはプロセスを実行すると同時にブロッキングなく、キャッシュ参照間に直面するビットエラーを修正する方法および装置に関する。

【背景技術】

20

【0002】

(関連技術)

現在の半導体技術の急速な発展は、マイクロプロセッサチップ上に多量のキャッシュを組み込むことを可能にする。例えば、いくつかのマイクロプロセッサは、複数のプロセッサと、多量のレベル 2 ( L 2 ) キャッシュにアクセスする関連するレベル 1 ( L 1 ) キャッシュとを含む。ここで、これらの構造の全ては、同一のマイクロプロセッサチップ上に常駐する。マイクロプロセッサチップ上に L 2 キャッシュを位置付けることは、劇的に、L 2 キャッシュにアクセスするために必要となる時間を減少させ、それにより、マイクロプロセッサシステムの性能を増大させ得る。

【0003】

30

しかし、大量のオンチップキャッシュは、ランダムなビットエラーを生じやすい。この問題に対する 1 つのソリューションは、これらのエラーを検出かつ訂正するエラー訂正コードを利用することである。マイクロプロセッサチップの外部に位置する半導体メモリは、多くの場合、各データワードのエラー訂正コードを格納する付加的なスペースを含む。データワードが最初にメモリに格納されるときに、そのデータワードからエラー訂正コードが計算され、このエラー訂正コードは、メモリ内のデータワードに沿って格納される。データワードがその後メモリから取り出されるときに、エラー訂正コードもまた取り出される。同時に、新しいエラー訂正コードが、取り出されたデータワードに対して計算される。新しいエラー訂正コードが取り出されたエラー訂正コードと異なる場合、ビットエラーが、エラー訂正コードのデータワードのどちらかに生じる。この場合、エラー訂正コードを利用して、ビットエラーを訂正することができる。

40

【0004】

データを読み出し、エラーを検出し、かつエラーを訂正するこのプロセスは、読み出し - 修正 - 書き込み ( R M W ) 動作に関する。R M W 動作を実装してエラーを訂正することは、キャッシュアクセスに、コンピュータシステム性能を大幅に低下させ得るさらなる遅延を導き、貴重なオンチップのリアルエステートを消費するさらなる回路を必要とし得る。結果的に、大量のオンチップキャッシュは、現在、ビットエラーを検出および訂正する R M W 動作をサポートしていない。既存のオンチップキャッシュは、単に、読み出し動作中のデータエラーをチェックし、かつ、データエラーが検出された場合、システム全体が停止し、それにより、他のリクエストがキャッシュにアクセスすることを妨げる。もしくは

50

は、トラップが発生し得、オーバーフローバッファを利用して、未処理のトランザクションを積み上げる。

【0005】

シングルチップマルチプロセッサシステムでは、複数のプロセッサおよびスレッドからの大量のトランザクションが、与えられたいくらかの時間で未処理となり得る。したがって、全てのトランザクションを停止させるメカニズム、または未処理のリクエストを積み上げるメカニズムを提供することは、著しい量の複雑性を導き、かつ、貴重なオンチップのリアルエステート消費する。

【発明の開示】

【発明が解決しようとする課題】

10

【0006】

したがって、必要とされるものは、オンチップキャッシュの設計を著しく複雑にし、メモリサブシステムを介する処理からの未処理のトランザクションを停止させることなく、オンチップキャッシュへの参照の間に直面するビットエラーを修正する方法および装置である。

【課題を解決するための手段】

【0007】

(要旨)

本発明のある実施形態は、キャッシュメモリへの参照の間に直面するビットエラーを修正するシステムを提供する。アプリケーションの実行の間に、システムは、キャッシュメモリからデータアイテムおよび関連するエラー訂正コードを取り出すことによって、キャッシュメモリへの参照を実行する。次に、システムは、取り出されたデータアイテムからエラー訂正コードを計算し、計算されたエラー訂正コードを関連するエラー訂正コードと比較する。計算されたエラー訂正コードが関連するエラー訂正コードと整合しない場合、ビットエラーが生じる。この場合、システムは、ビットエラーが後の時間に修正され得るように、キャッシュメモリに関連する1つ以上のレジスタのセット内のレジスタに参照の識別子を格納する。システムはまた、アプリケーションが実行し続けることを可能にする。

20

【0008】

この実施形態のある変形では、システムは、続いて、キャッシュメモリ内のビットエラーを修正する。このことは、レジスタから参照の識別子を読み出すこと、および、識別子を利用して、キャッシュからデータアイテムおよび関連するエラー訂正コードを取り出すことを含む。さらに、このことは、関連するエラー訂正コードを利用して、訂正されたデータアイテムおよびエラー訂正コードを生成すること、および、その後、キャッシュメモリへ訂正されたデータアイテムおよびエラー訂正コードを格納することを含む。

30

【0009】

この実施形態のある変形では、ビットエラーを修正することは、オペレーティングシステムのチックタイマーによって決定される周期的な間隔でビットエラーを修正することを含む。

【0010】

40

この実施形態のある変形では、関連するエラー訂正コードとともにデータアイテムを取り出すことは、キャッシュメモリから、参照のタグをタグの関連するエラー訂正コードとともに取り出すことを含む。このことは、さらに、参照のデータラインをデータラインの関連するエラー訂正コードとともに取り出すことを含む。

【0011】

この変形のある変形では、ビットエラーがタグ内で検出された場合、システムは、キャッシュのミスバッファに参照を格納させるようにキャッシュミスを導き、それにより、参照は、ビットエラーが修正された後に取り出され得る。このことは、参照を行ったスレッドが、参照が完了することを待ちつつ、システムがアプリケーションのほかのスレッドの実行を続けることを可能にする。さらなる変形では、参照がミスバッファ内に格納される

50

場合、システムは、ビットエラーが修正された後に、ミスバッファから参照を起動させるようにキャッシュフィル動作を導く。

【0012】

この実施形態のある変形では、ビットエラーがデータライン内で検出された場合、システムは、キャッシュメモリとプロセッサとの間のデータバスに配置された回路を介してビットエラーを訂正する。

【0013】

この実施形態のある変形では、1つ以上のレジスタのセットは、エントリが上書きされることを可能にする環状バッファとして組織化される。

【0014】

この実施形態のある変形では、キャッシュメモリは、メモリ階層のレベル2(L2)キャッシュまたはより下位のキャッシュである。

【0015】

この実施形態のある変形では、エラー訂正コードは、シングルビットエラーを訂正し、ダブルビットエラーを検出する。

【発明を実施するための最良の形態】

【0016】

以下の説明は、任意の当業者が本発明を製作および利用することを可能にし、かつ、特定の用途およびその要件の文脈で提供される。開示される実施形態に対するさまざまな変更が、当業者には容易に明らかであり、本明細書中で規定される一般原理は、本発明の意図および範囲から逸脱することなく、ほかの実施形態および用途に適用され得る。したがって、本発明は、示される実施形態に限定されず、本明細書中に開示される原理および特徴と一致する最大範囲で許容され得る。

(マルチプロセッサシステム)

図1は、本発明の実施形態によるエラー訂正を提供するレベル2(L2)キャッシュ106を含むマルチプロセッサシステム100を示す。なお、多くのマルチプロセッサシステム100が、単一の半導体チップ101内に位置付けられる。より詳細には、半導体チップ101は、多くのプロセッサ110、120、130、および140を含み、これらは、それぞれレベル1(L1)キャッシュ112、122、132、および142を含む。なお、L1キャッシュ112、122、132、および142は、別個の命令およびデータキャッシュであってもよく、あるいは、統一された命令/データキャッシュであってもよい。L1キャッシュ112、122、132、および142は、L2キャッシュ106に接続され、L2キャッシュ106は、以下の図2~4を参照してより詳細に説明されるエラー訂正コードを利用する。L2キャッシュ106は、メモリコントローラ104を介してオフチップメモリ102に接続される。

【0017】

本発明の1つの実施形態では、L1キャッシュ112、122、132、および142は、ライトスルーキャッシュである。ライトスルーキャッシュは、L1キャッシュ112、122、132、および142に対する全ての更新が、L2キャッシュ106に自動的に伝達されることを意味する。このことは、コヒーレンスプロトコルを単純化する。なぜなら、プロセッサ110は、L1キャッシュ112に存在するデータアイテムを必要とし、プロセッサ110が、L1キャッシュ112がデータのソースにたどり着くことを待つ必要なく、L2キャッシュ106からデータを取り出すことができるからである。

【0018】

本発明は、L2キャッシュ106の文脈で記載されるが、本発明は、概して、L3キャッシュ、または可能であればL1キャッシュでさえ含む、任意のタイプのキャッシュに適用され得る。さらに、本発明は、セットアソシアティブキャッシュの文脈で説明されるが、本発明はまた、ダイレクトマッピングキャッシュに適用され得る。

(キャッシュ構造)

図2は、本発明の実施形態による図1からのL2キャッシュ106の構造を示す。従来

10

20

30

40

50

のキャッシュメモリのように、L2キャッシュ106は、タグおよびステータス情報204を格納するメモリアレイ、ならびにデータライン218を格納するメモリアレイを含む。なお、L2キャッシュ106がセットアソシアティブである場合は、タグおよびステータスアレイ204ならびにデータアレイ218は、各セットに対する複数のエントリを含む。

**【0019】**

L2キャッシュ106はまた、ミスバッファ230を含む。ミスバッファ230は、L2キャッシュ106にアクセスしている間にミスをしたメモリリクエストを格納し、かつ、キャッシュラインがメモリ102から戻るのを待つ。なお、ミスバッファ230の各エントリは、キャッシュラインがメモリ102からリクエストされたかどうかを示すビットを含む。

10

**【0020】**

簡単のために、所与の参照がキャッシュヒットであるかを判定するコンパレータのような、多くの通常のキャッシュ構造体は、図2には示されない。

**【0021】**

L2キャッシュ106はまた、エラーレジスタ232、タグ202用のエラー訂正コード(ECC)、データ216用のECC、ECC回路206および220、コンパレータ208および226、ならびにマルチプレクサ(MUX)212および222を含む、エラー訂正コードに関する多くの新しい構造体を含む。

20

**【0022】**

エラーレジスタ232は、エラーが検出されたキャッシュ参照用のアドレスを含む。本発明のある実施形態では、エラーレジスタ232は、ヘッドポインタおよびテールポインタを有する環状バッファとして組織化される。しかし、従来の環状バッファと異なり、この環状バッファが満杯にある場合、環状バッファのより古いエントリが、単純にも上書きされる。つまり、訂正されないエラーがいくつか生じ得る。しかし、このシステムは、依然として機能を保持し、エラーは、次回にこのアドレスがアクセスされたときに訂正される可能性が高いことに留意されたい。

**【0023】**

タグ202用のECCは、タグおよびステータスアレイ204の各エントリ用のECCを含む。同様に、データ216用のECCは、データアレイ218の各エントリ用のECCを含む。

30

**【0024】**

タグおよびステータスアレイ204から所与のセットにアクセスされる1つ以上のタグは、ECC回路306を通過し、ECC回路306は、各タグ/ステータスワード用の新しいECCを生成する。コンパレータ208は、新しいECCをタグ202用のECCから取り出されたECCと比較する。これらのECCが整合しなければ、ECC信号210がアサートされる。

**【0025】**

回路は、訂正されたタグ/ステータスワード214を生成するマルチプレクサ212を任意に含み得る。この回路は、本発明のいくつかの実施形態では、タグ/ステータスワードを訂正するために利用される。

40

**【0026】**

データアレイ218から所与のセットにアクセスされた1つ以上のデータラインは、ECC回路220を通過し、ECC回路220は、各データライン用の新しいECCを生成する。コンパレータ226は、それぞれの新しいECCをタグ202用のECCから取り出されたECCと比較する。これらのECCが整合しない場合、エラー信号228はアサートされる。

**【0027】**

この回路はまた、訂正されたデータライン224を生成するマルチプレクサ222を含む。

50

(エラーを取り扱うプロセス)

図3は、本発明の実施形態によるキャッシュ参照の間のエラーを取り扱うプロセスを示すフローチャートである。L2キャッシュ106にアクセスするリクエストを受け取ると、システムは、リクエストのアドレスによって指定されたセットのタグを読む(ステップ302)。このシステムは、その後、このタグを比較して、リクエストがキャッシュヒットを生成するかどうかを判定する。同時に、図2に示される回路はまた、エラーが存在する場合のタグエラーを検出する(ステップ304)。

【0028】

タグエラーが検出される場合、システムは、エラーレジスタ232にリクエストのアドレスのログを取り(ステップ308)、キャッシュをミスさせて、キャッシュラインがメモリ102から取り出されることを待つリクエストに沿って、リクエストをミスバッファ230に格納させる(ステップ310)。なお、サイクル時間またはロードレイテンシが増大しない限り、タグがリクエストのアドレスと比較される前に、タグを訂正する時間はない。

【0029】

次に、このシステムは、リクエストを作成したスレッドがリクエストが完了することを待つ間に、他のプロセス/スレッドが実行し続けることを可能にする(ステップ322)。このように、マルチプロセッサシステム100は、単一のスレッドがタグエラーに直面したために、実行を完全に停止させる必要はない。

【0030】

エラーが、ステップ304で検出されなかった場合、システムは、キャッシュヒットが存在するかどうかを判定する(ステップ312)。キャッシュヒットが存在しなければ、システムは、キャッシュミスを生じ、キャッシュラインをメモリ102からリクエストさせる(ステップ314)。システムは、その後、キャッシュミスが戻ることを待たない他のスレッド/プロセスを実行させ続けることを可能にする(ステップ322)。

【0031】

ステップ312においてキャッシュヒットが存在する場合、システムは、データアレイから対応するデータラインを読み(ステップ316)、図2に示される回路を用いてデータエラーを検出することを試みる(ステップ318)。

【0032】

データエラーが存在する場合は、システムは、エラーレジスタ232にエラーを生じさせたリクエストのアドレスのログを取り、プロセッサまでのデータパス上の下流の回路のデータエラーを訂正する(ステップ320)。

【0033】

次に、システムは、実行を続ける(ステップ322)。

(エラーを修正するプロセス)

図4は、本発明の実施形態によるエラーを修正する方法を示すフローチャートである。このシステムは、オペレーティングシステムのチックタイマーから妨害を受け取ることによって開始する(ステップ402)。

【0034】

チックタイマーに関連する通常のプロセッサ管理タスクを実行させることに加えて、システムは、エラーレジスタ232が空であるかどうかを判定する(ステップ404)。空であれば、エラーは発生しておらず、システムは、通常のプロプログラムの実行に戻る。

【0035】

エラーレジスタ232が空でなければ、システムは、エラーレジスタからアドレスを読み出す(ステップ406)。次に、システムは、そのアドレスによって指定されたセットの全てのタグおよびデータを読み出す(ステップ408)。システムは、その後、訂正されたタグおよびデータをL2キャッシュ106へ格納して戻す前(ステップ410)に、必要となるときにタグおよびデータを訂正する(ステップ412)。

【0036】

10

20

30

40

50



タグエラーおよびキャッシュヒットが存在する場合、システムは、フェイクキャッシュフィル動作を実行して、ミスバッファ230に待機し得る任意のリクエストを起動する(ステップ414)。なお、ミスバッファ230は、ミスバッファ230がそのリクエストされたデータのキャッシュフィルが実行されたと考えるまで、キャッシュへアクセスすることを試みない。

【0037】

タグエラーおよびキャッシュミスが存在する場合、システムは、メモリ階層の下位レベルからキャッシュラインをリクエストする(ステップ416)。システムは、アドレス情報により満たされたすべてのエラーレジスタに対して、ステップ406、408、410、412、414、および416を繰り返す。

10

【0038】

なお、エラーレジスタ232が過度に満杯になり、レジスタが上書きされた場合、システムは、ミスバッファ230のエラーをチェックする。ミスバッファ230は、上書きされたタグエラーが修正されることを待ち得る(ステップ418)。この方法で上書きされた任意のタグエラーは、ステップ406、408、410、412、414、および416を繰り返すことによって訂正され得る。

【0039】

本発明の1つの実施形態では、エラーを修正する多くのプロセスは、ソフトウェアでインプリメントされる一方で、多くのエラーを検出しかつログを取るプロセスは、ハードウェアでインプリメントされる。

20

【0040】

本発明の実施形態の以上の説明は、例示および説明のためにのみ示された。これらは、包括的である、または、開示される形式に本発明を制限することを意図していない。したがって、多くの改変および変形が、当業者には明らかである。さらに、上述の開示は、本発明を制限することを意図しない。本発明の範囲は、添付の特許請求の範囲によって規定される。

【図面の簡単な説明】

【0041】

【図1】図1は、本発明の実施形態によるエラー訂正を提供するL2キャッシュを含むマルチプロセッサシステムを示す。

30

【図2】図2は、本発明の実施形態による図1からのL2キャッシュの構造を示す。

【図3】図3は、本発明の実施形態によるキャッシュ参照の間のエラーを取り扱うプロセスを示すフローチャートである。

【図4】図4は、本発明の実施形態によるエラーを修正する方法を示すフローチャートである。

【図 1】

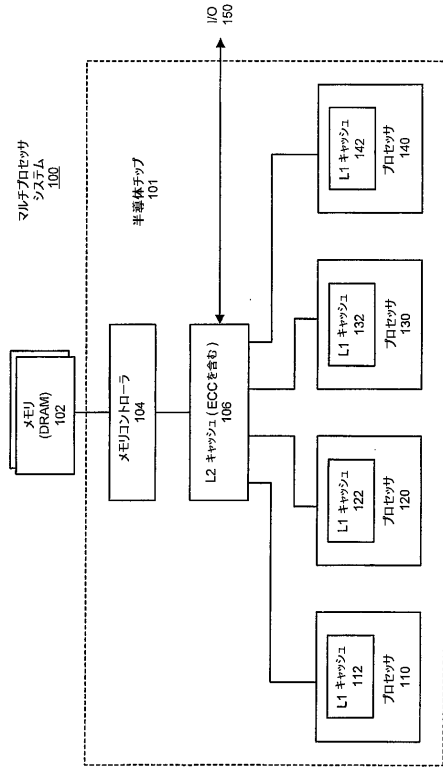


FIG. 1

【図 2】

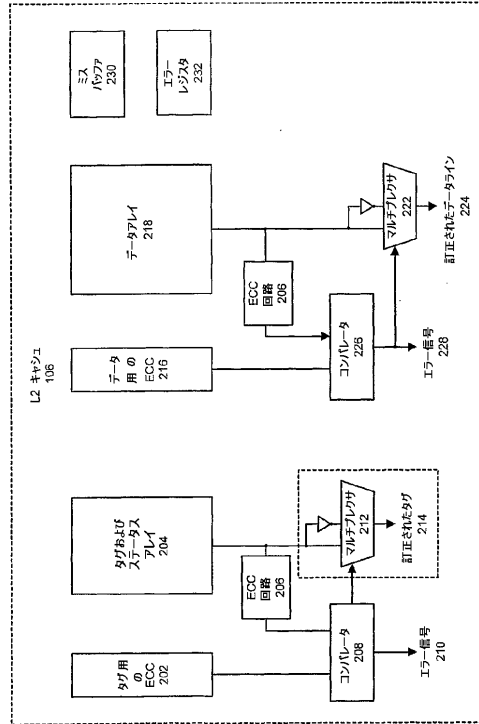


FIG. 2

【図 3】

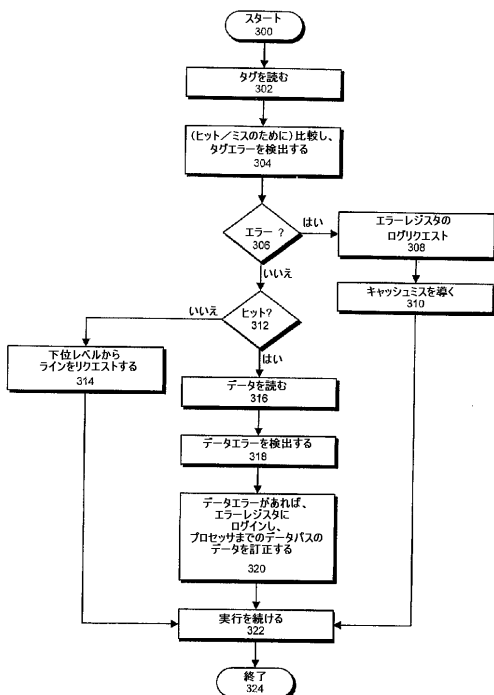


FIG. 3

【図 4】

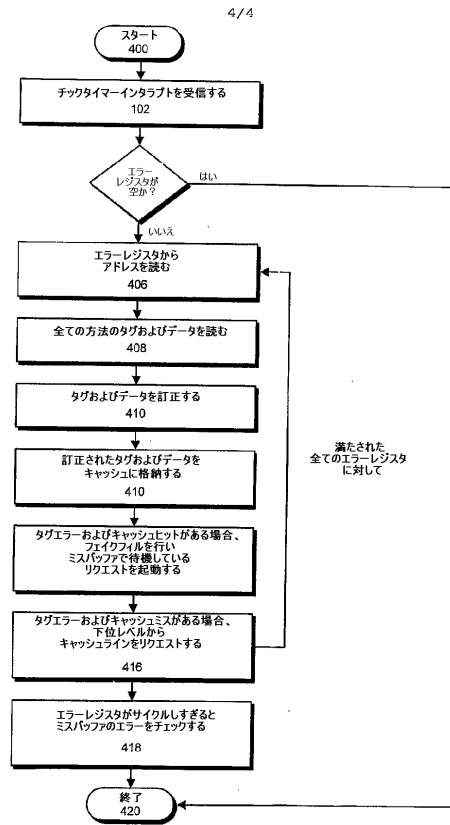


FIG. 4

---

フロントページの続き

- (72)発明者 トレンブレー, マーク  
アメリカ合衆国 カリフォルニア 94025, メンロ パーク, ハンナ ウェイ 140
- (72)発明者 チャウダリー, シャイレンダー  
アメリカ合衆国 カリフォルニア 94105, サン フランシスコ, ミッション ストリー  
ト 680, アパートメント 22エフ

審査官 清木 泰

- (56)参考文献 特開昭59-168997(JP,A)  
特開平06-075864(JP,A)  
特開平09-146836(JP,A)  
特開平08-095856(JP,A)  
特開平05-233472(JP,A)  
特開昭63-070357(JP,A)  
特開昭62-130444(JP,A)  
特開昭62-130443(JP,A)  
特開昭61-290556(JP,A)  
特開昭60-189553(JP,A)  
特開昭59-014198(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F12/08-12/12

G06F12/16