

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第5803614号  
(P5803614)

(45) 発行日 平成27年11月4日(2015.11.4)

(24) 登録日 平成27年9月11日(2015.9.11)

(51) Int.Cl.

F 1

G06F 12/08 (2006.01)

G06F 12/08 12/08 553B

G06F 12/16 (2006.01)

G06F 12/08 579

G06F 12/08 541Z

G06F 12/16 340Q

請求項の数 12 (全 31 頁)

(21) 出願番号

特願2011-259796 (P2011-259796)

(22) 出願日

平成23年11月29日(2011.11.29)

(65) 公開番号

特開2013-114441 (P2013-114441A)

(43) 公開日

平成25年6月10日(2013.6.10)

審査請求日

平成26年10月6日(2014.10.6)

(73) 特許権者 000002185

ソニー株式会社

東京都港区港南1丁目7番1号

(74) 代理人 100086841

弁理士 脇 篤夫

(74) 代理人 100114122

弁理士 鈴木 伸夫

(74) 代理人 100167704

弁理士 中川 裕人

(72) 発明者 肥後 豊

東京都港区港南1丁目7番1号 ソニー株式会社内

(72) 発明者 細見 政功

東京都港区港南1丁目7番1号 ソニー株式会社内

最終頁に続く

(54) 【発明の名称】不揮発性キャッシュメモリ、不揮発性キャッシュメモリの処理方法、コンピュータシステム

## (57) 【特許請求の範囲】

## 【請求項 1】

キャッシュデータの記憶に用いられる不揮発性メモリ部と、

上記不揮発性メモリ部に対してのキャッシュデータの読み出し及び書き込みを制御するとともに、電力供給遮断の準備として、待機状態データを生成して上記不揮発性メモリ部に記憶する待機準備処理を行い、また電力供給再開の際に、上記待機状態データを用いて上記不揮発性メモリ部に記憶されたキャッシュデータの復帰処理を行うキャッシュコントローラと、

を備えた不揮発性キャッシュメモリ。

## 【請求項 2】

上記キャッシュコントローラは、

上記待機準備処理として、少なくとも上記不揮発性メモリ部上に記憶されているキャッシュデータについてのエラー訂正を可能とするエラー訂正データを含む上記待機状態データを生成して上記不揮発性メモリ部上に記憶させ、

上記復帰処理として、上記待機状態データにおける上記エラー訂正データを用いて上記不揮発性メモリ部に記憶されたキャッシュデータのエラー訂正処理を行う請求項1に記載の不揮発性キャッシュメモリ。

## 【請求項 3】

上記不揮発性メモリ部には、キャッシュデータ記憶に用いられるキャッシュラインが複数設定されており、

上記キャッシュコントローラは、

上記待機準備処理として、少なくとも一部のキャッシュラインについて、そのキャッシュラインに記憶されているキャッシュデータのエラー訂正可能なエラー訂正データを含む待機状態データを生成し、該待機状態データを他のキャッシュラインに記憶させ、かつ当該他のキャッシュラインを、待機状態データを記憶したキャッシュラインであることを示す状態とする処理を行う請求項1に記載の不揮発性キャッシュメモリ。

【請求項4】

上記エラー訂正データを含む待機状態データには、さらに当該エラー訂正データでエラー訂正を行うキャッシュデータが記憶されたキャッシュラインのアドレス情報が含まれている請求項3に記載の不揮発性キャッシュメモリ。

10

【請求項5】

上記キャッシュコントローラは、上記待機状態データを、その待機状態データが対象とするキャッシュデータを記憶したキャッシュラインに対応する領域に記憶させる請求項3に記載の不揮発性キャッシュメモリ。

【請求項6】

上記不揮発性メモリ部には、キャッシュデータ記憶領域及び待機状態データ領域を有するキャッシュラインが複数設定されており、

上記キャッシュコントローラは、

上記待機準備処理として、少なくとも一部のキャッシュラインについて、そのキャッシュラインに記憶されているキャッシュデータのエラー訂正可能なエラー訂正データを含む待機状態データを生成し、該待機状態データを上記待機状態データ領域に記憶させ、かつ当該キャッシュラインを、待機状態データを記憶したキャッシュラインであることを示す状態とする処理を行う請求項1に記載の不揮発性キャッシュメモリ。

20

【請求項7】

上記キャッシュコントローラは、

上記待機準備処理として、少なくとも上記不揮発性メモリ部上に記憶されているキャッシュデータについてのエラー検出を可能とするエラー検出データを含む上記待機状態データを生成して上記不揮発性メモリ部上に記憶させ、

上記復帰処理として、上記待機状態データにおける上記エラー検出データを用いて上記不揮発性メモリ部に記憶されたキャッシュデータのエラー検出処理を行う請求項1に記載の不揮発性キャッシュメモリ。

30

【請求項8】

上記キャッシュコントローラは、

上記エラー検出処理においてエラーを検出したキャッシュデータを無効化する処理を行う請求項7に記載の不揮発性キャッシュメモリ。

【請求項9】

上記キャッシュコントローラは、電力供給を遮断する待機状態に移行することを要求する待機信号を受け取ることで、上記待機準備処理を行う請求項1に記載の不揮発性キャッシュメモリ。

【請求項10】

40

キャッシュデータの記憶に用いられる不揮発性メモリ部を有する不揮発性キャッシュメモリの処理方法として、

電力供給遮断の準備として、待機状態データを生成して上記不揮発性メモリ部上に記憶する待機準備処理を行い、

電力供給再開の際に、上記待機状態データを用いて上記不揮発性メモリ部に記憶されたキャッシュデータの復帰処理を行う不揮発性キャッシュメモリの処理方法。

【請求項11】

プロセッサと、

不揮発性キャッシュメモリと、

電力制御回路と、

50

メインメモリとを有し、

上記電力制御回路は、上記プロセッサ及び上記不揮発性キャッシュメモリへの電力供給を制御し、

上記プロセッサは、上記不揮発性キャッシュメモリを介しての上記メインメモリに対するデータの書き込み又は読み出しを実行するとともに、上記電力制御回路が電力供給を遮断する際に、上記不揮発性キャッシュメモリに対して、待機状態に移行することを要求する待機信号を伝達し、また上記電力制御回路が電力供給を再開する際に、上記不揮発性キャッシュメモリに対して電力供給を再開して動作状態に移行することを要求する復帰信号を伝達し、

上記不揮発性キャッシュメモリは、

キャッシュデータの記憶に用いられる不揮発性メモリ部と、

上記不揮発性メモリ部に対してのキャッシュデータの読み出し及び書き込みを制御するとともに、上記待機信号を受け取ることで、待機状態データを生成して上記不揮発性メモリ部上に記憶する待機準備処理を行い、また上記復帰信号を受け取ることで、上記待機状態データを用いて上記不揮発性メモリ部に記憶されたキャッシュデータの復帰処理を行うキャッシュコントローラと、を備えている、

コンピュータシステム。

#### 【請求項 1 2】

上記不揮発性キャッシュメモリの上記キャッシュコントローラは、

上記待機準備処理を完了することに応じて、上記プロセッサに対して、待機状態への移行が完了したことを通知する待機完了信号を伝達し、

また上記復帰処理を完了することに応じて、上記プロセッサに対して、動作状態への移行が完了したことを通知する復帰完了信号を伝達する請求項 1 1に記載のコンピュータシステム。

#### 【発明の詳細な説明】

##### 【技術分野】

##### 【0 0 0 1】

本開示は、不揮発性キャッシュメモリ、不揮発性キャッシュメモリの処理方法、及び不揮発性キャッシュメモリを有するコンピュータシステムに関する。

##### 【先行技術文献】

30

##### 【特許文献】

##### 【0 0 0 2】

【特許文献 1】特開 2011-150653

【特許文献 2】国際公開第 2009-028298 号パンフレット

##### 【背景技術】

##### 【0 0 0 3】

コンピュータシステムにおいては、プロセッサがデータや命令などの情報をメインメモリから取得したり、メインメモリ上の情報を更新したりする。この時、メインメモリのレイテンシ等のアクセス性能がプロセッサの処理性能よりも劣るために、この性能差が全体性能に対するボトルネックとなる。

40

##### 【0 0 0 4】

キャッシュメモリはこのボトルネックを解消するために、記憶階層に新たに設けられるメモリであり、プロセッサとメインメモリとの間に構成される。

通常は、キャッシュメモリには、メインメモリよりも高速な S R A M ( static random access memory ) が用いられる。

プロセッサが、メインメモリからデータを取得したり、メインメモリ上の情報を更新したりするとき、キャッシュメモリにも同じデータが格納される。次に同じデータを取得或いは更新する際には、メインメモリの代わりに高速なキャッシュメモリが入出力を行う。このようにすることで、メインメモリとプロセッサ間の性能差を隠蔽化し、高速なコンピュータシステムを構築する。

50

**【0005】**

ところで、コンピュータシステムを携帯機器へ応用するために、省電力制御技術の必要性が高まっている。最新の高性能プロセッサで使用されている半導体回路技術では、消費電力に対するリーク電流の比率が大きくなっている。そのため、処理すべき命令がない場合等に、プロセッサへの電力供給を遮断し、リーク電流が発生しないようにする、パワーゲーティングと呼ばれる技術が用いられている。

その一方で、キャッシュメモリに用いられるS R A Mは揮発性メモリであり、電力供給が遮断されると格納しているデータが失われてしまう。そこで、上記特許文献1には、キャッシュメモリの構成部分うち、データの入出力等を制御するコントローラのみの電力供給を遮断し、データを格納しているS R A Mセルアレイの電力供給は維持することによって、キャッシュメモリで消費される電力を低減する方法が開示されている。10

**【発明の概要】****【発明が解決しようとする課題】****【0006】**

しかしながら、S R A Mセルアレイ自体にもリーク電流が存在するために、上記の方法では、十分な省電力化が実現できない。そこで、キャッシュメモリを不揮発化し、プロセッサへの電力供給が遮断された際には、キャッシュメモリ全体も電力供給を遮断する構成が考えられる。

このような不揮発性キャッシュメモリに用いられる候補として、強磁性体を用いたM R A M (Magnetic random access memory)、相変化材料を用いたP C R A M (Phase change random access memory)、可変抵抗特性を持った材料を用いたR e R A M (Resistive random access memory)などが注目されている。20

さらに、通常動作時にはS R A Mとして動作するが、電力供給が遮断される際に不揮発性素子にデータを格納することで不揮発化するメモリが上記特許文献2に開示されている。。

**【0007】**

不揮発性メモリにおいては、格納されたデータは電力供給が遮断された場合においても、原則として保持されたままであるが、保持される時間には限度がある。例えば、M R A Mにおいては、熱擾乱によって書き込まれたデータが失われてしまうことがあり得る。このデータが失われるまでの時間をリテンション時間と呼ぶ。30

熱擾乱に対する耐性を高めることでリテンション時間を大きくすることは可能である。しかしながら、リテンション時間は、データを書き込む際に消費される電力や書き込みに要する時間に依存するということを考慮しなければならない。即ち、リテンション時間を長くするためには、大きな電力で書き込みを行わなければならない、或いは、書き込みに長い時間を要する、ということが起きる。これは省電力化や高速化とは相反するものである。そのために、不揮発性キャッシュメモリを用いる際には、リテンション時間を慎重に設計する必要がある。

**【0008】**

不揮発性キャッシュメモリが、電力供給がなされる動作状態にあるとき、データの入出力は頻繁に行われる。そのため、不揮発性キャッシュメモリへの入出力は高速に行わなければならない。また、省電力化のためには、少ないエネルギーで書き込みを行わなければならない。このような要求により、不揮発性キャッシュメモリのリテンション時間は短く設計することが望ましい。40

一方で、不揮発性キャッシュメモリが、電力供給が遮断される待機状態にあるとき、不揮発性キャッシュメモリ内のデータは保持されなければならない。ところが、不揮発性キャッシュメモリのリテンション時間が短く設計されるために、待機期間中にデータが失われる虞がある。これは、メインメモリと不揮発性キャッシュメモリとの間でデータの一貫性が失われることを意味し、コンピュータシステムの信頼性を低下させる。

**【0009】**

そこで本開示では、不揮発性キャッシュメモリであって、電力供給が遮断される待機状50

態にあるときのデータ信頼性を向上させることを目的とする。

【課題を解決するための手段】

【0010】

本開示の不揮発性キャッシュメモリは、キャッシュデータの記憶に用いられる不揮発性メモリ部と、上記不揮発性メモリ部に対してのキャッシュデータの読み出し及び書き込みを制御するとともに、電力供給遮断の準備として、待機状態データを生成して上記不揮発性メモリ部に記憶する待機準備処理を行い、また電力供給再開の際に、上記待機状態データを用いて上記不揮発性メモリ部に記憶されたキャッシュデータの復帰処理を行うキャッシュコントローラとを備える。

【0011】

10

また本開示の不揮発性キャッシュメモリの処理方法は、キャッシュデータの記憶に用いられる不揮発性メモリ部を有する不揮発性キャッシュメモリの処理方法として、電力供給遮断の準備として、待機状態データを生成して上記不揮発性メモリ部上に記憶する待機準備処理を行い、電力供給再開の際に、上記待機状態データを用いて上記不揮発性メモリ部に記憶されたキャッシュデータの復帰処理を行う処理方法である。

【0012】

20

本開示のコンピュータシステムは、プロセッサと、不揮発性キャッシュメモリと、電力制御回路と、メインメモリとを有する。そして上記電力制御回路は、上記プロセッサ及び上記不揮発性キャッシュメモリへの電力供給を制御する。

上記プロセッサは、上記不揮発性キャッシュメモリを介しての上記メインメモリに対するデータの書き込み又は読み出しを実行するとともに、上記電力制御回路が電力供給を遮断する際に、上記不揮発性キャッシュメモリに対して、待機状態に移行することを要求する待機信号を伝達し、また上記電力制御回路が電力供給を再開する際に、上記不揮発性キャッシュメモリに対して電力供給を再開して動作状態に移行することを要求する復帰信号を伝達する。

上記不揮発性キャッシュメモリは、キャッシュデータの記憶に用いられる不揮発性メモリ部と、上記不揮発性メモリ部に対してのキャッシュデータの読み出し及び書き込みを制御するとともに、上記待機信号を受け取ることで、待機状態データを生成して上記不揮発性メモリ部上に記憶する待機準備処理を行い、また上記復帰信号を受け取ることで、上記待機状態データを用いて上記不揮発性メモリ部に記憶されたキャッシュデータの復帰処理を行うキャッシュコントローラとを備えている。

【0013】

30

即ち本開示では、電源遮断となる際には、不揮発性キャッシュメモリでは、待機状態データを生成して不揮発性メモリ部上に記憶する待機準備処理を行う。例えばエラー訂正データ（ECC（Error Correcting Code）データ）やエラー検出データ（EDC（Error Detecting Code）データ）を有する待機状態データを記憶するようにしておく。

そして電源復帰の際には、復帰処理として、待機状態データを用いて不揮発性メモリ部に記憶されたキャッシュデータの復帰処理を行う。例えばエラー訂正処理やエラー検出処理を行う。これにより、電源遮断中のキャッシュデータについて、消失等があったとしても、電源復帰後に正確なキャッシュデータ状態でキャッシュ動作が実行できるようにする。

40

【発明の効果】

【0014】

本開示によれば、不揮発性キャッシュメモリの信頼性を向上させることができる。例えば不揮発性キャッシュメモリが待機中に情報を失ったとしても、その情報の検出や復元を行うことで、メインメモリとの一貫性を保つことができるためである。

【図面の簡単な説明】

【0015】

【図1】本開示の実施の形態のコンピュータシステムのブロック図である。

【図2】第1及び第2の実施の形態によるキャッシュラインの説明図である。

50

【図3】第1の実施の形態の動作状態におけるキャッシュラインのデータ構造図の説明図である。

【図4】第1の実施の形態の待機状態におけるキャッシュラインのデータ構造図の説明図である。

【図5】第1の実施の形態の読み出し動作のフローチャートである。

【図6】第1の実施の形態の書き込み動作のフローチャートである。

【図7】第1の実施の形態の待機状態への移行プロセスを示すフローチャートである。

【図8】第1の実施の形態の動作状態への復帰プロセスを示すフローチャートである。

【図9】第2の実施の形態の待機状態におけるキャッシュラインのデータ構造図の説明図である。

10

【図10】第2の実施の形態の待機状態への移行時におけるキャッシュラインの並び替えの説明図である。

【図11】第2の実施の形態における待機状態への移行プロセスを示すフローチャートである。

【図12】第3の実施の形態によるキャッシュラインの説明図である。

【図13】第3の実施の形態の動作状態と待機状態におけるキャッシュラインのデータ構造図の説明図である。

【図14】第3の実施の形態の待機状態への移行プロセスを示すフローチャートである。

【図15】第4の実施の形態によるキャッシュラインの説明図である。

【図16】第4の実施の形態の待機状態におけるキャッシュラインのデータ構造図の説明図である。

20

【図17】第4の実施の形態における待機状態への移行プロセスを示すフローチャートである。

【図18】第4の実施の形態における動作状態への復帰プロセスを示すフローチャートである。

#### 【発明を実施するための形態】

##### 【0016】

以下、本開示の実施の形態について図面を参照しながら詳細に説明する。ただし、本開示が以下の実施の形態に限定される訳ではない。なお、説明は以下の順序で行う。

<1. 実施の形態の基本構成>

30

<2. 第1の実施の形態>

<3. 第2の実施の形態>

<4. 第3の実施の形態>

<5. 第4の実施の形態>

<6. 変形例>

##### 【0017】

<1. 実施の形態の基本構成>

図1に実施の形態のコンピュータシステムの構成を示す。

図1のコンピュータシステムは、プロセッサ10と、不揮発性キャッシュメモリ11と、メインメモリ12と、電力制御回路13と、内部バス14と、システムバス15とを備える。

40

##### 【0018】

このコンピュータシステムにおいては、プロセッサ10は、メインメモリ12に対してのデータの書き込み／読み出しを、不揮発性キャッシュメモリ11を介して行う。

不揮発性キャッシュメモリ11は、メインメモリ12よりも高速なデータの入出力が可能である。

電力制御回路13は、プロセッサ10及び不揮発性キャッシュメモリ11に対しての電源供給の実行／停止を制御する。

##### 【0019】

50

まず不揮発性キャッシュメモリ 11 について説明する。

図示のように、不揮発性キャッシュメモリ 11 は、データの入出力を管理するキャッシュコントローラ 21 と、データを格納する不揮発性メモリ部 22 とを備える。

プロセッサ 10 によるデータの読み出し及び書き込み要求は、内部バス 14 を通してキャッシュコントローラ 21 に伝えられる。

キャッシュコントローラ 21 は、不揮発性メモリ部 22 上のデータをキャッシュラインと呼ばれる単位で管理する。

#### 【0020】

図 2 は、完全連想マッピング (fully associative mapping) 方式の場合のキャッシュライン CL の構成例を示す図である。各々のキャッシュライン CL 1、CL 2 … CL n は、タグ領域とキャッシュデータ領域とを有する。10

図では、タグ領域に記憶されるタグ情報を「TG」、キャッシュデータ領域に記憶されるキャッシュデータを「CD」として示している。「TG」「CD」には各キャッシュライン CL についての番号を付加している。例えばキャッシュライン CL 1 のタグ情報は「TG 1」、キャッシュデータは「CD 1」、キャッシュライン CL 2 のタグ情報は「TG 2」、キャッシュデータは「CD 2」とする。

図示のように、キャッシュデータ領域には、メインメモリ 12 上のデータのコピーであるキャッシュデータ CD が格納される。

タグ領域には、タグ情報 TG が格納される。

#### 【0021】

タグ情報 TG としては、キャッシュデータのメインメモリ 12 上での格納位置を示すアドレス情報（アドレス上位ビット）と、キャッシュラインの状態を示す制御ビットが含まれている。20

タグ領域に格納するアドレス情報は、すべてのアドレスビットを記憶する必要はない。例えば、キャッシュデータ領域が 32 バイト幅であれば、下位 5 ビットを除くアドレスの上位ビットがタグ領域に格納される。この意味で図 2 では「アドレス上位ビット」と表記している。

#### 【0022】

タグ領域に格納する制御ビットは、ECC ビット Eb と、有効ビット Vb と、ダーティービット Db とで構成される。30

ECC ビット Eb は、不揮発性キャッシュメモリ 11 が待機状態にあるときに用いるビットであり、詳しくは各実施の形態で述べる。

有効ビット Vb は、そのキャッシュラインが有効であることを示すビットである。有効ビット Vb が 1 であれば、キャッシュコントローラ 21 はそのキャッシュライン CL のキャッシュデータ領域からキャッシュデータを読み出すことができる。

ダーティービット Db は、キャッシュデータが読み出し後に変更されたことを示すビットである。ダーティービットが 1 であるキャッシュライン CL は、キャッシュデータがメインメモリ 12 上のデータと異なっている。そのために、そのキャッシュライン CL を解放する際には、メインメモリ 12 との一貫性を保つために、キャッシュデータをメインメモリ 12 の該当するアドレスに書き戻さなければならない。40

#### 【0023】

プロセッサ 10 によるデータの読み出し要求があった場合、キャッシュコントローラ 21 は、該当する有効なキャッシュデータが不揮発性メモリ部 22 上にあるかどうかを検索し、有効なキャッシュデータがある場合には、高速な不揮発性メモリ部 22 からデータを読み出して、内部バス 14 を通してプロセッサ 10 へ送る。

有効なキャッシュデータがない場合には、キャッシュコントローラ 21 は、システムバス 15 を通してメインメモリ 12 から該当するデータを読み出す。そして、内部バス 14 を通してプロセッサ 10 へ送るとともに、不揮発性メモリ部 22 内にそのアドレスと読み出したデータを格納する。

#### 【0024】

50

20

30

40

50

図1に戻り説明を続ける。電力制御回路13は、プロセッサ10及び不揮発性キャッシュメモリ11に入力されるクロック信号及び電力を供給するかどうかを制御する。プロセッサ10は、処理すべき命令がない場合等に消費電力を抑えるために、クロック信号及び電力の供給を停止する待機状態に移行する省電力機能を備える。

【0025】

プロセッサ10が待機状態に移行する場合、プロセッサ10は電力制御回路13に対して待機状態への移行を示す信号PW\_OFFを出力する。

電力制御回路13は、信号PW\_OFFを受け取ると、プロセッサ10に入力される電力PW\_COREとクロック信号CLK\_CORE、及び不揮発性キャッシュメモリ11に入力される電力PW\_CACHEとクロック信号CLK\_CACHEをローレベルにし、クロック信号及び電力の供給を停止する。

【0026】

この際、プロセッサ10内部のレジスタ等に保持されている内部データは、電力の供給が停止されると消失してしまう。そのために、これらのデータは事前に不揮発性メモリに退避しておくか、レジスタ等自体を不揮発化することが望ましい。レジスタ等自体を不揮発化する方法については、特許文献2に開示されている。

【0027】

さて、不揮発性キャッシュメモリ11は、不揮発性メモリ部22にキャッシュデータを格納している。このため、原則的には電力の供給が停止されたとしても、キャッシュデータは保持されたままである。

ただし、待機時間の長さによっては、キャッシュデータが失われる可能性もあり得る。そこで、本実施の形態に係るコンピュータシステムにおいては、不揮発性キャッシュメモリ11は待機状態において、そのキャッシュラインを待機状態用のデータ構造に変換する。例えば待機状態用のキャッシュラインがエラー訂正符号を含むことによって、待機状態中にデータが失われた場合においても、元のデータを復元するか、或いは、失われたデータを無効化する。このようにして、メインメモリとの一貫性を保つことができる。

【0028】

プロセッサ10が待機状態に移行する場合に、プロセッサ10が電力制御回路13に対して待機状態への移行を示す信号PW\_OFFを出力するのに先立って、プロセッサ10は不揮発性キャッシュメモリ11に対して、待機状態に移行することを要求する信号STBYを出力する。

不揮発性キャッシュメモリ11のキャッシュコントローラ21は、STBY信号を受け取ると、待機準備処理を行い、不揮発性メモリ部22内のキャッシュラインを待機状態用のデータ構造に変換する。詳しくは後述するが、この際に、ECCデータやEDCデータを含む待機状態データを生成して上記不揮発性メモリ部上に記憶する処理が行われる。

【0029】

不揮発性キャッシュメモリ11のキャッシュコントローラ21は、待機状態用のデータ構造への変換が完了したら、その旨を通知する信号STBYRをプロセッサ10に対して出力する。

プロセッサ10はSTBYR信号を受け取ることで、不揮発性キャッシュメモリ11が待機状態に移行する準備が完了したと判断し、電力制御回路13に対してPW\_OFF信号を出力する。

その後の動作は先に述べた通りであり、プロセッサ10及び不揮発性キャッシュメモリ11は待機状態に移行する。

【0030】

待機状態を終了し、動作状態に復帰するときは、電力制御回路13に外部から入力される信号INTを用いることができる。

INT信号が入力されると、電力制御回路13は、プロセッサ10に入力される電力PW\_COREとクロック信号CLK\_CORE、及び不揮発性キャッシュメモリ11に入力される電力PW\_CACHEとクロック信号CLK\_CACHEをアクティブにし、電

10

20

30

40

50

力及びクロック信号の供給を再開する。

【0031】

プロセッサ10は、電力及びクロック信号の供給が再開されたことを感知し、不揮発性キャッシュメモリ11に対して通常動作への復帰を要求する信号RESを出力する。

不揮発性キャッシュメモリ11のキャッシュコントローラ21は、RES信号を受け取ると、不揮発性メモリ部22内のキャッシュラインを動作状態用のデータ構造に再変換する。

キャッシュコントローラ21は、動作状態用のデータ構造への変換が完了したら、その旨を通知する信号RESRをプロセッサ10に対して出力する。

プロセッサ10はRESR信号を受け取ることで、不揮発性キャッシュメモリ11が動作状態に移行する準備が完了したと判断し、通常の動作を再開する。 10

【0032】

なお、不揮発性キャッシュメモリ11はRES信号を受け取らずに、クロック信号と電力の供給再開を検出して、不揮発性メモリ部22内のキャッシュラインを動作状態用のデータ構造に再変換してもよい。

【0033】

以上に述べたように、不揮発性メモリ部22内のキャッシュラインは、動作状態と待機状態において異なったデータ構造を持つ。

即ち、不揮発性キャッシュメモリ11では、電力供給遮断の準備として、ECCデータ、EDCデータ等を含む待機状態データを生成して不揮発性メモリ部22に記憶する待機準備処理を行う。これにより、電源遮断後の待機中は、不揮発性メモリ部22は、待機状態データを含めたデータ構造となっている。 20

また電力供給再開の際には、記憶されている待機状態データを用いて不揮発性メモリ部22に記憶されたキャッシュデータの復帰処理を行う。後述するが、例えばエラー訂正処理やエラー検出処理、さらには他の必要な処理が行われる。

【0034】

このようにすることで、例えば待機中に情報を失ったとしても、その情報の検出や復元を行うことで、メインメモリ12との一貫性を保つことができるようになる。つまり少なくとも有効とされているキャッシュデータは、メインメモリ12のデータと一致している状態となっているようになる。 30

【0035】

即ち以下説明する第1～第4の実施の形態は、キャッシュデータの記憶に用いられる不揮発性メモリ部22と、不揮発性メモリ部22に対してのキャッシュデータの読み出し及び書き込みを制御するとともに、電力供給遮断の準備として、待機状態データを生成して不揮発性メモリ部22に記憶する待機準備処理を行い、また電力供給再開の際に、待機状態データを用いて不揮発性メモリ部22に記憶されたキャッシュデータの復帰処理を行うキャッシュコントローラ21を備えている。

【0036】

そして第1～第3の実施の形態では、キャッシュコントローラ21は、待機準備処理として、少なくとも不揮発性メモリ部22上に記憶されているキャッシュデータについてのエラー訂正を可能とするエラー訂正データ(ECCデータ)を含む待機状態データを生成して不揮発性メモリ部22上に記憶させ、復帰処理として、待機状態データにおけるエラー訂正データを用いて不揮発性メモリ部22に記憶されたキャッシュデータのエラー訂正処理を行うものである。 40

【0037】

また第4の実施の形態では、キャッシュコントローラ21は、待機準備処理として、少なくとも不揮発性メモリ部22上に記憶されているキャッシュデータについてのエラー検出を可能とするエラー検出データ(EDCデータ)を含む待機状態データを生成して不揮発性メモリ部22上に記憶させ、復帰処理として、待機状態データにおけるエラー検出データを用いて不揮発性メモリ部22に記憶されたキャッシュデータのエラー検出処理を行 50

うものである。

【0038】

<2. 第1の実施の形態>

続いて第1の実施の形態を説明する。

第1の実施の形態において、キャッシュラインCLは、上記の図2で示した構成とされる。そして、動作状態(通電中の通常の状態)と待機状態(電源遮断中の状態)とにおいて、キャッシュラインのデータ構造は図3、図4に示した構成とされる。

【0039】

図3は動作状態のキャッシュラインCLを示している。

10

この場合、すべてのキャッシュラインCL(CL1~CLm)がキャッシュラインモードとなる。

キャッシュラインモードでは、制御ビットのECCビットEbが「0」に設定され、当該キャッシュラインがキャッシュラインモードであることを示す。

有効ビットVbとダーティービットDbは、当該キャッシュラインの状態に応じて設定される。図の「X」は「1」又は「0」の意味である。

【0040】

アドレス上位ビットには、キャッシュデータCDの参照元であるメインメモリ12のアドレス(メインメモリアドレス)の必要な上位ビットが格納され、キャッシュデータCDには、そのメインメモリアドレスに対応したデータが格納される。

20

【0041】

次に、動作状態から待機状態に移行する際には、待機準備処理が行われることで、キャッシュラインCL1~CLmは、キャッシュラインモードにあるキャッシュラインと、ECCモードにあるキャッシュラインとに分割される。

ECCモードとは、待機状態データを記憶したキャッシュラインとなっている状態である。

【0042】

待機状態においてキャッシュラインモードにあるキャッシュライン(例えばキャッシュラインCL1, CL2...)は、動作状態のキャッシュラインCLと同じデータ構造である。

30

【0043】

ECCモードにあるキャッシュライン(例えばキャッシュラインCLx)は、ECCビットEbが「1」に設定され、当該キャッシュラインがECCモードであること(つまり待機状態データであるECCデータを記憶したキャッシュラインであること)を示す。

さらに、制御ビットの残りの有効ビットVbとダーティービットDbは「0」に設定される。

そして、アドレス上位ビットを格納する領域とキャッシュデータ領域は併せて新たにECC領域とされる。ECC領域はキャッシュアドレスCAを格納するキャッシュアドレス領域とECCデータECを格納するECCデータ領域とで構成される。

【0044】

40

このように、待機状態において2つのモードを利用したデータの格納方法について、以下に詳細に述べる。

前述のように、不揮発性キャッシュメモリにおいては、待機期間中に格納していたデータが失われる虞がある。そこで、本実施の形態の不揮発性キャッシュメモリ11においては、エラー訂正符号(ECC)を用いて失われたデータを検出し、必要であればエラーを訂正する。

【0045】

待機状態においてキャッシュラインモードにあるキャッシュラインCLの中で、ダーティービットDb、アドレス上位ビット、キャッシュデータCDをECC対象データと呼ぶ。そして、ECC対象データをエラー訂正符号化してECCデータを得る。

50

ここで、ECC対象データとECCデータを併せたものがエラー訂正符号であり、生成されたエラー訂正符号のうち、ECC対象データ以外の部分（ECCパリティ）をECCデータとする。

【0046】

ECCデータは、ECCモードにあるキャッシュラインCLのECCデータ領域に格納される。

その際、ECCデータがどのキャッシュラインCLのものであるかを知るために、キャッシュアドレス領域にキャッシュラインのアドレス（キャッシュアドレスCA）が格納される。

【0047】

図4に示す例の場合、キャッシュラインCL1のECC対象データに対してのECCデータEC1、及びキャッシュラインCL1を示すキャッシュアドレスCA1は、キャッシュラインCL1とは別のキャッシュラインCLxに格納される。

またキャッシュラインCL2のECC対象データに対してのECCデータEC2、及びキャッシュラインCL2を示すキャッシュアドレスCA2も、キャッシュラインCLxに格納される。

この場合、キャッシュアドレスCAとECCデータECの合計ビット数は、ECC領域のビット数よりも小さいものとしている。

そのために、ひとつのECC領域には複数のキャッシュアドレスCAとECCデータECのペアを格納することができる。

図4ではその一例として2つのペアを格納する場合が図示されているものである。しかし、これに限定されることなく、1つのペアや3つ以上のペアを格納してもよい。

【0048】

次にキャッシュラインモードとECCモードの割り当て方法について述べる。

キャッシュラインモードは常に、有効ビットが1である有効なキャッシュラインCLが対象となる。これは有効ビットが0、即ち無効なキャッシュラインを待機中に保持する必要がないためである。

【0049】

ECCモードは、基本的には、有効ビットが0である無効なキャッシュラインCLに割り当てられる。

但し、ECCモードに割り当てるべきキャッシュラインCLの数が、無効なキャッシュラインCLの数よりも大きい場合は、全ての有効なキャッシュラインCLに対応するECCデータECを格納する領域が不足することになる。この場合は、有効なキャッシュラインCLを必要なだけ選択して、ECCモードに割り当てる。この場合、ECCモードに割り当てる有効なキャッシュラインCLについては、その有効ビットVbを「0」にしてそのキャッシュラインCLを無効化する。

有効なキャッシュラインのうちどのキャッシュラインを選択するかは、通常のキャッシュメモリで用いられるさまざまなライン入れ替え方式を用いることができる。

例えば、ラウンドロビン方式、LRU (least recently used) 方式、ランダム方式等が挙げられる。

以上のような選択によって、キャッシュラインモードのキャッシュラインCLのECCビットEbは「0」、有効ビットVbは「1」となり、ECCモードのキャッシュラインCLのECCビットEbは「1」、有効ビットVbは「0」となる。

【0050】

ここで、待機中のエラーはECCビット、有効ビットにも起こり得ることに注意する。待機中にキャッシュラインモードのキャッシュラインCLのECCビットEbが「1」に変化したとすると、ECCビットEbと有効ビットVbがともに「1」となる。

このようなキャッシュラインCLはキャッシュラインモードでもECCモードでもないために無効とされる。

同様に、キャッシュラインモードのキャッシュラインCLの有効ビットVbが「0」に

10

20

30

40

50

なった場合、ECCモードのキャッシュラインCLのECCビットEbが「0」になった場合、及びECCモードのキャッシュラインCLの有効ビットVbが「1」になった場合も無効とされる。

このようにすることで、エラー訂正の保護を受けないECCビットおよび有効ビットでエラーが起きた場合でも、誤動作をすることないようにできる。

#### 【0051】

電源供給の再開により、待機状態から動作状態への移行する際には、キャッシュラインモードにあるキャッシュラインCLに対して、ECCモードのキャッシュラインCLの中から一致するキャッシュアドレスCAを持つキャッシュラインCLを走査し、ECCデータECを取得する。

そして、ECC対象データとECCデータとを併せてエラー訂正符号を構成し、エラー訂正を行う。必要であれば正しいデータをECC対象データに書き込む。場合によっては、ECCビットEbや有効ビットVbがエラーを起こしたために、一致するキャッシュアドレスCAを持つECCモードのキャッシュラインCLが見つからない場合があるが、そのような場合にはエラー訂正が行えないために、当該キャッシュラインCLは無効化する。

#### 【0052】

以下、本実施の形態においてキャッシュコントローラ21によって実行される処理を説明する。

まず図5は、通常の動作状態において、プロセッサ10から読み出し要求を受けた場合の処理例を示している。

#### 【0053】

ステップF101でキャッシュコントローラ21がプロセッサ10からアドレスAに対する読み出し要求を受けると、キャッシュコントローラ21は、ステップF102で、不揮発性メモリ部22の中にアドレスAに対応する有効なキャッシュラインCLがあるかどうか判断する。つまり有効ビットVbが「1」であり、メインメモリアドレスとしてアドレスAが記憶されているキャッシュラインが有るか否かを判断する。

#### 【0054】

有効なキャッシュラインがある場合には、キャッシュコントローラ21はステップF103に進み、当該キャッシュラインCLからキャッシュデータCDを取得し、プロセッサ10に送信する。

つまり、この場合は、いわゆるキャッシュヒット転送として、メインメモリ12にアクセスすることなくアドレスAのデータがプロセッサ10に転送される。

#### 【0055】

一方、アドレスAに対応する有効なキャッシュラインCLがない場合は、キャッシュコントローラ21はステップF104に進み、メインメモリ12からアドレスAのデータDを読み出す。

そしてステップF105でキャッシュコントローラ21は、このメインメモリ12から読み出したデータDを不揮発性メモリ部22に格納するために、無効なキャッシュラインCLがあるかどうか判断する。

#### 【0056】

無効なキャッシュラインCLがある場合、キャッシュコントローラ21はステップF106で、その中からキャッシュラインCLを一つ選ぶ。

そしてステップF110で、選択したキャッシュラインCLに、メインメモリ12から読み出したデータDとアドレスAを新規登録する。つまり選択したキャッシュラインCLにおいて、データDをキャッシュデータCDとして格納し、メインメモリアドレスとしてアドレスAを格納する。

また当該キャッシュラインCLは、新規な、有効キャッシュラインとなるため、有効ビットVbを「1」に、ダーティービットDbを「0」に設定する。

#### 【0057】

10

20

30

40

50

そしてキャッシュコントローラ21はステップF111で、データDをプロセッサ10に送信する。

つまりこの場合は、キャッシュヒットしなかったため、メインメモリ12から読み出したデータが、プロセッサ10に転送されることになる。また、メインメモリ12から読み出されたデータDは、キャッシュデータCDとして或るキャッシュラインCLに格納される。

#### 【0058】

一方、ステップF105で無効なキャッシュラインCLが存在しないと判定された場合は、キャッシュコントローラ21はステップF107に進み、有効なキャッシュラインCLを一つ選ぶ。10

そしてステップF108では、選択したキャッシュラインCLのダーティービットDbが「0」であるか否かを確認する。

もし「0」であれば、そのままステップF110, F111で、キャッシュデータ新規登録と、プロセッサ10への送信の処理を行う。

この場合、或る有効なキャッシュラインCLに登録されていたキャッシュデータCDを消失させて、新たに今回メインメモリ12から読み出したデータDをキャッシュデータCDとして記憶する動作となる。

#### 【0059】

またステップF108で、選択したキャッシュラインCLのダーティービットDbが「1」であった場合は、キャッシュコントローラ21はステップF109の処理を行う。この場合、キャッシュデータCDは、メインメモリ12におけるアドレスAのデータとは異なるデータになっている。そこで、当該キャッシュラインCLからキャッシュデータCD及びメインメモリアドレスを読み出し、メインメモリ12におけるアドレスAに書き戻す処理を行う。20

これによって、メインメモリ12におけるアドレスAのデータに、それまでキャッシュデータCD上で行われていた更新を反映させる。

その後キャッシュコントローラ21はステップF110, F111の処理を行う。

#### 【0060】

次に図6は、通常の動作状態において、プロセッサ10から書き込み要求を受けた場合の処理例を示している。30

#### 【0061】

ステップF201で、キャッシュコントローラ21はプロセッサ10からアドレスAに対するデータDの書き込み要求を受ける。

まずキャッシュコントローラ21はステップF202で、不揮発性メモリ部22の中にアドレスAに対応する有効なキャッシュラインCLがあるかどうか判断する。

#### 【0062】

有効なキャッシュラインCLがある場合には、当該キャッシュラインCLのキャッシュデータCDを、今回プロセッサ10から供給されたデータDに書き替える。つまりキャッシュデータCDを更新する。また、これに伴いダーティービットDbを「1」にする。有効ビットVbも「1」とする。40

#### 【0063】

そしてステップF211で、メインメモリ12におけるアドレスAに、今回のデータDを書き込む処理を行う。

但し、キャッシュ上で更新する度にメインメモリ12上での更新を必ず行う必要は必ずしもない。例えば後の或る時点でメインメモリ12上でのデータ更新を行うのであれば、この段階でステップF211の書き込みを行う必要はない。

また、ステップF211で、キャッシュデータCDの更新データをメインメモリ12上に反映させた場合は、キャッシュラインCLのダーティービットDbを「0」に戻すようになる。

#### 【0064】

ステップF202で、不揮発性メモリ部22の中にアドレスAに対応する有効なキャッシュラインCLがないと判断された場合は、キャッシュコントローラ21はステップF205に進み、今回のデータDを不揮発性メモリ部22に格納するために、無効なキャッシュラインCLがあるかどうか判断する。

【0065】

無効なキャッシュラインCLがある場合、キャッシュコントローラ21はステップF206で、その中からキャッシュラインCLを一つ選ぶ。

そしてステップF210で、選択したキャッシュラインCLにアドレスAとデータDを新規登録する。つまり選択したキャッシュラインCLにおいて、データDをキャッシュデータCDとして格納し、メインメモリアドレスとしてアドレスAを格納する。

10

また当該キャッシュラインCLは、新規な、有効キャッシュラインとなるため、有効ビットVbを「1」に設定する。またこの時点では、新たなデータD(キャッシュデータCD)はメインメモリ12のアドレスAのデータと一致していないため、ダーティービットDbを「1」に設定する。

【0066】

そしてステップF211でメインメモリ12のアドレスAへのデータDの書き込みを実行する。その場合は、ダーティービットを0に戻す。

或いは、ステップF211は、この時点では実行しなくてもよい。

【0067】

ステップF205で無効なキャッシュラインCLがないと判断した場合、キャッシュコントローラ21はステップF207で有効なキャッシュラインCLを一つ選ぶ。

20

そしてステップF208で、選んだ有効なキャッシュラインCLのダーティービットDbが「0」かどうか判断する。

もし「0」であれば、そのままステップF210で、キャッシュデータ新規登録を行う。またステップF211でメインメモリ12への書き込みを行う(或いはこの時点では行わないとしてもよい)。

この場合、或る有効なキャッシュラインCLに登録されていたキャッシュデータCDを消失させて、新たに今回の書き込みデータDをキャッシュデータCDとして記憶する動作となる。

【0068】

30

またステップF208で、選択したキャッシュラインCLのダーティービットDbが「1」であった場合は、キャッシュコントローラ21はステップF209の処理を行う。この場合、キャッシュデータCDは、メインメモリ12におけるアドレスAのデータとは異なるデータになっている。そこで、当該キャッシュラインCLからキャッシュデータCD及びメインメモリアドレスを読み出し、メインメモリ12におけるアドレスAに書き戻す処理を行う。

これによって、メインメモリ12におけるアドレスAのデータに、それまでキャッシュデータCD上で行われていた更新を反映させる。

その後キャッシュコントローラ21はステップF210(及びF211)の処理を行う。

40

【0069】

次に図7に、通常の動作状態から待機状態への移行プロセスにおいて、プロセッサ10から待機信号を受けた場合のキャッシュコントローラ21の処理例、つまり待機準備処理の例を示す。

【0070】

ステップF301で、キャッシュコントローラ21はプロセッサ10から待機信号を受ける。

まずキャッシュコントローラ21はステップF302で、不揮発性メモリ部22における有効なキャッシュラインCLの数をカウントする。即ち有効ビットVbが「1」のキャッシュライン数である。これを有効キャッシュライン数Vとしてカウントする。

50

## 【0071】

次にキャッシュコントローラ21はステップF303で、ECCモードに必要なキャッシュラインCLの数を求める。必要キャッシュライン数Eとする。

例えば図4の例のように、1つのECCモードのキャッシュラインCLに2つのキャッシュラインCLについてのECCデータEC及びキャッシュアドレスCAを格納するデータ構造とする場合、 $E = V / 2$ となる。

## 【0072】

次にステップF304でキャッシュコントローラ21は、無効なキャッシュラインCLの数Iが、必要キャッシュライン数Eより大きいか否かを確認する。

無効キャッシュライン数Iは、 $I = (\text{全キャッシュライン数}) - V$ である。

10

## 【0073】

I = Eであれば、無効なキャッシュラインCLを用いてECCモードのキャッシュラインを形成できる。

そこで、キャッシュコントローラ21はステップF309に進み、無効なキャッシュラインCLをE個選ぶ。

選択したE個のキャッシュラインCLについては、全てECCビットEbを「1」に、ダーティービットDbを「0」にする。

## 【0074】

ステップF310でキャッシュコントローラ21は、有効なキャッシュラインCLのそれぞれについて、ECCデータECを計算する。

20

例えば有効なキャッシュラインCLとして、キャッシュラインCL1、CL2、CL5、CL6、CL7、CL10が存在していたとした場合、それぞれについてのECCデータEC1, EC2, EC5, EC6, EC7, EC10を算出する。

## 【0075】

そしてステップF311でキャッシュコントローラ21は、キャッシュアドレスCAとECCデータECを、ステップF309で選択したキャッシュラインCLに書き込む。

例えばキャッシュラインCL15, CL16, CL17がECCモード用に選択されていたとする。その場合例えば、キャッシュラインCL1、CL2に対応するECCデータEC1, EC2及びキャッシュアドレスCA1, CA2を、キャッシュラインCL15に書き込む。またキャッシュラインCL5、CL6に対応するECCデータEC5, EC6及びキャッシュアドレスCA5, CA6を、キャッシュラインCL16に書き込む。またキャッシュラインCL7、CL10に対応するECCデータEC7, EC10及びキャッシュアドレスCA7, CA10を、キャッシュラインCL17に書き込む。

30

これによって、キャッシュラインCL15, CL16, CL17がECCモードのデータ構造のキャッシュラインCLとされたこととなる。

## 【0076】

以上で不揮発性メモリ部22での待機準備が完了したことになり、キャッシュコントローラ21はステップF312でプロセッサ10に対して待機完了信号を送って、一連の処理を終える。

## 【0077】

40

ところで、ステップF304で無効キャッシュライン数Iが必要キャッシュライン数Eより小さい場合は、無効なキャッシュラインCLを用いるだけではキャッシュライン数が足りないこととなる。

そこでその場合は、キャッシュコントローラ21はステップF305に進む。

まずステップF305で、有効なキャッシュラインCLのうちで( $E - I$ )個のキャッシュラインCLを選択する。

そして、ステップF306で、選択した( $E - I$ )個のキャッシュラインCLのそれぞれについてダーティービットDbが「0」であるか否かを確認する。

## 【0078】

もし( $E - I$ )個のキャッシュラインCLの全てにおいて、ダーティービットDbが「

50

0」であれば、そのままステップF 308に進み、選択したキャッシュラインCLの有効ビットVbを「0」にする。つまり無効化する。キャッシュラインCLは、あくまで図1のコンピュータシステムにおいてキャッシュを記憶する領域であるため、有効なキャッシュラインCLを無効化すること(キャッシュデータCDを破棄すること)は問題とはならない。単にキャッシュの一部が破棄されるだけだからである。

そしてステップF 302, F 303の処理を再度行い、ステップF 304で無効キャッシュライン数Iが必要キャッシュライン数Eより大きくなつていれば、ステップF 309～F 312の処理を行うことになる。

#### 【0079】

またステップF 306で、選択した(E-I)個のキャッシュラインCLのうちで、ダーティービットDbが「1」のキャッシュラインCLが存在したと判定した場合は、キャッシュコントローラ21はステップF 307の処理を行う。この場合、選択したキャッシュラインCLのうちのダーティービットDbが「1」のキャッシュラインCLのキャッシュデータCDは、メインメモリ12におけるデータとは異なるデータになっている。そこで、当該キャッシュラインCLからキャッシュデータCD及びメインメモアドレスを読み出し、メインメモリ12に書き戻す処理を行う。

これによって、メインメモリ12におけるデータに、それまでキャッシュデータCD上で行われていた更新を反映させる。

その後キャッシュコントローラ21はステップF 308の処理を経て、ステップF 302, F 303の処理を再度行い、ステップF 304で無効キャッシュライン数Iが必要キャッシュライン数Eより大きくなつていれば、ステップF 309～F 312の処理を行うことになる。

#### 【0080】

なお、ステップF 305で選択する有効なキャッシュラインCLの数(つまり無効化するキャッシュラインCLの数)は、1個、又は2個という固定値とし、処理を繰り返すことで、或る時点で、ステップF 304で肯定結果が得られるようにしてもよい。

#### 【0081】

次に図8に、待機状態から動作状態への移行プロセスにおいて、プロセッサ10から復帰信号を受けた場合のキャッシュコントローラ21の処理例、つまり復帰処理の例を示す。

ステップF 401で、キャッシュコントローラ21はプロセッサ10から復帰信号を受ける。

#### 【0082】

キャッシュコントローラ21はステップF 402で、不揮発性メモリ部22においてキャッシュラインモードとされているキャッシュラインCLを一つ選ぶ。

ステップF 403でキャッシュコントローラ21は、選択したキャッシュラインCLのキャッシュアドレスCAが記憶されているECCモードのキャッシュラインを選ぶ。

ステップF 404でキャッシュコントローラ21は、ステップF 402で選択したキャッシュラインモードのキャッシュラインCLに対応するECCデータECを、ステップF 403で選択したECCモードのキャッシュラインCLから読み出し、キャッシュラインモードのキャッシュラインCLのエラー訂正処理を行う。

#### 【0083】

以上の処理は、例えば図4の状態に沿って述べると次のようになる。

ステップF 402でキャッシュラインCL1を選択し、ステップF 403でキャッシュラインCL1を示すキャッシュアドレスCA1を記憶しているECCモードのキャッシュラインCLxを選択する。そしてステップF 404で、キャッシュラインCLxに記憶されているECCデータEC1を読み出し、このECCデータEC1を用いて、キャッシュラインCL1のECC対象データのエラー訂正処理を行うこととなる。

#### 【0084】

ステップF 405では、キャッシュコントローラ21は、全てのキャッシュラインモー

10

20

30

40

50

ドのキャッシュラインについて、ステップF402～F404によるエラー訂正処理が完了したか否かを確認する。完了していなければステップF402に戻って、残りのキャッシュラインモードのキャッシュラインCLについてステップF402～F404の処理を行う。例えば図4の例で述べると、次にキャッシュラインCL2について、同様にエラー訂正処理を行うことになる。

【0085】

全てのキャッシュラインモードのキャッシュラインについて、ステップF402～F404によるエラー訂正処理が完了したら、キャッシュコントローラ21はステップF406に進み、全てのECCモードのキャッシュラインCLについては、ECCビットEbを「0」として、単なる無効なキャッシュラインCLにする。

そしてステップF407でキャッシュコントローラ21はプロセッサ10に復帰完了信号を送って、一連の復帰処理を終える。

【0086】

以上に述べたように、第1の実施の形態は、不揮発性メモリ部22には、キャッシュデータCDの記憶に用いられるキャッシュラインCLが複数設定されている。そしてキャッシュコントローラ21は、待機準備処理として、少なくとも一部のキャッシュラインCLについて、そのキャッシュラインCLに記憶されているキャッシュデータCDのエラー訂正可能なエラー訂正データ(ECCデータEC)を含む待機状態データを生成し、該待機状態データを他のキャッシュライン(ECCモードのキャッシュラインCL)に記憶させ、かつ当該他のキャッシュラインを、待機状態データを記憶したキャッシュラインであることを示す状態とする処理(ECCビットEbを「1」とする処理)を行う。

そしてエラー訂正データを含む待機状態データには、さらに当該エラー訂正データでエラー訂正を行うキャッシュデータが記憶されたキャッシュラインのアドレス情報CAが含まれている。

【0087】

このような第1の実施の形態においては、待機状態に移行する前に有効なキャッシュラインのECCデータを計算し、ECCデータを無効なキャッシュラインに格納することによって、待機中にデータが失われた場合においても、復帰動作の際にデータの訂正が可能となる。

【0088】

なお、ステップF406の処理は行わない例も考えられる。つまり、ある程度の期間、ECCモードのキャッシュラインCLをそのまま残しておいてもよい。例えば通常のキャッシュ動作においてキャッシュラインCLが足りなくなるまで、ECCモードのキャッシュラインCLを残しておき、足りなくなったら、このキャッシュラインCLを使用するようにする。

このようにECCモードのキャッシュラインCLを残しておくことで、例えば電源瞬断時など、待機準備処理ができなかったような場合に、その復帰時にエラー訂正できるキャッシュラインCLが存在する可能性を得ることができる。

【0089】

<3. 第2の実施の形態>

続いて第2の実施の形態を説明する。なお、第1の実施の形態と重複する部分については説明を省略する。

第2の実施の形態は、キャッシュコントローラ21が、待機状態データであるECCデータECを、その待機状態データが対象とするキャッシュデータCDを記憶したキャッシュラインCLに対応する領域に記憶させる例である。

第2の実施の形態において、第1の実施の形態と同じく、キャッシュラインCLは図2で示した構成とされる。そして、動作状態と待機状態とにおいて、キャッシュラインのデータ構造は図3、図9に示した構成とされる。

【0090】

10

20

30

40

50

動作状態において、すべてのキャッシュラインCLがキャッシュラインモードとなることは図3のとおりであり、つまり第1の実施の形態と同様である。

また動作状態から待機状態に移行する際に、キャッシュラインCLは、キャッシュラインモードにあるキャッシュラインCLとECCモードにあるキャッシュラインCLとに分割される点も、第1の実施の形態と同じである。

#### 【0091】

この第2の実施の形態では、図9に示すように、ECCモードのキャッシュラインCLでは、制御ビット以外の領域はECCデータ領域とされる。つまり図4と比較してわかるようにキャッシュアドレス領域は設けられない。

なおECCデータ領域の働きや制御ビットの設定方法は、第1の実施の形態と同じである。

#### 【0092】

このように第2の実施の形態においては、ECCモードにあるキャッシュラインCL内にキャッシュアドレス領域がない。このため、ECCデータ領域が、どのキャッシュラインCLに対応づけられているかを以下のように決定する。

#### 【0093】

図10は待機状態への移行の際のキャッシュラインの並べ替えを示した図である。

動作状態においては、有効なキャッシュラインCL（有効ビットVb = 1）と無効なキャッシュラインCL（有効ビットVb = 0）が混在している。

待機状態に移行する際には、有効なキャッシュラインCLをキャッシュアドレス順に整列させる（キャッシュラインCL1からキャッシュラインCL8）。

この有効なキャッシュラインを整列させる領域（キャッシュラインCL1～CL8）をキャッシュラインモード領域とする。

そして、無効なキャッシュラインは、有効なキャッシュラインの後に配置する（キャッシュラインCL9～CL12）。この領域をECCモード領域とする。

#### 【0094】

ここで、キャッシュラインCL1に対応したECCデータEC1はキャッシュラインCL9の前半に、キャッシュラインCL2に対応したECCデータEC2はキャッシュラインCL9の後半に、キャッシュラインCL3に対応したECCデータEC3はキャッシュラインCL10の前半に・・・、というように、キャッシュラインCLの位置とECCデータECの位置を一一に対応させる。

このようにすると、ECCモードにあるキャッシュラインCL内にキャッシュアドレスCAを格納する必要がない。

#### 【0095】

なお、この例では一つのキャッシュラインCLに2つのECCデータECを格納しているが、これに限定されることなく、1つ或いは3つ以上のECCデータECを格納してもよい。

キャッシュラインCLの総数をT、一つのキャッシュラインCLに格納できるECCデータECの個数をNとすると、キャッシュラインモード領域のキャッシュライン数Cは、 $C + C / N = T$ を満たす。即ち、 $C = NT / (N + 1)$ である。

図10の例で言うと、 $N = 2$ 、 $T = 12$ であるから、 $C = 8$ となる。このとき、キャッシュラインモード領域はキャッシュラインCL1からキャッシュラインCL8に、ECCモード領域はキャッシュラインCL9からキャッシュラインCL12に、それぞれ割り当てる。

#### 【0096】

図11は、第2の実施の形態における待機準備処理のフローチャートである。

ステップF501で、キャッシュコントローラ21はプロセッサ10から待機信号を受ける。

するとキャッシュコントローラ21はステップF502で、不揮発性メモリ部22における有効なキャッシュラインCLの数をカウントする。即ち有効ビットVbが「1」のキ

10

20

30

40

50

ヤッシュライン数である。これを有効キャッシュライン数  $V$  としてカウントする。

【0097】

次にキャッシュコントローラ 21 はステップ F503 で、上述のキャッシュラインモード領域のキャッシュライン数  $C$  が、有効キャッシュライン数  $V$  よりも大きいかどうか判断する。

$C > V$  であれば、キャッシュコントローラ 21 はステップ F508 に進み、有効なキャッシュライン  $C_L$  をキャッシュラインモード領域の先頭から順に整列させる。

【0098】

続いてキャッシュコントローラ 21 はステップ F509 で ECC モードに必要なキャッシュライン数  $E$  を求める。

そしてステップ F510 で、ECC モード領域の先頭から  $E$  個のキャッシュライン  $C_L$  の ECC ビット  $E_b$  を「1」に、ダーティービット  $D_b$  を「0」にする。つまり ECC モードのキャッシュラインを形成する。

【0099】

ステップ F511 でキャッシュコントローラ 21 は、有効なキャッシュライン  $C_L$  のそれについて、ECC データ  $E_C$  を計算する。

そしてステップ F512 でキャッシュコントローラ 21 は、ECC データ  $E_C$  を、それぞれ図 10 で説明したように、対応するキャッシュライン  $C_L$  に書き込む。

これによって、ECC データ領域に必要な ECC モードのキャッシュライン  $C_L$  が形成されたこととなる。

【0100】

以上で不揮発性メモリ部 22 での待機準備が完了したことになり、キャッシュコントローラ 21 はステップ F513 でプロセッサ 10 に対して待機完了信号を送って、一連の処理を終える。

【0101】

一方、ステップ F503 でキャッシュラインモード領域のキャッシュライン数  $C$  が、有効キャッシュライン数  $V$  よりも小さい場合は、キャッシュコントローラ 21 はステップ F504 に進む。

まずステップ F504 で、有効なキャッシュライン  $C_L$  のうちで  $(V - C)$  個のキャッシュライン  $C_L$  を選択する。

そして、ステップ F505 で、選択した  $(V - C)$  個のキャッシュライン  $C_L$  のそれについてダーティービット  $D_b$  が「0」であるか否かを確認する。

【0102】

もし  $(V - C)$  個のキャッシュライン  $C_L$  の全てにおいて、ダーティービット  $D_b$  が「0」であれば、そのままステップ F507 に進み、選択したキャッシュライン  $C_L$  の有効ビット  $V_b$  を「0」にする。つまり無効化する。

そしてステップ F502 の処理を再度行い、ステップ F503 で  $C = V$  となつていれば、ステップ F508 ~ F513 の処理を行うことになる。

【0103】

また、ステップ F504 で、選択した  $(E - I)$  個のキャッシュライン  $C_L$  のうちで、ダーティービット  $D_b$  が「1」のキャッシュライン  $C_L$  が存在したと判定した場合は、キャッシュコントローラ 21 はステップ F506 の処理を行う。この場合、選択したキャッシュライン  $C_L$  のうちのダーティービット  $D_b$  が「1」のキャッシュライン  $C_L$  のキャッシュデータ  $C_D$  は、メインメモリ 12 上のデータとは異なるデータになっている。そこで、当該キャッシュライン  $C_L$  からキャッシュデータ  $C_D$  及びメインメモアドレスを読み出し、メインメモリ 12 に書き戻す処理を行う。

これによって、メインメモリ 12 上のデータに、それまでキャッシュデータ  $C_D$  上で行われていた更新を反映させる。

その後キャッシュコントローラ 21 はステップ F507 の処理を経て、ステップ F502 の処理を再度行い、ステップ F503 で  $C = V$  となつていれば、ステップ F508 ~ F

10

20

30

40

50

513の処理を行うことになる。

【0104】

第2の実施の形態における復帰処理については、基本的に図8と同様である。異なる点は図8のステップF403において、対応するECCモードのキャッシュラインCLの選択が、キャッシュアドレスCAに基づくのではなく、図10で説明したように一意的に決定される点である。

【0105】

以上の第2の実施の形態でも第1の実施の形態と同様の効果が得られる。加えて、ECCモードのキャッシュラインCLにおいては、キャッシュアドレスCAを記憶する必要はないため、ECCデータECの記憶に有効できる。例えば1つのECCモードのキャッシュラインCLで、より多数のキャッシュラインモードのキャッシュラインCLに対応できたり、或いはECCパリティの冗長度を上げ、訂正能力を上げるといった点で有利となる。

10

【0106】

<4. 第3の実施の形態>

続いて第3の実施の形態を説明する。なお、第1の実施の形態と重複する部分については説明を省略する。

第3の実施の形態において、キャッシュラインCLは図12で示した構成とされる。そして、動作状態と待機状態において、キャッシュラインのデータ構造は図13に示した構成とされる。

20

【0107】

図12に示すように、キャッシュラインCLは、第1の実施の形態の構成に加えて、ECCデータ領域を含む。

図13に示すように、動作状態においては、データの入出力はタグ領域とキャッシュデータ領域のみを対象に行われ、ECCデータ領域は用いられない。

そして待機状態への移行の際には、すべての有効なキャッシュラインはECCモードとなり、ECCビットEbが「1」に設定される。即ち、ECCビットEbと有効ビットVbはともに「1」となる。

【0108】

30

エラー訂正の対象となるECC対象データは、データビットDb、アドレス上位ビット(メインメモリアドレス)、キャッシュデータCDである。

このECC対象データを元にしてECCデータが計算され、同じキャッシュラインのECCデータ領域に格納される。

ECC対象データとECCデータが同じキャッシュラインにあるために、キャッシュアドレスを格納する必要はない。

【0109】

動作状態への復帰の際には、ECCビットEbと有効ビットVbがともに「1」であるキャッシュラインに対して、ECC対象データとECCデータとを併せてエラー訂正符号を構成し、エラー訂正を行う。必要であれば、正しいデータをECC対象データに書き込む。仮に、待機中にECCビットEbと有効ビットVbのいずれかが「0」に変化したキャッシュラインCLは、無効とされる。

40

【0110】

このようにすると、各キャッシュラインCLに対して、そのECCデータECがどこに存在するかを走査する必要がなくなる。

また、各キャッシュラインCLがECCデータ領域を備えているために、有効なキャッシュラインであっても無効なキャッシュラインが足りないためにECCモードにせざるを得ないという状況が起きない利点がある。

【0111】

図14は、第3の実施の形態における待機準備処理のフローチャートである。

50

ステップF601で、キャッシュコントローラ21はプロセッサ10から待機信号を受ける。

するとキャッシュコントローラ21はステップF602で、不揮発性メモリ部22における有効なキャッシュラインCLを1つ選択する。

次にキャッシュコントローラ21はステップF603で、選択したキャッシュラインCLのECCビットEbを「1」にする。つまりECCモードのキャッシュラインとする。

#### 【0112】

ステップF604でキャッシュコントローラ21は、選択しているキャッシュラインCLのECCデータECを計算する。

そしてステップF605でキャッシュコントローラ21は、ECCデータECを、そのキャッシュラインCLにおけるECCデータ領域に書き込む。 10

これによって、1つの有効なキャッシュラインCLがECCモードのキャッシュラインCLとされたこととなる。

#### 【0113】

ステップF606ではキャッシュコントローラ21は、全ての有効なキャッシュラインCLについて、ステップF602～F605の処理を完了したか否かを確認する。完了していないければステップF602に戻って、残りのキャッシュラインモードのキャッシュラインCLについてステップF602～F605の処理を行う。

#### 【0114】

全てのキャッシュラインモードのキャッシュラインについて、ステップF602～F605のECCモード化の処理を完了したら、キャッシュコントローラ21はステップF607に進み、プロセッサ10に対して待機完了信号を送って、一連の処理を終える。 20

#### 【0115】

第3の実施の形態における復帰処理については、図8と同様に考えればよいが、異なる点は図8のステップF402は、ECCモードのキャッシュラインCLを1つ選択すること、ステップF403が不要なことである。ステップF404は、選択したキャッシュラインにおけるECCデータを用いて、当該ECCモードのキャッシュラインCLのエラー訂正を行うことになる。

つまり、ECCデータECが、エラー訂正対象のキャッシュラインCL内に記憶されていることで、復帰処理が簡易化される。 30

#### 【0116】

以上のように第3の実施の形態は、不揮発性メモリ部22には、キャッシュデータ記憶領域及び待機状態データ領域(ECCデータ領域)を有するキャッシュラインCLが複数設定されている。

キャッシュコントローラ21は、待機準備処理として、少なくとも一部のキャッシュラインCLについて、そのキャッシュラインCLに記憶されているキャッシュデータのエラー訂正可能なECCデータECを含む待機状態データを生成し、該待機状態データを待機状態データ領域に記憶させる。また当該キャッシュラインCLを、待機状態データを記憶したキャッシュラインであることを示す状態とする処理(ECCビットEb=1とする処理)を行う。 40

この第3の実施の形態でも第1、第2の実施の形態と同様の効果が得られる。加えて待機準備処理や復帰処理が簡易化できる。また待機準備処理の際にキャッシュラインCLが足りないために有効なキャッシュラインCLを無効化して用いる必要はなくなる。

#### 【0117】

<5. 第4の実施の形態>

第4の実施の形態を説明する。これは待機状態データとしてEDCデータ(エラー検出データ)を用いる例である。

図15にキャッシュラインCLのデータ構造を示す。これは第1の実施の形態で説明した図2のデータ構造において、ECCビットEbを、EDCビットEDbに代えた例であ

る。

E D C ビット E D b は、不揮発性キャッシュメモリ 11 が待機状態にあるときに用いるビットであり、キャッシュライン C L が E D C モードとされていることを示すビットである。他は図 3 で説明したものと同様である。

( 0 1 1 8 )

動作状態において、すべてのキャッシュライン C\_L がキャッシュラインモードとなることは図 3 で説明したものと同様である。

また動作状態から待機状態に移行する際に、キャッシュラインCLは、キャッシュラインモードにあるキャッシュラインCLとEDCモードにあるキャッシュラインCLとに分割される。

この第4の実施の形態では、図16に示すように、EDCモードのキャッシュラインCLでは、EDCビットEDbは「1」とされ、有効ビットVb、ダーティービットDbは「0」とされる。

そして EDC モードのキャッシュライン CL は、制御ビット以外の領域は EDC 領域とされる。EDC 領域はキャッシュアドレスを格納するキャッシュアドレス領域と EDC データ ED を格納する EDC データ領域とで構成される。

[ 0 1 1 9 ]

この図16に示すデータ構造は、基本的には第1の実施の形態のECCデータECをEDCデータEDに変更したものであり、基本的な構造やデータ内容は第1の実施の形態において図4について説明した毛のと同様である。

待機状態においてキャッシュラインモードにあるキャッシュライン C L の中で、データビット D b、アドレス上位ビット、キャッシュデータ C D を E D C 対象データと呼ぶ。そして、E D C 対象データをエラー検出符号化して E D C データを得る。

ここで、EDC 対象データと EDC データを併せたものがエラー訂正符号であり、生成されたエラー訂正符号のうち、EDC 対象データ以外の部分（EDC パリティ）を EDC データとする。

〔 0 1 2 0 1 〕

EDCデータは、EDCモードにあるキャッシュラインCLのEDCデータ領域に格納される。

その際、EDCデータがどのキャッシュラインCLのものであるかを知るために、キャッシュアドレス領域にキャッシュラインのアドレス(キャッシュアドレスCA)が格納される。

【0131】

図17に、通常の動作状態から待機状態への移行プロセスにおいて、プロセッサ10から待機信号を受けた場合のキャッシュコントローラ21の処理例、つまり待機準備処理の例を示す。基本的な処理の流れは図7と同様となる。

ステップ F701 で、キャッシュコントローラ 21 はプロセッサ 10 から待機信号を受ける。

キャッシュコントローラ 21 はステップ F702 で、不揮発性メモリ部 22 における有効なキャッシュライン C1 の数をカウントする。

次にキャッシュコントローラ21はステップF703で、EDCモードに必要なキャッシュラインC1の数を求める。必要キャッシュライン数Eとする。

【0122】 次にステップE704でキャッシュコントローラ21は、無効なキャッシュラインC1

の数  $I$  が、必要キャッシュライン数  $E$  より大きいか否かを確認する。  
 $I = E$  であれば、無効なキャッシュライン  $C_L$  を用いて  $EDC$  モードのキャッシュラインを形成できるため、キャッシュコントローラ 21 はステップ F709 に進み、無効なキャッシュライン  $C_L$  を  $E$  個選ぶ。そして選択した  $E$  個のキャッシュライン  $C_L$  については

、星と E D C ビット E D B を「1」に、ターディービット D C と「0」にする。

それぞれについて、E D C データ E D を計算する。

そしてステップ F 7 1 1 でキャッシュコントローラ 2 1 は、キャッシュアドレス C A と E D C データ E D を、ステップ F 7 0 9 で選択したキャッシュライン C L に書き込む。

#### 【 0 1 2 3 】

以上で不揮発性メモリ部 2 2 での待機準備が完了したことになり、キャッシュコントローラ 2 1 はステップ F 7 1 2 でプロセッサ 1 0 に対して待機完了信号を送って、一連の処理を終える。

#### 【 0 1 2 4 】

一方、ステップ F 7 0 4 で無効キャッシュライン数 I が必要キャッシュライン数 E より小さい場合は、キャッシュコントローラ 2 1 はステップ F 7 0 5 に進む。なおステップ F 7 0 5 ~ F 7 0 8 は、図 7 のステップ F 3 0 5 ~ F 3 0 8 と同様であるため説明を省略する。

#### 【 0 1 2 5 】

次に図 1 8 で、待機状態から動作状態への移行プロセスにおいて、プロセッサ 1 0 から復帰信号を受けた場合のキャッシュコントローラ 2 1 の処理例、つまり復帰処理の例を説明する。

ステップ F 9 0 1 で、キャッシュコントローラ 2 1 はプロセッサ 1 0 から復帰信号を受ける。

#### 【 0 1 2 6 】

キャッシュコントローラ 2 1 はステップ F 9 0 2 で、不揮発性メモリ部 2 2 においてキャッシュラインモードとされているキャッシュライン C L を一つ選ぶ。

ステップ F 9 0 3 でキャッシュコントローラ 2 1 は、選択したキャッシュライン C L のキャッシュアドレス C A が記憶されている E D C モードのキャッシュライン C L を選ぶ。

ステップ F 9 0 4 でキャッシュコントローラ 2 1 は、ステップ F 9 0 2 で選択したキャッシュラインモードのキャッシュライン C L に対応する E D C データ E D を、ステップ F 9 0 3 で選択した E D C モードのキャッシュライン C L から読み出し、キャッシュラインモードのキャッシュライン C L のエラー検出処理を行う。

#### 【 0 1 2 7 】

ここでエラーが検出されなければ、そのキャッシュラインモードのキャッシュライン C L はデータ消失等がなく、正常な状態である。

一方、エラーが検出されたキャッシュライン C L は、何らかのエラーが発生し、データ内容 ( E D C 対象データの内容 ) は、正確な内容ではなくなっている。

そこで、エラーが検出された場合は、ステップ F 9 0 5 から F 9 0 6 に進み、キャッシュコントローラ 2 1 は当該キャッシュライン C L を無効化する。具体的には有効ビット V b を「 0 」とする。

エラーが検出されなければ、当該キャッシュライン C L をそのまま維持する。

#### 【 0 1 2 8 】

ステップ F 9 0 7 では、キャッシュコントローラ 2 1 は、全てのキャッシュラインモードのキャッシュラインについて、エラー検出処理が完了したか否かを確認する。完了していないければステップ F 9 0 2 に戻って、残りのキャッシュラインモードのキャッシュライン C L についてステップ F 9 0 2 ~ F 9 0 6 の処理を行う。

#### 【 0 1 2 9 】

全てのキャッシュラインモードのキャッシュラインについて、エラー検出処理が完了したら、キャッシュコントローラ 2 1 はステップ F 9 0 8 に進み、全ての E D C モードのキャッシュライン C L については、E D C ビット E D b を「 0 」として、単なる無効なキャッシュライン C L にする。

そしてステップ F 9 0 9 でキャッシュコントローラ 2 1 はプロセッサ 1 0 に復帰完了信号を終え、一連の復帰処理を終える。

#### 【 0 1 3 0 】

以上の第 4 の実施の形態のように、E D C データ E D を待機状態データとして用いるこ

10

20

30

40

50

ともできる。

復帰処理においてエラーを検出したら、そのキャッシュライン C L は無効化することで、誤ったキャッシュデータがその後用いられることが防止される。

キャッシュライン当たりのパリティ領域としては、ECCデータ領域に比べてEDCデータ領域の場合、削減することができる。

#### 【0131】

なお、エラーを検出したらそのキャッシュラインは無効化するわけであるが、この場合には、ダーティービット D b が「1」であるキャッシュライン C L が無効化されることによって、不揮発性キャッシュメモリ 1 1 でのデータ更新がメインメモリ 1 2 に反映されなくなる場合が生ずる。そこで、これを防ぐために、待機信号を得てデータの変換を行うプロセスに至る前に、ダーティービット D b が「1」であるキャッシュライン C L をまとめてメインメモリ 1 2 に書き戻しておくことが望ましい。10

#### 【0132】

< 6. 変形例 >

本開示は、上述の実施の形態に限定されるものではなく、本開示の要旨を逸脱しない範囲でその他様々な構成を取り得る。

#### 【0133】

上記の各実施の形態においては、待機信号を得てデータの変換を行うプロセスに至る前に、ダーティービット D b が「1」であるキャッシュライン C L をまとめてメインメモリ 1 2 に書き戻すこともできる。書き戻しを行ったらダーティービット D b を「0」にする。このようにすると、有効なキャッシュライン C L を ECC モード ( 又は EDC モード ) に割り当てる際に、その都度ダーティービット D b のチェックを行う必要がなくなる。20

#### 【0134】

また第 4 の実施の形態では、EDC データ ED を使う例を、第 1 の実施の形態と同様のデータ構造に適用した例で述べたが、EDC データを、第 2, 第 3 の実施の形態と同様のデータ構造において ECC データに代えて適用することも当然想定される。

#### 【0135】

また各実施の形態のでは、電力供給遮断の準備として、プロセッサ 1 0 からの待機信号があったとき、つまり電源遮断直前に待機状態データを生成して不揮発性メモリ部 2 2 に記憶する待機準備処理を行うこととしたが、他のタイミングで待機準備処理を行ってもよい。30

例えばある程度定期的に待機準備処理を行うなどである。但し、電源遮断直前のキャッシュデータ内容が必ずしも反映されない恐れがあるため、電源遮断直前以外で実行する場合は、その後、ECC モード ( 又は EDC モード ) としたキャッシュラインの有効、無効を管理する必要がある。

また、例えば電池駆動のコンピュータシステムなどにおいて、不揮発性キャッシュメモリ 1 1 が電源電圧状態を検知し、電源落ちとなるタイミング直前に自動的に待機準備処理を行うような例も考えられる。

#### 【0136】

復帰処理については、再開後の最初のキャッシュ動作の実行前におこなっておけばよい。従って、プロセッサ 1 0 からの復帰信号以外で復帰処理を行う例も考えられる。

#### 【0137】

また、エラー訂正或いはエラー検出以外の目的で、動作状態と待機状態でキャッシュラインのデータ構造を変えることもできる。

#### 【0138】

なお本技術は以下のような構成も採ることができる。

(1) キャッシュデータの記憶に用いられる不揮発性メモリ部と、

上記不揮発性メモリ部に対してのキャッシュデータの読み出し及び書き込みを制御するとともに、電力供給遮断の準備として、待機状態データを生成して上記不揮発性メモリ部50

に記憶する待機準備処理を行い、また電力供給再開の際に、上記待機状態データを用いて上記不揮発性メモリ部に記憶されたキャッシュデータの復帰処理を行うキャッシュコントローラと、

を備えた不揮発性キャッシュメモリ。

(2) 上記キャッシュコントローラは、

上記待機準備処理として、少なくとも上記不揮発性メモリ部上に記憶されているキャッシュデータについてのエラー訂正を可能とするエラー訂正データを含む上記待機状態データを生成して上記不揮発性メモリ部上に記憶させ、

上記復帰処理として、上記待機状態データにおける上記エラー訂正データを用いて上記不揮発性メモリ部に記憶されたキャッシュデータのエラー訂正処理を行う上記(1)に記載の不揮発性キャッシュメモリ。10

(3) 上記不揮発性メモリ部には、キャッシュデータ記憶に用いられるキャッシュラインが複数設定されており、

上記キャッシュコントローラは、

上記待機準備処理として、少なくとも一部のキャッシュラインについて、そのキャッシュラインに記憶されているキャッシュデータのエラー訂正可能なエラー訂正データを含む待機状態データを生成し、該待機状態データを他のキャッシュラインに記憶させ、かつ当該他のキャッシュラインを、待機状態データを記憶したキャッシュラインであることを示す状態とする処理を行う上記(1)又は(2)に記載の不揮発性キャッシュメモリ。

(4) 上記エラー訂正データを含む待機状態データには、さらに当該エラー訂正データでエラー訂正を行うキャッシュデータが記憶されたキャッシュラインのアドレス情報が含まれている上記(3)に記載の不揮発性キャッシュメモリ。20

(5) 上記キャッシュコントローラは、上記待機状態データを、その待機状態データが対象とするキャッシュデータを記憶したキャッシュラインに対応する領域に記憶させる上記(3)に記載の不揮発性キャッシュメモリ。

(6) 上記不揮発性メモリ部には、キャッシュデータ記憶領域及び待機状態データ領域を有するキャッシュラインが複数設定されており、

上記キャッシュコントローラは、

上記待機準備処理として、少なくとも一部のキャッシュラインについて、そのキャッシュラインに記憶されているキャッシュデータのエラー訂正可能なエラー訂正データを含む待機状態データを生成し、該待機状態データを上記待機状態データ領域に記憶させ、かつ当該キャッシュラインを、待機状態データを記憶したキャッシュラインであることを示す状態とする処理を行う上記(1)又は(2)に記載の不揮発性キャッシュメモリ。30

(7) 上記キャッシュコントローラは、

上記待機準備処理として、少なくとも上記不揮発性メモリ部上に記憶されているキャッシュデータについてのエラー検出を可能とするエラー検出データを含む上記待機状態データを生成して上記不揮発性メモリ部上に記憶させ、

上記復帰処理として、上記待機状態データにおける上記エラー検出データを用いて上記不揮発性メモリ部に記憶されたキャッシュデータのエラー検出処理を行う上記(1)に記載の不揮発性キャッシュメモリ。40

(8) 上記キャッシュコントローラは、

上記エラー検出処理においてエラーを検出したキャッシュデータを無効化する処理を行う上記(7)に記載の不揮発性キャッシュメモリ。

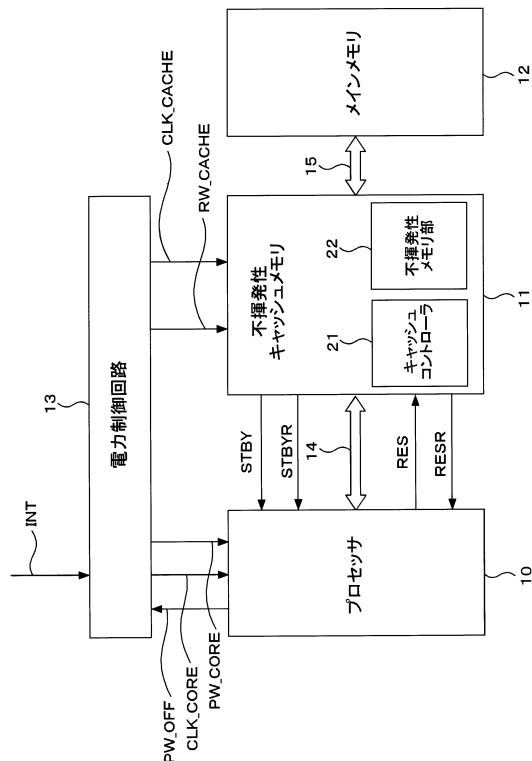
(9) 上記キャッシュコントローラは、電力供給を遮断する待機状態に移行することを要求する待機信号を受け取ることで、上記待機準備処理を行う上記(1)乃至(8)に記載の不揮発性キャッシュメモリ。

【符号の説明】

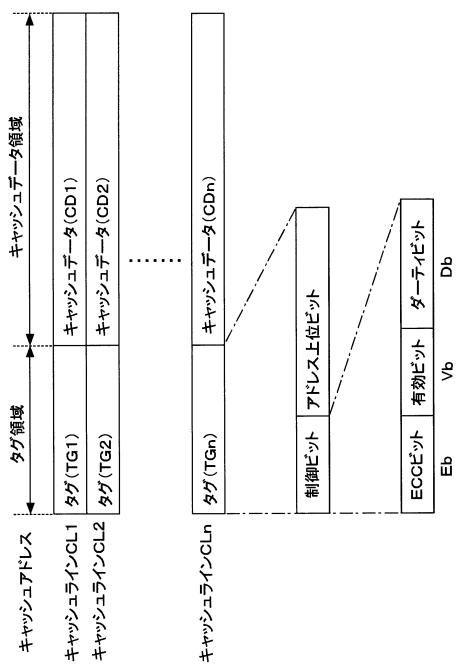
【0139】

10 プロセッサ、11 不揮発性キャッシュメモリ、12 メインメモリ、13 電源制御回路、21 キャッシュコントローラ、22 不揮発性メモリ部、50

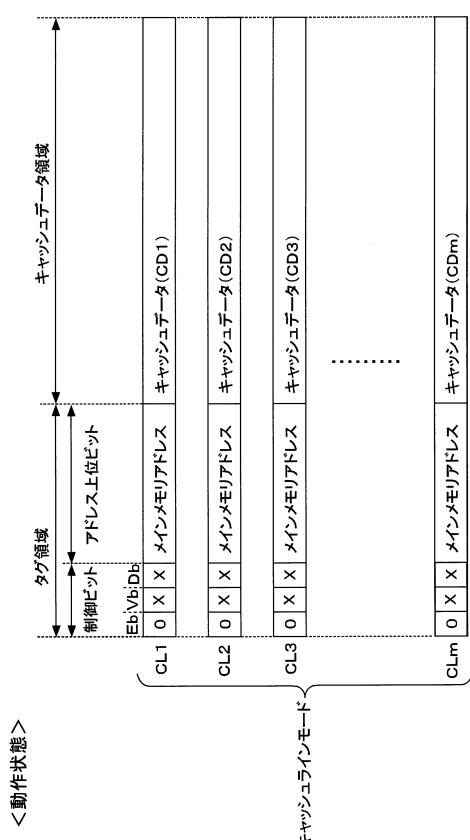
【図1】



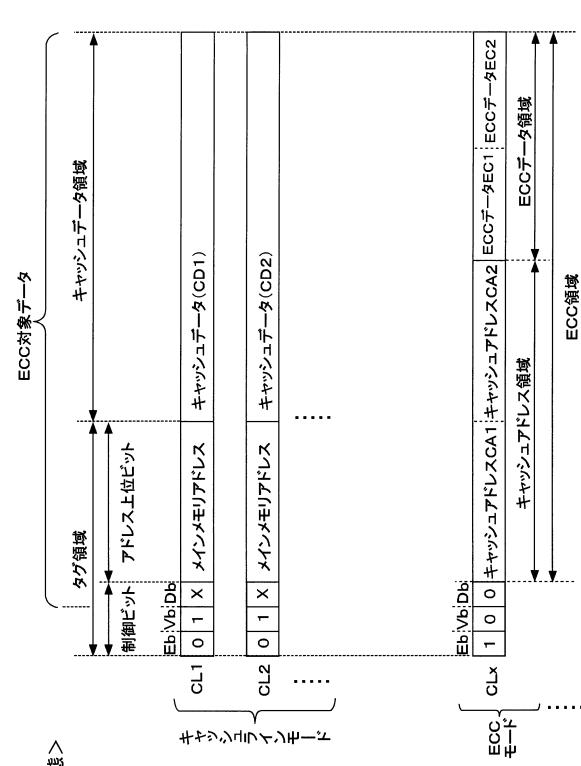
【図2】



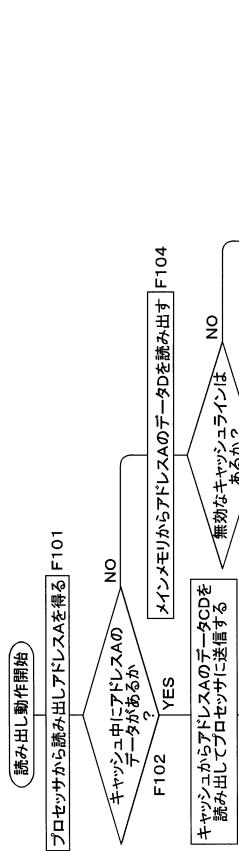
【図3】



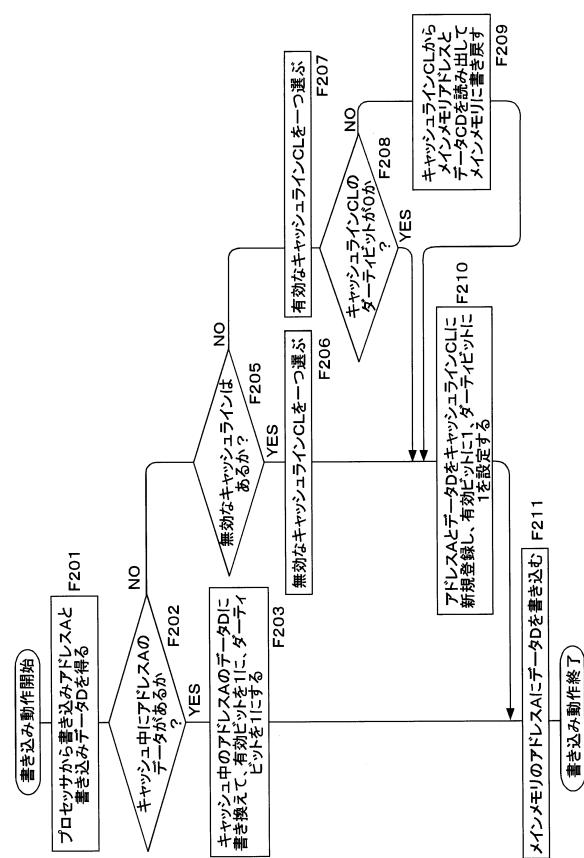
【図4】



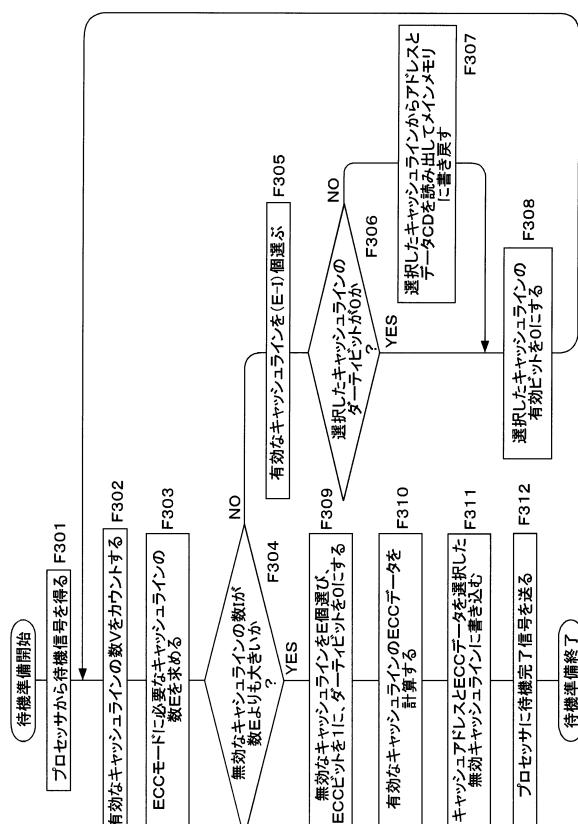
【図5】



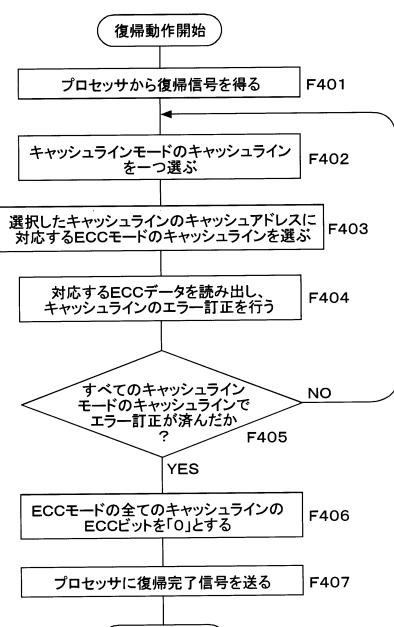
【図6】



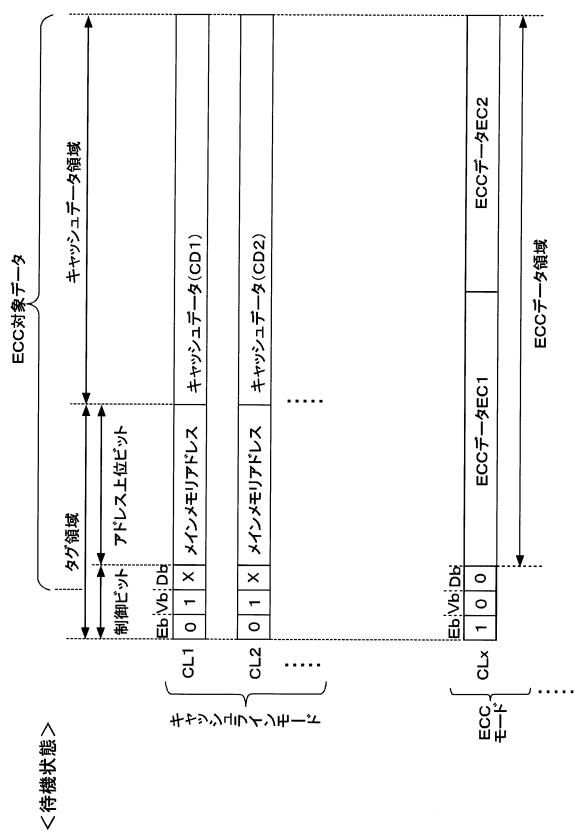
【図7】



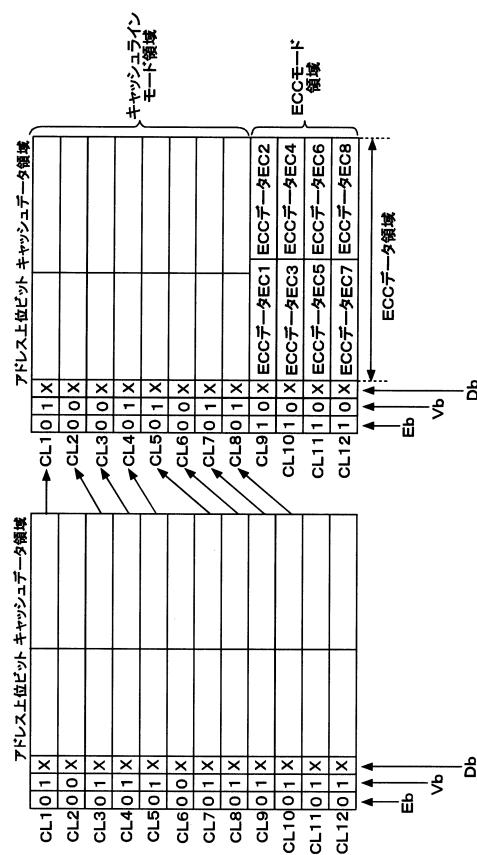
【図8】



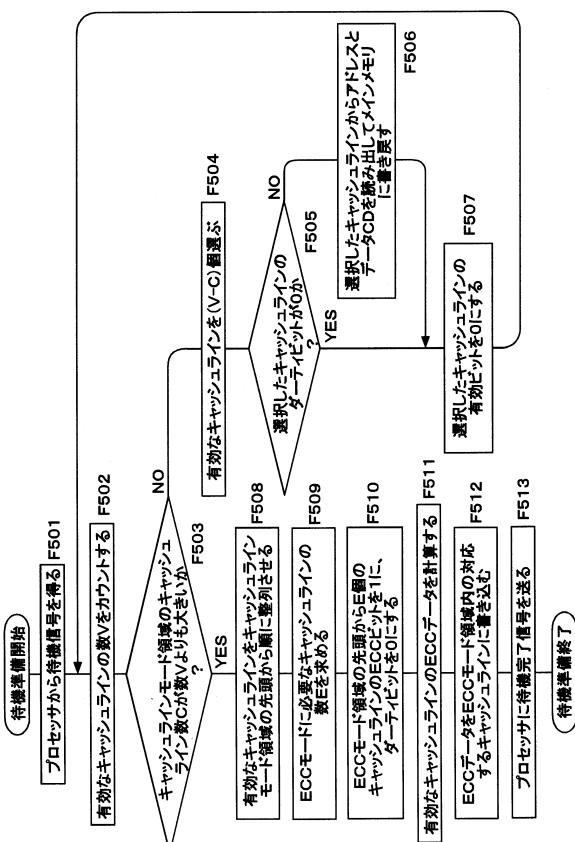
【図 9】



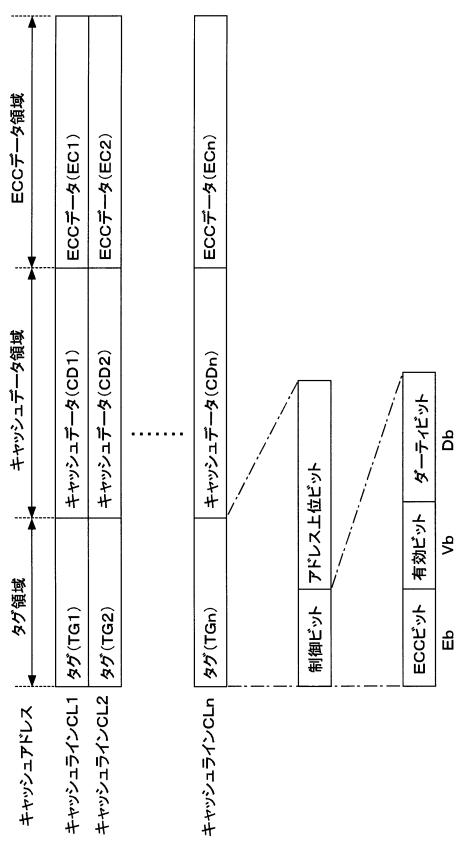
【図 10】



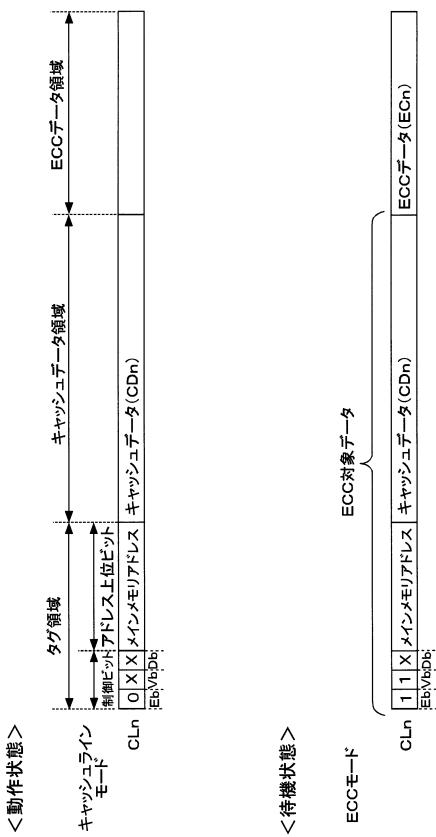
【図 11】



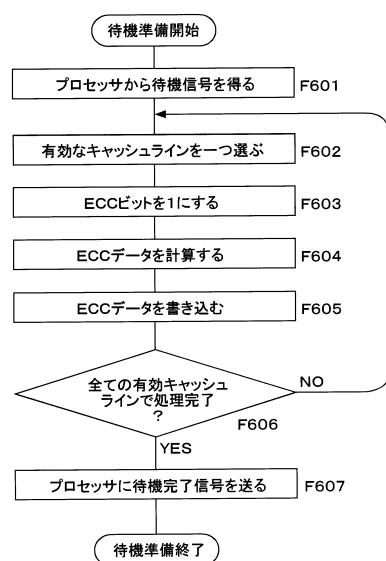
【図 12】



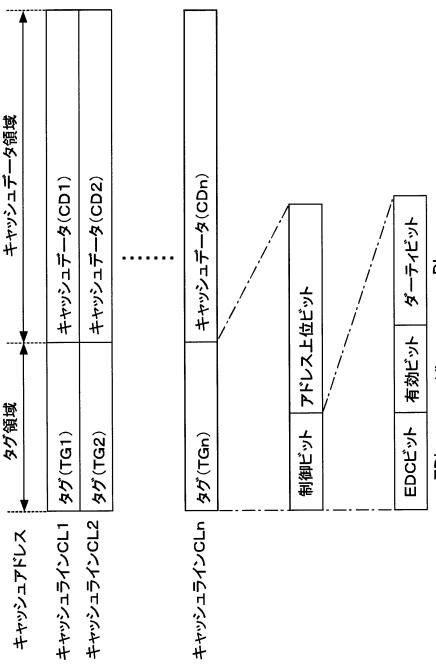
【図13】



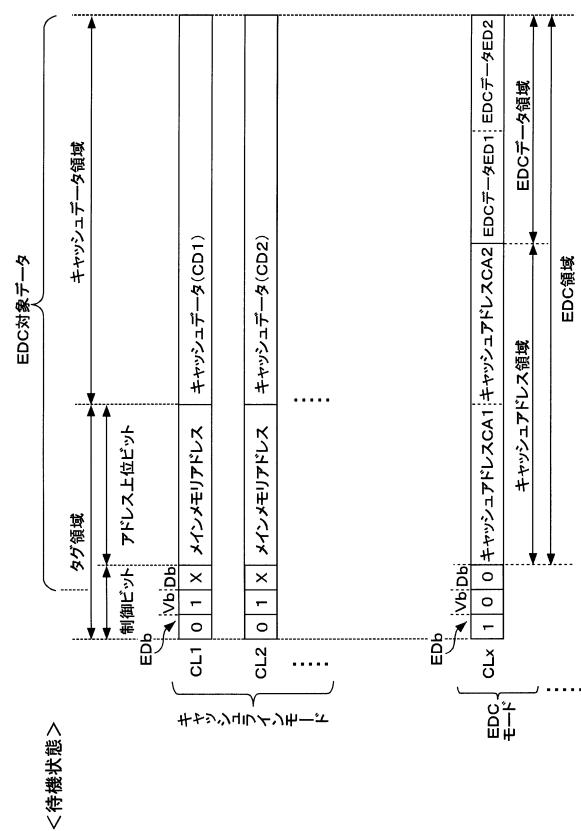
【図14】



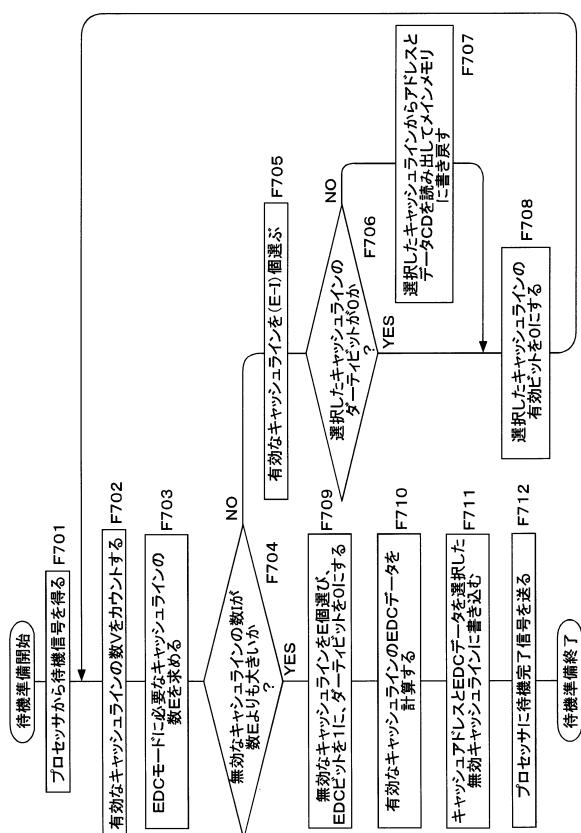
【図15】



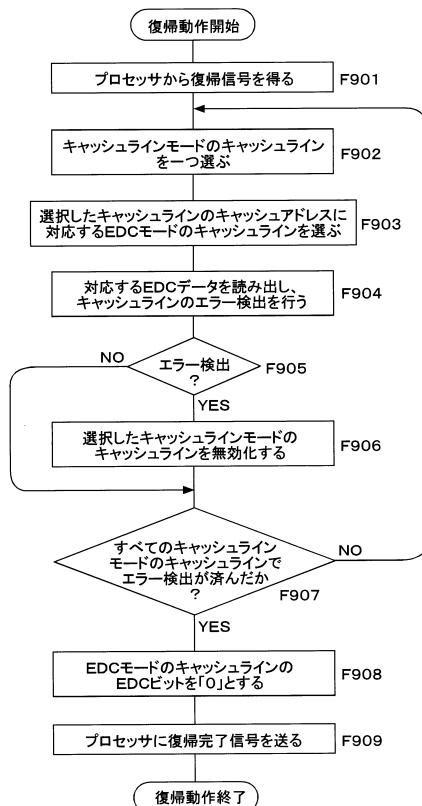
【図16】



【図17】



【図18】



---

フロントページの続き

(72)発明者 大森 広之  
東京都港区港南1丁目7番1号 ソニー株式会社内

(72)発明者 別所 和宏  
東京都港区港南1丁目7番1号 ソニー株式会社内

(72)発明者 浅山 徹哉  
東京都港区港南1丁目7番1号 ソニー株式会社内

(72)発明者 山根 一陽  
東京都港区港南1丁目7番1号 ソニー株式会社内

(72)発明者 内田 裕行  
東京都港区港南1丁目7番1号 ソニー株式会社内

審査官 原 秀人

(56)参考文献 國際公開第2011/107882 (WO, A1)  
特表2004-502237 (JP, A)  
特開2007-041798 (JP, A)  
特開2011-138273 (JP, A)  
國際公開第2011/081168 (WO, A1)  
米国特許出願公開第2010/0023681 (US, A1)  
Hongbin Sun 外4名, Design Techniques to Improve the Device Write Margin forMRAM-based Cache Memory, ProceedingGLSVLSI '11 Proceedings of the 21st edition of the great lakes symposium on Great lakes symposium on VLSI, 米国, ACM, 2011年 5月, Pages 97-102

(58)調査した分野(Int.Cl., DB名)

G 06 F 12 / 08  
G 06 F 12 / 16