

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5633125号
(P5633125)

(45) 発行日 平成26年12月3日(2014.12.3)

(24) 登録日 平成26年10月24日(2014.10.24)

(51) Int.Cl.

G06F 12/16 (2006.01)
G06K 17/00 (2006.01)

F 1

G06F 12/16 340 P
G06K 17/00 D
G06K 17/00 E
G06K 17/00 F

請求項の数 4 (全 29 頁)

(21) 出願番号	特願2009-176667 (P2009-176667)
(22) 出願日	平成21年7月29日 (2009.7.29)
(65) 公開番号	特開2011-34125 (P2011-34125A)
(43) 公開日	平成23年2月17日 (2011.2.17)
審査請求日	平成24年7月24日 (2012.7.24)

前置審査

(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	100121131 弁理士 西川 孝
(74) 代理人	100082131 弁理士 稲本 義雄
(72) 発明者	山本 英雄 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	赤井田 優郎 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 桜井 茂行

最終頁に続く

(54) 【発明の名称】情報処理方法、情報処理装置、およびプログラム

(57) 【特許請求の範囲】

【請求項 1】

所定のブロック単位で情報が記録される記録媒体を用いた情報処理装置の情報処理方法において、

前記記録媒体は、

複数のブロックから構成され、

前記ブロック単位でデータを記憶するデータ領域と、

前記データ領域の各ブロックに割当てられた複数のブロック番号を記憶するポインタ領域と、

更新前のデータが記憶されている前記データ領域の対象ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の対照ブロックポインタ、更新後のデータが記憶される前記データ領域の更新ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の更新ブロックポインタ、および記憶内容の新しさに関する最新情報を記憶する第1の領域と

を有し、

前記情報処理装置による、

前記第1の領域に記憶されている全ての前記更新ブロックポインタにそれぞれ対応する前記ポインタ領域が記憶する全てのブロック番号にそれぞれ対応する前記データ領域の前記更新ブロックのデータを一括して消去させる第1の消去ステップと、

一括してデータが消去された、前記データ領域の全ての前記更新ブロックの少なくと

10

20

も一部に更新後のデータを書き込ませる書込みステップと、

前記更新後のデータが書き込まれたデータ領域のブロックのブロック番号を、前記ポインタ領域のブロックに記憶させる第1の記憶ステップと、

データ領域のブロック番号を記憶させたポインタ領域の前記ブロックのブロック番号を、前記第1の領域に対象ブロックポインタとして記憶させる第2の記憶ステップとを含み、

前記書込みステップの処理は、書込みコマンドに応じて実行され、

前記第1の消去ステップの処理は、前記書込みコマンドの有無に拘わらず、前記書込みステップの処理に先行した所定タイミングで実行される

情報処理方法。

10

【請求項2】

前記記録媒体は、さらに、

更新前のデータが記憶されている前記データ領域の対象ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の対照ブロックポインタ、更新後のデータが記憶される前記データ領域の更新ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の更新ブロックポインタ、および記憶内容の新しさに関する最新情報を記憶する第2の領域を

有し、

前記情報処理装置による、

前記第1の消去ステップは、前記最新情報に基づいて選択された前記第1または第2の領域の一方に記憶されている全ての前記更新ブロックポインタにそれぞれ対応する前記ポインタ領域が記憶する全てのブロック番号にそれぞれ対応する前記データ領域の前記更新ブロックのデータを一括して消去させ、

20

前記第2の記憶ステップは、データ領域のブロック番号を記憶させたポインタ領域の前記ブロックのブロック番号を、前記第1または第2の領域の他方に記憶させ、

前記第2の記憶ステップの後、前記第1または第2の領域の前記一方に記憶されている前記対象ブロックポインタ、前記更新ブロックポインタ、および前記最新情報を消去させる第2の消去ステップをさらに含む

請求項1に記載の情報処理方法。

【請求項3】

30

所定のブロック単位で情報が記録される記録媒体に対して情報を記録する情報処理装置において、

前記記録媒体は、

複数のブロックから構成され、

前記ブロック単位でデータを記憶するデータ領域と、

前記データ領域の各ブロックに割当てられた複数のブロック番号を記憶するポインタ領域と、

更新前のデータが記憶されている前記データ領域の対象ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の対照ブロックポインタ、更新後のデータが記憶される前記データ領域の更新ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の更新ブロックポインタ、および記憶内容の新しさに関する最新情報を記憶する第1の領域と

40

を有し、

前記第1の領域に記憶されている全ての前記更新ブロックポインタにそれぞれ対応する前記ポインタ領域が記憶する全てのブロック番号にそれぞれ対応する前記データ領域の前記更新ブロックのデータを一括して消去させる第1の消去手段と、

一括してデータが消去された、前記データ領域の全ての前記更新ブロックの少なくとも一部に更新後のデータを書き込ませる書込み手段と、

前記更新後のデータが書き込まれたデータ領域のブロックのブロック番号を、前記ポインタ領域のブロックに記憶させる第1の記憶手段と、

50

データ領域のブロック番号を記憶させたポインタ領域の前記ブロックのブロック番号を、前記第1の領域に対象ブロックポインタとして記憶させる第2の記憶手段とを備え、

前記書き込み手段は、書き込みコマンドに応じて動作し、

前記第1の消去手段は、前記書き込みコマンドの有無に拘わらず、前記書き込み手段の動作に先行した所定タイミングで動作する

情報処理装置。

【請求項4】

所定のブロック単位で情報が記録される記録媒体を用いた情報処理装置の制御用のプログラムであって、

10

前記記録媒体は、

複数のブロックから構成され、

前記ブロック単位でデータを記憶するデータ領域と、

前記データ領域の各ブロックに割当てられた複数のブロック番号を記憶するポインタ領域と、

更新前のデータが記憶されている前記データ領域の対象ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の対照ブロックポインタ、更新後のデータが記憶される前記データ領域の更新ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の更新ブロックポインタ、および記憶内容の新しさに関する最新情報を記憶する第1の領域と

20

を有し、

前記第1の領域に記憶されている全ての前記更新ブロックポインタにそれぞれ対応する前記ポインタ領域が記憶する全てのブロック番号にそれぞれ対応する前記データ領域の前記更新ブロックのデータを一括して消去させる第1の消去ステップと、

一括してデータが消去された、前記データ領域の全ての前記更新ブロックの少なくとも一部に更新後のデータを書きませる書き込みステップと、

前記更新後のデータが書き込まれたデータ領域のブロックのブロック番号を、前記ポインタ領域のブロックに記憶させる第1の記憶ステップと、

データ領域のブロック番号を記憶させたポインタ領域の前記ブロックのブロック番号を、前記第1の領域に対象ブロックポインタとして記憶させる第2の記憶ステップと

30

を含む処理を情報処理装置のコンピュータに実行させ、

前記書き込みステップの処理は、書き込みコマンドに応じて実行され、

前記第1の消去ステップの処理は、前記書き込みコマンドの有無に拘わらず、前記書き込みステップの処理に先行した所定タイミングで実行される

プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理方法、情報処理装置、およびプログラムに関し、特に、例えば、EEPROM(electrically erasable programmable ROM)のように、記録領域に既に記録されているデータに上書きすることができず、既に記録されているデータを消去してから新たなデータを記録しなければならない記録媒体に対してデータ記録する場合に用いて好適な情報処理方法、情報処理装置、およびプログラムに関する。

40

【背景技術】

【0002】

現在、Felica(商標)に代表される非接触通信システムが広く実用化されている。

【0003】

既存の非接触通信システムにおいては、リーダ/ライタ(以下、R/Wと略記する)が、コントローラからの制御に従って、ICカードとデータを非接触で通信する。

【0004】

50

この I C カードは、記録媒体を内蔵しており、例えば、非接触通信システムが電子マネー・システムに利用される場合においては、プリペイドマネーの残高や購買履歴などの情報が記録され、駅改札システムに利用される場合においては、定期券情報や乗降履歴情報が記録される。

【 0 0 0 5 】

I C カードに内蔵される記録媒体としては、不揮発性メモリのEEPROMが適用されることが多い。EEPROMは、その特性として、データを書き込む場合、記録領域に既に記録されているデータに上書きすることができず、既に記録されているデータを消去してから新たなデータを記録しなければならない。

【 0 0 0 6 】

ところで、I C カードには、自らはバッテリを有していないバッテリレス型のI C カードがあり、このようなバッテリレス型のI C カードは、R / Wから放射される電磁波を利用した電磁誘導により駆動電力を取得できるようになされている。

【 0 0 0 7 】

しかしながら、I C カードとR / Wとの間でデータを非接触で通信する場合、I C カードが内蔵するメモリにアクセスしている途中で、電磁波の受信状態が不良になると十分な駆動電力を取得することができなくなり、メモリにおけるデータの整合性に欠陥が生じてしまうことが発生し得る。以下、メモリにおけるデータの整合性に欠陥が生じた状態を、メモリコラプション (Memory Corruption) と称する。

【 0 0 0 8 】

また、非接触通信に限らず、I C カードとR / Wとを接触させてデータの送受信を行う場合においても、ユーザが、I C カードを、R / Wに対して自由に抜き差しすることができるときには、メモリにアクセスしている途中で、I C カードがR / Wから抜かれると、やはり、メモリコラプションが生じてしまうことが起こり得た。

【 0 0 0 9 】

そこで従来から、メモリコラプションに対する様々な対処方法が提案されている。

【 0 0 1 0 】

例えば、本出願人は、I C カードに、前回の通信結果のデータと前々回の通信結果のデータを記録しておき、古い方のデータ（前々回のデータ）を、今回の通信結果のデータで更新するようにし、更新の処理を良好に行うことができなかった場合（古いデータが破壊されてしまった場合）でも、前回の有効なデータが残るようにした方法を提案している（特許文献1参照）。

【先行技術文献】

【特許文献】

【 0 0 1 1 】

【特許文献1】特開平11-25003号公報

【発明の概要】

【発明が解決しようとする課題】

【 0 0 1 2 】

特許文献1の方法によれば、古い方のデータを更新中に例えば電源が遮断されたとしても、少なくとも前々回のデータまたは前回のデータの一方を安定して読み出すことができる。

【 0 0 1 3 】

しかしながら、特許文献1の方法には、データの書き込み処理において、その所要時間をより短縮できる余地が残されていた。

【 0 0 1 4 】

本発明は、このような状況に鑑みてなされたものであり、データを書き込む場合、記録領域に既に記録されているデータに上書きすることができず、既に記録されているデータを消去してから新たなデータを記録しなければならないEEPROMなどの記録媒体に対し、データの書き込み処理に要する時間を短縮できるようにするものである。

10

20

30

40

50

【課題を解決するための手段】**【0015】**

本発明の一側面である情報処理方法は、所定のブロック単位で情報が記録される記録媒体を用いた情報処理装置の情報処理方法において、前記記録媒体は、複数のブロックから構成され、前記ブロック単位でデータを記憶するデータ領域と、前記データ領域の各ブロックに割当てられた複数のブロック番号を記憶するポインタ領域と、更新前のデータが記憶されている前記データ領域の対象ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の対照ブロックポインタ、更新後のデータが記憶される前記データ領域の更新ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の更新ブロックポインタ、および記憶内容の新しさに関する最新情報を記憶する第1の領域とを有し、前記情報処理装置による、前記第1の領域に記憶されている全ての前記更新ブロックポインタにそれぞれ対応する前記ポインタ領域が記憶する全てのブロック番号にそれぞれ対応する前記データ領域の前記更新ブロックのデータを一括して消去させる第1の消去ステップと、一括してデータが消去された、前記データ領域の全ての前記更新ブロックの少なくとも一部に更新後のデータを書きませる書き込みステップと、前記更新後のデータが書き込まれたデータ領域のブロックのブロック番号を、前記ポインタ領域のブロックに記憶させる第1の記憶ステップと、データ領域のブロック番号を記憶させたポインタ領域の前記ブロック番号を、前記第1の領域に対象ブロックポインタとして記憶させる第2の記憶ステップとを含み、前記書き込みステップの処理は、書き込みコマンドに応じて実行され、前記第1の消去ステップの処理は、前記書き込みコマンドの有無に拘わらず、前記書き込みステップの処理に先行した所定タイミングで実行される。
10

【0016】

前記記録媒体は、さらに、更新前のデータが記憶されている前記データ領域の対象ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の対照ブロックポインタ、更新後のデータが記憶される前記データ領域の更新ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の更新ブロックポインタ、および記憶内容の新しさに関する最新情報を記憶する第2の領域を有し、前記情報処理装置による、前記第1の消去ステップは、前記最新情報に基づいて選択された前記第1または第2の領域の一方に記憶されている全ての前記更新ブロックポインタにそれぞれ対応する前記ポインタ領域が記憶する全てのブロック番号にそれぞれ対応する前記データ領域の前記更新ブロックのデータを一括して消去させ、前記第2の記憶ステップは、データ領域のブロック番号を記憶させたポインタ領域の前記ブロックのブロック番号を、前記第1または第2の領域の他方に記憶させ、前記第2の記憶ステップの後、前記第1または第2の領域の前記一方に記憶されている前記対象ブロックポインタ、前記更新ブロックポインタ、および前記最新情報を消去させる第2の消去ステップをさらに含むことができる。
20

【0019】

本発明の一側面である情報処理装置は、所定のブロック単位で情報が記録される記録媒体に対して情報を記録する情報処理装置において、前記記録媒体は、複数のブロックから構成され、前記ブロック単位でデータを記憶するデータ領域と、前記データ領域の各ブロックに割当てられた複数のブロック番号を記憶するポインタ領域と、更新前のデータが記憶されている前記データ領域の対象ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の対照ブロックポインタ、更新後のデータが記憶される前記データ領域の更新ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の更新ブロックポインタ、および記憶内容の新しさに関する最新情報を記憶する第1の領域とを有し、前記第1の領域に記憶されている全ての前記更新ブロックポインタにそれぞれ対応する前記ポインタ領域が記憶する全てのブロック番号にそれぞれ対応する前記データ領域の前記更新ブロックのデータを一括して消去させる第1の消去手段と、一括してデータが消去された、前記データ領域の全ての前記更新ブロックの少なくとも一部に更新後のデータを書きませる書き込み手段と、前記更新後のデータが書き込まれたデータ領域のブロックのブロック番号を、前記ポインタ領域のブロックに記憶させる第1の記憶手段と、データ
40
50

領域のブロック番号を記憶させたポインタ領域の前記ブロックのブロック番号を、前記第1の領域に対象ブロックポインタとして記憶させる第2の記憶手段とを備え、前記書込み手段は、書込みコマンドに応じて動作し、前記第1の消去手段は、前記書込みコマンドの有無に拘わらず、前記書込み手段の動作に先行した所定タイミングで動作する。

【0020】

本発明の一側面であるプログラムは、所定のブロック単位で情報が記録される記録媒体を用いた情報処理装置の制御用のプログラムであって、前記記録媒体は、複数のブロックから構成され、前記ブロック単位でデータを記憶するデータ領域と、前記データ領域の各ブロックに割当てられた複数のブロック番号を記憶するポインタ領域と、更新前のデータが記憶されている前記データ領域の対象ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の対照ブロックポインタ、更新後のデータが記憶される前記データ領域の更新ブロックを指示する前記ポインタ領域のブロック番号を示す1または複数の更新ブロックポインタ、および記憶内容の新しさに関する最新情報を記憶する第1の領域とを有し、前記第1の領域に記憶されている全ての前記更新ブロックポインタにそれぞれ対応する前記ポインタ領域が記憶する全てのブロック番号にそれぞれ対応する前記データ領域の前記更新ブロックのデータを一括して消去させる第1の消去ステップと、一括してデータが消去された、前記データ領域の全ての前記更新ブロックの少なくとも一部に更新後のデータを書きませる書込みステップと、前記更新後のデータが書き込まれたデータ領域のブロックのブロック番号を、前記ポインタ領域のブロックに記憶させる第1の記憶ステップと、データ領域のブロック番号を記憶させたポインタ領域の前記ブロックのブロック番号を、前記第1の領域に対象ブロックポインタとして記憶させる第2の記憶ステップとを含み、前記書込みステップの処理は、書込みコマンドに応じて実行され、前記第1の消去ステップの処理は、前記書込みコマンドの有無に拘わらず、前記書込みステップの処理に先行した所定タイミングで実行される処理を情報処理装置のコンピュータに実行させる。

10

20

【0021】

本発明の一側面においては、第1の領域に記憶されている全ての更新ブロックポインタにそれぞれ対応するポインタ領域が記憶する全てのブロック番号にそれぞれ対応するデータ領域の前記更新ブロックのデータが一括して消去され、一括してデータが消去された、前記データ領域の全ての前記更新ブロックの少なくとも一部に更新後のデータが書き込まれ、前記更新後のデータが書き込まれたデータ領域のブロックのブロック番号が、前記ポインタ領域のブロックに記憶され、データ領域のブロック番号を記憶させたポインタ領域の前記ブロックのブロック番号が、前記第1の領域に対象ブロックポインタとして記憶される。

30

【発明の効果】

【0022】

本発明の一側面によれば、メモリコラプションに対処することができる。

【0023】

また、本発明の一側面によれば、メモリコラプションに対処しつつ、データの書き込み処理に要する時間を短縮することが可能となる。

40

【図面の簡単な説明】

【0024】

【図1】本発明を適用した非接触通信システムの構成例を示すブロック図である。

【図2】図1のリーダ／ライタの構成例を示すブロック図である。

【図3】図1のI Cカードの構成例を示すブロック図である。

【図4】メモリコラプションに対する対処方法の基本原理を説明する図である。

【図5】シングルポインタ方式を説明する図である。

【図6】シングルポインタ方式を説明する図である。

【図7】ダブルポインタ方式を説明する図である。

【図8】ダブルポインタ方式を説明する図である。

50

【図9】ダブルポインタ方式に対応する読み出し処理を説明するフローチャートである。

【図10】ダブルポインタ方式に対応する書き込み処理を説明するフローチャートである。

【発明を実施するための形態】

【0025】

以下、発明を実施するための最良の形態（以下、実施の形態と称する）について、図面を参照しながら詳細に説明する。

【0026】

<1. 実施の形態>

【非接触通信システムの構成例】

図1は、本発明の実施の形態であるICカードを含む非接触通信システムの構成例を示している。 10

【0027】

この非接触通信システム10は、コントローラ11、R/W（リーダ／ライタ）12、およびICカード13から構成される。R/W12は、コントローラ11からの制御に従い、ICカード13との間で、電磁波を利用した非接触通信を行う。

【0028】

すなわち、コントローラ11がR/W12に対してICカード13との通信を指示するコマンドを送信すると、R/W12が、所定のコマンドをICカード13に送信する。当該所定のコマンドを受信したICカード13は、そのコマンドに対応する処理を行い、処理結果を示す応答データをR/W12に送信するようになされている。 20

【0029】

【R/W12の構成例】

図2は、図1のR/W12の構成例を示している。

【0030】

R/W12は、IC21、フラッシュメモリ22、変調部23、発振部24、復調部25、およびアンテナ26から構成される。 20

【0031】

IC21は、DPU(Data Processing Unit)31、SPU(Signal Processing Unit)32、SC33、およびメモリ部34がバス35を介して相互に接続されて構成される。

【0032】

また、バス35には、所定のデータを記憶するフラッシュメモリ22も接続されている。

【0033】

DPU31は、ICカード13に送信するコマンドをSPU32に出力するとともに、ICカード13から受信した応答データをSPU32から取得する。

【0034】

SPU32は、ICカード13に送信するコマンドに対して所定の変調符号化処理（例えば、BPSK(Binary Phase Shift Keying)変調符号化処理など）を行った後、変調部23に出力するとともに、ICカード13により送信してきた応答データを復調部25から取得し、そのデータに対して変調復号処理（BPSK変調の復号処理など）を行う。 40

【0035】

変調部23は、発振部24より供給された所定の周波数（例えば13.56MHz）の搬送波を、SPU32より供給されたデータで、ASK(Amplitude Shift Keying)変調し、生成された変調波を、アンテナ27を介して、電磁波としてICカード13に出力する。このとき、変調部23は、変調度を1未満にして、ASK変調を行う。すなわち、データがローレベルのときにおいても、変調波の最大振幅がゼロにならないようにする。

【0036】

復調部25は、アンテナ27を介して受信した変調波(ASK変調波)を復調し、復調されたデータをSPU32に出力する。

【0037】

50

20

30

40

50

[I C カード 1 3 の構成例]

図 3 は、図 1 の I C カード 1 3 の構成例を示している。

【 0 0 3 8 】

I C カード 1 3 は、I C 5 1、コンデンサ 5 2、およびアンテナ 5 3 から構成される。

【 0 0 3 9 】

I C 5 1 は、R / W 1 2 により送信された変調波を、アンテナ 5 3 を介して受信する。コンデンサ 5 2 は、アンテナ 5 3 とともに L C 回路を構成し、所定の周波数（キャリア周波数）の電磁波に同調（共振）するようになされている。

【 0 0 4 0 】

I C 5 1 は、R F インタフェース部 6 1、BPSK復調部、PLL(Phase Locked Loop)部 6 3 10 、演算部 6 4 、ROM 6 5 、EEPROM(Electrically Erasable and Programmable ROM) 6 6 、RAM 6 7 、およびBPSK変調部 6 8 から構成される。

【 0 0 4 1 】

R F インタフェース部 6 1 は、さらに、ASK復調部 8 1 、電圧レギュレータ 8 2 、発振部 8 3 、およびASK変調部 8 4 に細分化されている。

【 0 0 4 2 】

R F インタフェース部 6 1 は、ASK復調部 8 1 がアンテナ 5 3 を介して受信した変調波（ASK変調波）を検波して復調し、復調後のデータをBPSK復調部 6 2 およびPLL部 6 3 に出力する。また、R F インタフェース部 6 2 は、電圧レギュレータ 8 2 によってASK復調部 8 1 が検波した信号を安定化し、その結果得られる直流電力を I C 5 1 の各部に駆動電力として供給する。さらに、R F インタフェース部 6 1 は、発振部 8 3 でデータのクロック周波数と同一の周波数の信号を発振し、その信号をPLL部 6 3 に出力する。 20

【 0 0 4 3 】

そして、R F インタフェース部 6 1 のASK変調部 8 4 は、演算部 6 4 より供給されたデータに対応して、I C カード 1 3 の電源としてのアンテナ 5 3 の負荷を変動させる（例えば、データに対応して所定のスイッチング素子をオン／オフさせ、スイッチング素子がオン状態であるときだけ所定の負荷をアンテナ 5 3 に並列に接続させる）ことにより、アンテナ 5 3 を介して受信している変調波をASK変調し、アンテナ 5 3 を介して R / W 1 2 に送信する。

【 0 0 4 4 】

PLL部 6 3 は、ASK復調部 8 1 より供給されたデータより、そのデータに同期したクロック信号を生成し、そのクロック信号をBPSK復調部 6 2 およびBPSK変調部 6 8 に出力する。BPSK復調部 6 2 は、ASK復調部 8 1 で復調されたデータがBPSK変調されている場合、PLL部 6 3 より供給されたクロック信号に従って、そのデータの復調（マンチェスタコードのデコード）を行い、復調したデータを演算部 6 4 に出力する。 30

【 0 0 4 5 】

演算部 6 4 は、シーケンス部 9 1 、暗号／復号部 9 2 、およびパリティ演算部 9 3 から成り、BPSK復調部 6 2 より供給されたデータが暗号化されている場合、そのデータを暗号／復号部 9 2 で復号化した後、その復号結果のデータをシーケンサ 9 1 で処理する。BPSK復調部 6 2 より供給されたデータが暗号化されていない場合、BPSK復調部 6 2 より供給されたデータは、暗号／復号部 9 2 を介さず、シーケンサ 9 1 に直接供給される。 40

【 0 0 4 6 】

シーケンサ 9 1 は、そこに供給されるコマンドとしてのデータに対応する処理を行う。すなわち、例えば、シーケンサ 9 1 は、EEPROM 6 6 に対するデータの書き込みや読み出しなどの処理を行う。さらに、シーケンサ 9 1 は、コマンドとしてのデータに対応する処理を行った後、その処理に対応する応答データ（R / W 1 2 に送信するデータ）を生成してBPSK変調部 6 8 に出力する。

【 0 0 4 7 】

パリティ演算部 9 3 は、EEPROM 6 6 に記憶されるデータや、EEPROM 6 6 に記憶されているデータから、パリティとして、リードソロモン符号を算出する。 50

【0048】

BPSK変調部68は、演算部64より供給されたデータをBPSK変調し、変調後のデータをR F インタフェース部61のASK変調部84に出力する。

【0049】

ROM65は、シーケンサ91が行うべき処理プログラムその他の必要なデータを記憶している。RAM67は、シーケンサ91が処理を行うとき、処理の途中のデータなどを、一時的に記憶する。EEPROM66は、不揮発性のメモリであり、I Cカード13がR / W12との通信を終了し、電力供給が停止した後も、記憶しているデータを消失することなく保持するようになされている。

【0050】

10

[メモリコラプションに対する対処方法]

次に、I Cカード13のシーケンサ91による、EEPROM66に対するデータの読み書き処理について説明するが、その前に、その前段階の準備として、メモリに記憶されている情報が破壊されてしまうことを意味するメモリコラプション(Memory Corruption)に対する対処方法について説明する。

【0051】

まず、メモリコラプションに対する対処方法の基本原理について説明する。

【0052】

例えば、メモリ(EEPROM66に相当)に対して、情報の読み書きが所定のブロック単位で行われるとし、あるブロックB1に記憶されたデータを更新する(ブロックB1に記憶されているデータを、他のデータに書き換える)ことを考える。

20

【0053】

この場合、ブロックB1自体に、新たなデータを書き込むと、ブロックB1に既に記憶されているデータに上書きすることになり、この書き込みの最中に、I Cカードへの電力の供給が不足したりした場合、新たなデータは完全には書き込まれないことになる。さらに、ブロックB1に記憶されていたデータは破壊されることになり、メモリコラプションが生じてしまう。

【0054】

このような事態の発生を防止するためには、ブロックB1に書き込むべき新たなデータを、そのブロックB1とは異なるブロックB2に書き込むようにすればよい。これにより、ブロックB2への書き込みの最中にメモリコラプションが生じた場合には、新たなデータの書き込みは完全に行われず、その有効性は保証されないものの、少なくとも、ブロックB1に記憶されていたデータの破壊は防止することができる。そして、ブロックB2への書き込みが終了した後、ブロックB1に記憶されているデータを消去する。

30

【0055】

なお、さらに新たなデータが供給されたときには、そのデータは、ブロックB2とは異なるブロックである、例えばブロックB1に書き込まれる。このように、新たなデータが供給された場合には、前回書き込まれたデータが記憶されているブロック以外のブロックに、その新たなデータを書き込むようにする。これにより、少なくとも、新たなデータの書き込みによって、前回書き込まれたデータが破壊されないようにし、これにより、I Cカードが最悪の場合でも、使用不可にならないようにする。

40

【0056】

次に、図4を参照して、以上のような基本原理を適用したメモリの読み書き方法について説明する。

【0057】

図4においては説明の便宜上、1ブロックが、1バイト(8ビット)の最新性情報、8バイトのデータ、および2バイトの有効性情報が、その先頭から順次配置された11バイトで構成されているものとする。ただし、1ブロックのビット数、さらには、最新性情報、データ、および有効性情報に割り当てるビット数は、上述したものに特に限定されるものではない。

50

【0058】

最新性情報は、そのブロックの記憶内容の新しさを示すもので、例えば、絶対的な日付および時刻や、シーケンシャルな番号その他を用いることができる。すなわち、絶対的な日付および時刻を用いる場合には、データが記憶された日付および時刻を最新性情報として記憶するようすれば、その最新性情報により、データの記憶がなされた最新のブロックを検出することができる。

【0059】

また、シーケンシャルな番号を用いる場合には、データの書き込みが行われるたびに、例えばインクリメントされる番号を最新性情報として記憶するようすれば、その値の最も大きいブロックが、データの書き込みがなされた最新のものということになる。

10

【0060】

なお、ここでは、少なくとも2つのブロックのうちのいずれかが新しいかを認識できれば充分であり、従って、最新性情報は、少なくとも3つの状態を表すことができればよい。すなわち、例えば、いま、3つの状態を表すことのできる最新性情報として、0, 1, 2の3値を用いることになると、2つのブロックのうちのいずれかに書き込みが行われる毎に、0, 1, 2, 0, …を、その書き込みが行われたブロックの最新性情報として書き込むようにすればよい。

【0061】

この場合、2つのブロックの最新性情報が0と1ならば1のブロックの方が新しく、1と2ならば2のブロックの方が新しく、2と0ならば0のブロックの方が新しいということになる。このように、最新性情報として、0, 1, 2の3値を用いる場合には、そのビット数は2ビットで済む。その他、例えば、最新性情報としては、1ビットのフラグを3個用い、ブロックの書き込みが行われる毎に、ビットをたてるフラグを順次変化させることとしてもよい。

20

【0062】

以下、最新性情報として、上述した0, 1, 2の3値を用いることとする。

【0063】

有効性情報は、そのブロックの最新性情報およびデータの書き込みが正常に終了したかどうかを示す、いわば、最新性情報およびデータの有効性（有効であるのか無効であるのか）を示すもので、例えば、RS（リードソロモン）符号などを用いることができる。

30

【0064】

なお、有効性情報は、RS符号などの誤り訂正符号に限定されるものではない。すなわち、有効性情報は、上述のように最新性情報およびデータが有効であるのか、または無効であるのかを表すことができればよい。従って、有効性情報は、例えば、最新性情報およびデータを書き込みながら計算されるパリティであっても、また、例えば、最新性情報およびデータが正常に書き込まれた後に付加される1ビットのフラグなどであってもよい。

【0065】

図4では、メモリが、第1および第2の領域としての2つのブロックを有しており、同図Aは、第1の領域の8バイトのデータとして01, 02, 03, 04, 05, 06, 07, 08（各値はそれぞれ16進数とする）が既に記憶されており、第2の領域は、全てのデータが消去されている様子を示している。

40

【0066】

この状態において、新たに書き込むべきデータ11乃至18が供給されると、まず、第1および第2の領域としての2つのブロックの有効性が、それぞれの有効性情報に基づいて判定される。なお、ここでは、第1の領域としてのブロックのみが有効であると判定される。

【0067】

新たなデータの書き込みを行う場合、上述したように、前回書き込まれたデータを破壊しないようにするために、現状でデータが記録されていないブロックに、新たなデータの書き込みを行う。従って、同図Bに示すように、新たなデータ11乃至18は、第2の領域と

50

してのブロックに書き込まれる。データ11乃至18の書き込みの終了後は、まず、その書き込みが行われたブロックの最新性情報を更新する。すなわち、それまで最新のブロックであった、第1の領域としてのブロックの最新性情報が01であったので、いま書き込みが行われた第2の領域としてのブロックの最新性情報として、第1の領域としてのブロックの最新性情報01に1を加算した値02が書き込まれ、さらに、最新性情報02の有効性を表す有効性情報の書き込みが行われる。

【0068】

そして、第2の領域としてのブロックへの書き込みが全て終了した後、同図Cに示すように、第1の領域としてのブロックに記録されている全てのデータが消去される。

【0069】

ところで、図4の第1および第2の領域としての2つのブロックは、外部からは、同一の論理的なブロックである論理ブロックとして認識されるようになされている。すなわち、第1および第2の領域としてのブロックは、物理的に存在するブロックである物理ブロックであり、各物理ブロックには、その識別のために、ユニークな番号である物理ブロック番号が割り当てられている。そして、第1および第2の領域としての物理ブロックの物理ブロック番号には、いずれにも、論理ブロックを識別するための論理ブロック番号として、同一の番号が対応付けられており、これにより、第1および第2の領域としての物理ブロックは、外部からは1つの論理ブロックとして認識されるようになされている。

【0070】

つまり、図4において、第1および第2の領域としての物理ブロックの物理ブロック番号は異なるが、いずれにも、同一の論理ブロック番号が割り当てられており、外部からは、そのような論理ブロック番号を有する論理ブロックにアクセスの要求がなされる。そして、メモリ側では、論理ブロックへのアクセスの要求があると、それに対応する物理ブロック番号のブロック、すなわち、第1および第2の領域に対してデータの読み出し処理、またはデータの書き込み処理が行われる。

【0071】

なお、以下では、物理ブロック番号および論理ブロック番号のいずれも1バイトで表されるものとする。

【0072】

次に、上述の場合においては、1の物理ブロックのデータの書き込みの終了後に、その物理ブロックの最新性情報と有効性情報が更新されるため、関連する複数ブロック分のデータを、複数の物理ブロックに書き込む場合においては、新旧混在したデータが生じことがある。

【0073】

すなわち、外部から見た場合において、複数の論理ブロックに、複数ブロック分のデータを書き込んでいる最中にメモリコラプションが生じると、メモリコラプションが生じる前に書き込みが完了した論理ブロックには、新しいデータが存在することになるが、メモリコラプションが生じた最中に書き込みを行っている論理ブロック以降には、古いデータが存在することになる。

【0074】

関連する複数ブロック分のデータが一括で意味のあるものであり、従って、それ全体で有用なものである場合においては、上述のように、その中に、新しいものと古いものとが混在すると、その整合性がとれなくなる。すなわち、関連する複数ブロック分のデータが、例えば、前述したように、乗車した駅と時刻、降車した駅と時刻、乗降車した駅間の運賃などである場合において、乗車した駅については新しいデータが書き込まれ、乗車した時刻については古いデータのままでは、その整合性がとれず、意味のないものとなる。

【0075】

従って、複数の論理ブロックに、関連する複数ブロック分のデータが書き込まれる場合において、その複数ブロック分のデータの書き込みが終了するまでにメモリコラプションが生じたときには、外部からは、論理ブロックにその関連する複数ブロック分の古いデータ

10

20

30

40

50

が存在するように認識させる必要がある。

【0076】

[シングルポインタ方式の説明]

そこで、メモリを、例えば、図5に示すように構成する。すなわち、メモリを、データを記憶するデータ領域としての物理ブロックと、データ領域を構成する物理ブロックの物理ブロック番号を記憶する第1および第2の領域としての物理ブロックとで構成する。

【0077】

なお、図5においても、図4における場合と同様に、1個の物理ブロック（同図における横1列）は、11バイトで構成されている。

【0078】

データ領域を構成する各物理ブロックは、11バイト単位でデータを記憶するようになされており、同図の場合、データ領域は、物理ブロック番号#00乃至#07それぞれが割り当てられた8個の物理ブロックで構成されている（#は、物理ブロック番号であることを表す）。

【0079】

また、同図の場合、第1の領域の物理ブロック番号には#FEが割り当てられており、第2の領域の物理ブロック番号には#FFが割り当てられている。

【0080】

なお、同図のように構成されたメモリは、外部からは4個の論理ブロックが認識されるようになされており、その4個の論理ブロックそれぞれには、論理ブロック番号として%00乃至%03が割り当てられている。なお、同図の場合、論理ブロック番号%00乃至%03には、物理ブロック番号#00乃至#03が対応付けられている。

【0081】

同図に示された第1および第2の領域としての物理ブロックは、図4の場合と同様、その先頭の1バイトが最新性情報に、終わりの2バイトが有効性情報に、それぞれ割り当てられている。ただし、最新性情報と有効性情報との間の8バイトには、データ領域の物理ブロックの物理ブロック番号、すなわち、データ領域を構成する物理ブロックへのポインタが配置される。

【0082】

データ領域を構成する物理ブロックへのポインタが配置される8バイトのうち、前半の4バイトには、データ領域の対象ブロックの物理ブロック番号が配置され、後半の4バイトには、データ領域の更新ブロックの物理ブロック番号が配置される。

【0083】

ここで、対象ブロックとは、情報を書き込む場合に、本来、その書き込みの対象となるべき物理ブロックを意味する。すなわち、例えば、上述した基本原理においては、ブロックB1に記憶されたデータを新しいデータに書き換える場合、その新しいデータをブロック1に書き込まず、ブロックB2に書き込むが、この場合において、本来、新しいデータを書き込むべきブロックB1が対象ブロックに相当する。

【0084】

また、更新ブロックとは、ある物理ブロックに情報を書き込み、その記憶内容を更新するときに用いられる更新用の物理ブロックを意味する。すなわち、例えば、上述した基本原理においては、ブロックB1に記憶されたデータを新しいデータに書き換える場合、その新しいデータをブロックB1に書き込まず、ブロックB2に書き込むが、この場合において、新しいデータを書き込むブロックB2が更新ブロックに相当する。

【0085】

第1および第2の領域における最新性情報と有効性情報との間の8バイトのうちの前半の4バイトに配置される4個の物理ブロック番号は、論理ブロック番号と対応付けられるようになされている。すなわち、例えば、その1乃至4バイト目に配置される物理ブロック番号は、それぞれ論理ブロック番号%00乃至%03に対応付けられている。

【0086】

10

20

30

40

50

図5に示されたように構成されるメモリにおいては、関連する4ブロック分のデータが物理ブロック#00（物理ブロック番号が#00の物理ブロック）乃至#03に記憶されている。そして、その物理ブロック番号#00乃至#03が、第1の領域における最新性情報と有効性情報との間の8バイトのうちの前半の4バイト（以下、適宜、データ領域の対象ブロックへのポインタの欄と称する）に、その順番で配置されている。

【0087】

従って、例えば、第1の領域における、データ領域の対象ブロックへのポインタの欄に注目した場合、論理ブロック番号%00乃至%03には物理ブロック番号#00乃至03が対応付けられている。

【0088】

ここで、第1の領域に注目した場合においては、論理ブロック%00（論理ブロック番号が%00の論理ブロック）乃至%03に対応付けられた物理ブロック#00乃至#03のうちのいずれかが、アクセスの対象となるべきブロック、すなわち、対象ブロックとなる。

【0089】

更新ブロックには、その時点で対象ブロックではないデータ領域の物理ブロック、すなわち、空き領域となっている物理ブロックがなり得る。従って、図5において、第1の領域に注目した場合には、物理ブロック#04乃至#07が更新ブロックとなり得るので、その第1の領域における最新性情報と有効性情報との間の8バイトのうちの後半の4バイト（以下、適宜、データ領域の更新ブロックへのポインタの欄と称する）には、それらの更新ブロックの物理ブロック番号#04乃至#07が配置されている。

【0090】

ここで、例えば、第1または第2の領域のうちの第1の領域のみが有効であるとして、外部から、論理ブロック番号%00と%02に対応する論理ブロックそれぞれに、関連する2ブロック分のデータの書き込みの要求があったとする。

【0091】

この場合、有効である第1の領域のうちの、データ領域の対象ブロックへのポインタの欄を参照することにより、論理ブロック番号%00と%02に対応する物理ブロック番号として#00と#02が認識される。そして、物理ブロック#00および#02が、関連する2ブロック分のデータそれぞれを書き込むべき対象ブロックとして認識される。

【0092】

さらに、第1の領域のうちの、データ領域の更新ブロックへのポインタの欄を参照することで、全ての更新ブロックの物理ブロック番号が認識される。いまの場合、物理ブロック番号#04, #05, #06, #07が認識される。そして、更新ブロックとして認識された全ての物理ブロック番号にそれぞれ対応する物理ブロック（いまの場合、物理ブロック#04, #05, #06, #07）が一括して消去される。なお、複数の物理ブロックを一括して消去するための要する時間は、1個の物理ブロックを消去するための要する時間と同等である。

【0093】

この後、対象ブロックの数（いまの場合、2）と同数の更新ブロックの物理ブロック番号が、例えば、データ領域の更新ブロックへのポインタの欄の左から配置されているものから順番に認識され、本来ならば、対象ブロックである物理ブロック#00または#02に書き込むべき2ブロック分のデータが、図6に示すように、更新ブロックである物理ブロック#04または#05それぞれに書き込まれる。

【0094】

このように、全ての更新ブロックとされた物理ブロックを一括して消去することにより、物理ブロックにデータを書き込み毎に、その都度、物理ブロックのデータを消去する場合に比較して、処理時間を短縮することができる。

【0095】

ここで、図6は、論理ブロック番号%00と%02に対応する物理ブロック番号#00

10

20

30

40

50

と # 0 2 に書き込むべき 2 つのデータが、それぞれ 4 0 , 4 1 , . . . , 4 A と 5 0 , 5 1 , . . . , 5 A であったとして、これらが、更新ブロックである物理ブロック # 0 4 と # 0 5 それぞれに書き込まれた状態を示している。

【 0 0 9 6 】

物理ブロック # 0 4 と # 0 5 それぞれにデータ書き込まれた後、更新ブロックであった物理ブロックのブロック番号 # 0 4 と # 0 5 とが、これまで無効であった第 2 の領域におけるデータ領域の対象ブロックへのポインタの欄に書き込まれる。すなわち、第 1 の領域におけるデータ領域の対象ブロックへのポインタの欄の記載 # 0 0 , # 0 1 , # 0 2 , # 0 3 が第 2 の領域にコピーされ、このうち、対象ブロックの物理ブロック番号 # 0 0 と # 0 2 が、同図に示すように、実際にデータが書き込まれた更新ブロックの物理ブロック番号 # 0 4 と # 0 5 にそれぞれ書き換えられる。10

【 0 0 9 7 】

この結果、第 2 の領域に注目した場合、論理ブロック番号 % 0 0 が物理ブロック番号 # 0 4 に対応付けられ、論理ブロック番号 % 0 2 が物理ブロック番号 # 0 2 に対応付けられたことがわかる。

【 0 0 9 8 】

また、データを書き込むべきであった物理ブロック # 0 0 と # 0 2 は更新ブロックとされ、その物理ブロック番号 # 0 0 と # 0 2 が、第 2 の領域におけるデータ領域の更新ブロックへのポインタの欄に書き込まれる。すなわち、第 1 の領域におけるデータ領域の更新ブロックへのポインタの欄の記載 # 0 4 , # 0 5 , # 0 6 , # 0 7 が、第 2 の領域にコピーされ、このうち、データが書き込まれた物理ブロックの物理ブロック番号 # 0 4 と # 0 5 が、同図に示すように、対象ブロックの物理ブロック番号 # 0 0 と # 0 2 にそれぞれ書き換えられる。20

【 0 0 9 9 】

この結果、第 2 の領域に注目した場合、物理ブロック # 0 0 および # 0 2 が新たに更新ブロックとされたことがわかる。

【 0 1 0 0 】

以上のようにして、第 2 の領域の、データ領域の対象ブロックへのポインタと、データ領域の更新ブロックへのポインタが更新された後、その第 2 の領域における最新性情報と有効性情報とが順次書き込まれ（書き換えられ）、その後、第 1 の領域のデータが全て消去される。30

【 0 1 0 1 】

従って、この場合、第 2 の領域について、データ領域の対象ブロックへのポインタの欄の書き込みが終了しないと、有効である旨の有効性情報は書き込まれない。すなわち、上述の場合においては、2 つの論理ブロック % 0 0 と % 0 2 へのデータの書き込みが両方とも終了しなければ、第 2 の領域には、有効である旨の有効性情報は書き込まれない。

【 0 1 0 2 】

その結果、2 つの論理ブロック % 0 0 と % 0 2 へのデータの書き込みが行われている間にメモリコラプションが生じた場合には、第 1 の領域のみが有効な状態となったままになり、論理ブロック % 0 0 と % 0 2 からは、有効である第 1 の領域を参照することで、その論理ブロック % 0 0 と % 0 2 にそれぞれ対応付けられている物理ブロック # 0 0 と # 0 2 に記憶されている古いデータが読み出されることになる。40

【 0 1 0 3 】

一方、2 つの論理ブロック % 0 0 と % 0 2 へのデータの書き込みが終了し、第 2 の領域における最新性情報と有効性情報の更新が行われ、第 1 の領域のデータが全て消去された場合には、論理ブロック % 0 0 と % 0 2 からは、唯一有効である第 2 の領域を参照することで、その論理ブロック % 0 0 と % 0 2 にそれぞれ対応付けられている物理ブロック # 0 4 と # 0 5 に記憶されている新しいデータが読み出されることになる。

【 0 1 0 4 】

従って、この場合、論理ブロック % 0 0 と % 0 2 に記憶される、複数のデータとしての50

2つデータが関連性を有するものであるときには、メモリコラプションが生じても、個々のデータの整合性とともに、その2つのデータの間の整合性を維持することが可能となる。

【0105】

なお、図5および図6を参照して上述したメモリに対する読み書き方式は、データ領域への1段階のポインタを利用したものなので、以下、適宜、シングルポインタ方式と称する。

【0106】

ところで、図5および図6に示されたシングルポインタ方式では、データ領域の対象ブロックへのポインタの欄として、4つの物理ブロック番号を記載できるようになされている。また、データ領域の更新ブロックへのポインタの欄も、データ領域の対象ブロックへのポインタの欄と同様に、4つの物理ブロック番号を記載できるようになされている。これにより、最大4ブロック分のデータの整合性を維持することができる。10

【0107】

しかしながら、逆に言えば、図5および図6に示されたシングルポインタ方式では、4ブロックよりも多いブロック分の整合性の維持を図ることは困難である。

【0108】

すなわち、図5および図6に示されたシングルポインタ方式では、データの整合性を維持することができるブロックの数が、データ領域の対象ブロックへのポインタのブロックの数に制限されてしまう。20

【0109】

[ダブルポインタ方式の説明]

そこで、データの整合性を維持することができるブロック数の制限を解除するため、メモリの構成を、例えば図7に示すように拡張する。

【0110】

すなわち、ここでは、メモリを、データを記憶するデータ領域としての物理ブロック#00乃至#17と、データ領域を構成する物理ブロックへのポインタとしての、そのブロック番号を記憶するポインタ領域としての物理ブロック#18乃至#1Dと、ポインタ領域を構成する物理ブロックへのポインタとしての、そのブロック番号を記憶する第1および第2の領域としての物理ブロック#1Eおよび#1Fとで構成する。30

【0111】

なお、図7では、図示の都合上、1つの物理ブロックを8バイトで構成している。

【0112】

データ領域を構成する各物理ブロックは、8バイト単位でデータを記憶するようになされており、図7においては、データ領域は、物理ブロック番号として#00乃至#17それぞれが割り当てられた24個の物理ブロックで構成されている。

【0113】

なお、外部からは、16個の論理ブロックが認識されるようになされており、その16個の論理ブロックそれぞれには、論理ブロック番号として%00乃至%0Fが割り当てられている。40

【0114】

ただし、説明の都合上、データ領域を構成する物理ブロックの数(24個)と同数だけ論理ブロックが存在するものとし、外部から論理ブロックとして認識される物理ブロック以外の8個の物理ブロックそれぞれに、論理ブロック番号として%10乃至%17を仮に割り当てておく。図7の例では、論理ブロック番号%00乃至%17と、物理ブロック番号#00乃至#17とがそれぞれ対応付けられている。

【0115】

ポインタ領域(ブロック番号領域)を構成する各物理ブロックは、8バイト単位でデータ領域を構成する物理ブロックへのポインタとして、その物理ブロック番号を記憶する。図7の場合、ポインタ領域は、物理ブロック番号として#18乃至#1Dそれぞれが割り50

当てられた 6 個の物理ブロックで構成されている。ここでは、ポインタ領域を構成する各物理ブロックは、8 バイト単位で物理ブロック番号を記憶するようになされており、物理ブロック番号は 1 バイトで表されるので、1 個の物理ブロックには、8 個の物理ブロック番号が記憶される。

【 0 1 1 6 】

ポインタ領域の各物理ブロックに配置される 8 個の物理ブロック番号は、論理ブロック番号と対応付けられる。すなわち、その 1 乃至 8 バイト目に配置される物理ブロック番号は、ある論理ブロック番号（後述するように、% 0 0 , % 0 8 、または% 1 0 ）を基準として 0 乃至 7 だけオフセットした論理ブロック番号に対応付けられる。

【 0 1 1 7 】

第 1 および第 2 の領域としてのブロックは、その先頭の 1 バイトが最新性情報に割り当てられ、終わりの 1 バイトが有効性情報に割り当てられている。そして、最新性情報と有効性情報との間の 6 バイトには、ポインタ領域の物理ブロックへのポインタとしての、その物理ブロック番号が配置されている。

【 0 1 1 8 】

第 1 および第 2 の領域における最新性情報と有効性情報との間の 6 バイトのうち、前半の 3 バイト（以下、適宜、ポインタ領域の対象ブロックへのポインタの欄と称する）には、ポインタ領域の対象ブロックの物理ブロック番号が配置され、その後半の 3 バイト（以下、適宜、ポインタ領域の更新ブロックへのポインタの欄と称する）には、ポインタ領域の更新ブロックの物理ブロック番号が配置される。

【 0 1 1 9 】

第 1 および第 2 の領域におけるポインタ領域の対象ブロックへのポインタの欄に配置される 3 つの物理ブロック番号に対応するポインタ領域の物理ブロックが記憶している物理ブロック番号は、論理ブロック番号と対応付けられている。すなわち、例えば、その 1 乃至 3 バイト目に配置される物理ブロック番号に対応するポインタ領域の物理ブロックが記憶している物理ブロック番号は、それぞれ論理ブロック番号% 0 0 乃至% 0 7 、% 0 8 乃至% 0 F 、または% 1 0 乃至% 1 7 に対応付けられている。

【 0 1 2 0 】

具体的には、図 7 の場合、第 1 の領域においては、ポインタ領域の対象ブロックへのポインタの欄における 1 バイト目の物理ブロック番号# 1 8 が指しているポインタ領域の物理ブロック# 1 8 の 1 乃至 8 バイト目に配置された物理ブロック番号# 0 0 乃至# 0 7 は、第 1 の領域におけるポインタ領域の対象ブロックへのポインタの欄における 1 バイト目に対応する論理ブロック番号% 0 0 乃至% 0 7 にそれぞれ対応付けられる。

【 0 1 2 1 】

また、ポインタ領域の対象ブロックへのポインタの欄における 2 バイト目の物理ブロック番号# 1 9 が指しているポインタ領域の物理ブロック# 1 9 の 1 乃至 8 バイト目に配置された物理ブロック番号# 0 8 乃至# 0 F は、第 1 の領域におけるポインタ領域の対象ブロックへのポインタの欄における 2 バイト目に対応付けられた論理ブロック番号% 0 8 乃至% 0 F にそれぞれ対応付けられる。

【 0 1 2 2 】

同様に、ポインタ領域の対象ブロックへのポインタの欄における 3 バイト目の物理ブロック番号# 1 A が指すポインタ領域の物理ブロック# 1 A の 1 乃至 8 バイト目に配置された物理ブロック番号# 1 0 乃至# 1 7 は、論理ブロック番号% 1 0 乃至% 1 7 にそれぞれ対応付けられる。

【 0 1 2 3 】

以上のように構成されるメモリにおいて、例えば、第 1 の領域に注目した場合には、ポインタ領域の対象ブロックへのポインタの欄に記述された物理ブロック番号に対応する物理ブロックが、アクセス対象となるべきポインタ領域の対象ブロックとなる。

【 0 1 2 4 】

従って、図 7 の場合、ポインタ領域の物理ブロック# 1 8 乃至# 1 A がポインタ領域の

10

20

30

40

50

対象ブロックとなっており、残りの物理ブロック # 1 B 乃至 # 1 D が、ポインタ領域の更新のための更新ブロックとなっている。これにより、その物理ブロック番号 # 1 B 乃至 # 1 D が、第 1 の領域におけるポインタ領域の更新ブロックへのポインタの欄に記述されている。

【 0 1 2 5 】

さらに、ポインタ領域の対象ブロックに記述された物理ブロック番号に対応する、データ領域の物理ブロックのうち、例えば、論理ブロック % 0 0 乃至 % 0 F に対応するものがデータ領域の対象ブロックとされている。いまの場合、論理ブロック % 0 0 乃至 % 0 F に対応する物理ブロック # 0 0 乃至 # 0 F がデータ領域の対象ブロックとなり、データ領域の残りの物理ブロック（論理ブロック % 1 0 乃至 % 1 7 に対応する物理ブロック）# 1 0 乃至 # 1 7 が、データ領域の更新のための更新ブロックとなっている。10

【 0 1 2 6 】

ここで、上述したように、ポインタ領域の更新ブロックは、その物理ブロック番号が、第 1 または第 2 の領域の、ポインタ領域の更新ブロックへのポインタの欄に記述されているので、その欄を参照することで認識することができる。一方、データ領域の更新ブロックは、論理ブロック % 1 0 乃至 % 1 7 に対応する物理ブロックであるから、ポインタ領域の対象ブロックへのポインタの欄の 3 バイト目に記述されている物理ブロック番号に対応するポインタ領域の物理ブロックを参照することで、そこに記述されている物理ブロック番号から認識することができる。

【 0 1 2 7 】

いま、第 1 または第 2 の領域のうち、第 1 の領域のみが有効であると仮定し、外部から、例えば、論理ブロック番号 % 0 0 と % 0 2 に対応する物理ブロックそれぞれに、関連する 2 ブロック分のデータの書き込みの要求があったとする。20

【 0 1 2 8 】

この場合、有効である第 1 の領域の、ポインタ領域の対象ブロックへのポインタの欄を参照し、さらに、そこに記述されている物理ブロック番号に対応する物理ブロックを参照することで、関連する 2 ブロック分のデータを書き込むべきデータ領域の物理ブロックである対象ブロックが認識される。この後、データ領域の更新ブロックが認識される。

【 0 1 2 9 】

具体的には、第 1 の領域におけるポインタ領域の対象ブロックへのポインタの欄の 1 バイト目を参照することで、論理ブロック % 0 0 と % 0 2 に対応するデータ領域の物理ブロックの物理ブロック番号が記述された、ポインタ領域の物理ブロック # 1 8 を認識することができる。そして、その物理ブロック # 1 8 の 1 バイト目と 3 バイト目を参照することで、論理ブロック % 0 0 と % 0 2 に対応するデータ領域の物理ブロック（対象ブロック）の物理ブロック番号 # 0 0 と # 0 2 をそれぞれ認識することができる。30

【 0 1 3 0 】

さらに、第 1 の領域におけるポインタ領域の対象ブロックへのポインタの欄の 3 バイト目を参照することで、データ領域の更新ブロックとなる論理ブロック % 1 0 乃至 % 1 7 に対応する物理ブロックの物理ブロック番号が記述された、ポインタ領域の物理ブロック # 1 A を認識することができる。40

【 0 1 3 1 】

そして、その物理ブロック # 1 A を参照することで、データ領域の更新ブロックとなっている全ての物理ブロックの物理ブロック番号 # 1 0 乃至 # 1 7 を認識することができ、更新ブロックとされた全ての物理ブロック # 1 0 乃至 # 1 7 が一括して消去される。なお、複数の物理ブロックを一括して消去するための要する時間は、1 個の物理ブロックを消去するための要する時間と同等である。

【 0 1 3 2 】

この後、一括消去されたデータ領域の全ての更新ブロックのうち、データの書き込み対象である、データ領域の対象ブロックの数と同数の物理ブロック番号が、物理ブロック # 1 A の、例えば左から順次参照されて、データを書き込む更新ブロックとして認識され。50

まの場合、物理ブロック番号 # 1 0 と # 1 1 に対応する、データ領域の物理ブロック # 1 0 および # 1 1 がデータを書き込む更新ブロックとされる。

【 0 1 3 3 】

そして、本来ならば、論理ブロック % 0 0 と % 0 2 に対応する、データ領域の対象ブロックである物理ブロック # 0 0 と # 0 2 に書き込むべき 2 つのデータが、図 8 に示すように、データ領域の更新ブロックである物理ブロック # 1 0 と # 1 1 に書き込まれる。

【 0 1 3 4 】

すなわち、図 8 は、論理ブロック番号 % 0 0 と % 0 2 に対応する物理ブロック番号 # 0 0 と # 0 2 に書き込むべき 2 つのデータが、0 0 , 0 0 , 0 0 , 0 0 , 0 0 , 0 0 , 0 0 と 0 2 , 0 2 , 0 2 , 0 2 , 0 2 , 0 2 , 0 2 であったとして、それらが、データ領域の更新ブロックである物理ブロック # 1 0 と # 1 1 それぞれに書き込まれた状態を示している。10

【 0 1 3 5 】

以上のようにして、データ領域の更新ブロックである物理ブロック # 1 0 と # 1 1 にデータを書き込んだ後、その物理ブロック番号 # 1 0 と # 1 1 が、ポインタ領域の更新ブロックに書き込まれるとともに、本来ならばデータが書き込まれたはずであったデータ領域の物理ブロック（対象ブロック）の物理ブロック番号 # 0 0 および # 0 2 が、ポインタ領域の、他の更新ブロックに書き込まれる。

【 0 1 3 6 】

すなわち、まず、第 1 の領域の、ポインタ領域の更新ブロックへのポインタの欄を参照することで、ポインタ領域の更新ブロックが認識される。なお、ここでは、ポインタ領域の更新ブロックとして、ポインタ領域の対象ブロックのうち、上述のようなデータ領域への書き込みに伴って変更（更新）する必要のある対象ブロックの数と同数の物理ブロックが、ポインタ領域の更新ブロックへのポインタの欄を、例えば左から順次参照することで認識される。20

【 0 1 3 7 】

従って、いまの場合、本来、データが書き込まれるべき物理ブロック（データ領域の対象ブロック）の物理ブロック番号 # 0 0 と # 0 2 が記憶されている、ポインタ領域の物理ブロック # 1 8 、および実際にデータが書き込まれた物理ブロックの物理ブロック番号 # 1 0 と # 1 1 が記憶されている、ポインタ領域の物理ブロック # 1 A の 2 個の物理ブロックを変更する必要があるので、第 1 の領域の、ポインタ領域の更新ブロックへのポインタの欄が左から 2 バイト分だけ参照され、物理ブロック # 1 B と # 1 C が更新ブロックとして認識される。30

【 0 1 3 8 】

そして、認識されたポインタ領域の更新ブロック # 1 B または # 1 C のうち、一方の更新ブロック # 1 B には、データ領域の対象ブロックの物理ブロック番号 # 0 0 と # 0 2 が記憶されている、ポインタ領域の物理ブロック # 1 8 の記憶内容がコピーされ、そのうちの、データ領域の対象ブロックの物理ブロック番号 # 0 0 と # 0 2 が、実際にデータが書き込まれたデータ領域の更新ブロックの物理ブロック番号 # 1 0 と # 1 1 に書き換えられる。40

【 0 1 3 9 】

また、認識されたポインタ領域の更新ブロック # 1 B または # 1 C のうち、他方の更新ブロック # 1 C には、実際にデータが書き込まれた物理ブロックの物理ブロック番号 # 1 0 と # 1 1 が記憶されている、ポインタ領域の物理ブロック # 1 A の記憶内容がコピーされ、その物理ブロック番号 # 1 0 と # 1 1 が、本来、データが書き込まれるべき物理ブロックの物理ブロック番号（データ領域の対象ブロック）# 0 0 と # 0 2 に書き換えられる。

【 0 1 4 0 】

この後、物理ブロック番号が更新されたポインタ領域の更新ブロックであった物理ブロックのブロック番号 # 1 B と # 1 C が、有効でなかった第 2 の領域におけるポインタ領域50

の対象ブロックへのポインタの欄に書き込まれる。すなわち、第1の領域におけるポインタ領域の対象ブロックへのポインタの欄が、第2の領域にコピーされ、そのうちの、論理アドレス%00乃至%07に対応する1バイト目が、ポインタ領域の物理ブロックの物理ブロック番号#1Bに書き換えられるとともに、論理アドレス%10乃至%17に対応する3バイト目が、ポインタ領域の物理ブロックの物理ブロック番号#1Cに書き換えられる。

【0141】

この結果、第2の領域に注目すると、物理ブロック#1B, #19、および#1Cが、ポインタ領域の対象ブロックとなり、これにより、データ領域の更新ブロックであった物理ブロック#10と#11が、データ領域の対象ブロックであった物理ブロック#00と#02が対応付けられていた論理ブロック番号%00と%02に新たに対応付けられる。
さらに、データ領域の対象ブロックであった物理ブロック#00と#02は、論理ブロック%10と%11に対応付けられて更新ブロックとされる。

【0142】

すなわち、第2の領域の、ポインタ領域の対象ブロックへのポインタの欄の1バイト目を参照すると、論理ブロック%00と%02に対応するデータ領域の物理ブロックの物理ブロック番号が記述された、ポインタ領域の物理ブロックは、物理ブロック#1Bということになる。さらに、その物理ブロック#1Bの1バイト目と3バイト目を参照すると、論理ブロック%00と%02に対応するデータ領域の物理ブロックは、それぞれ物理ブロック#10と#11ということになる。

【0143】

また、第2の領域の、ポインタ領域の対象ブロックへのポインタの欄の3バイト目を参照すると、論理ブロック%10と%11に対応するデータ領域の物理ブロックの物理ブロック番号が記述された、ポインタ領域の物理ブロックは、物理ブロック#1Cということになり、さらに、その物理ブロック#1Cの1バイト目と2バイト目を参照すると、論理ブロック%10と%11に対応するデータ領域の物理ブロック、つまりデータ領域の更新ブロックは、物理ブロック#00と#02ということになる。

【0144】

さらに、ポインタ領域のうちの、本来、データが書き込まれるべき物理ブロックの物理ブロック番号#00と#02が記憶されていた物理ブロックの物理ブロック番号#18、および実際にデータが書き込まれた物理ブロックの物理ブロック番号#10と#11が記憶されていた物理ブロックの物理ブロック番号#1Aが、有効でなかった第2の領域におけるポインタ領域の更新ブロックへのポインタの欄に書き込まれる。

【0145】

すなわち、第1の領域におけるポインタ領域の更新ブロックへのポインタの欄が、第2の領域にコピーされ、そのうちの、記憶内容の更新がされたポインタ領域の物理ブロックの物理ブロック番号#1Bまたは#1Cが、図8に示すように、論理アドレス%00乃至%07に対応していた物理ブロック番号が記憶された物理ブロック番号#18、または論理アドレス%10乃至%17に対応していた物理ブロック番号が記憶された物理ブロック番号#1Aに、それぞれ書き換えられる。

【0146】

この結果、第2の領域に注目した場合には、物理ブロック#18, #1A、および#1Dが、ポインタ領域の更新ブロックとなる。

【0147】

以上のようにして、第2の領域を更新した後は、シングルポインタ方式における場合と同様に、その第2の領域における最新性情報と有効性情報とが順次書き込まれる。

【0148】

そして、その第2の領域における最新性情報と有効性情報とが書き込まれた後に、第1の領域のデータが消去される。

【0149】

10

20

30

40

50

従って、この場合、データ領域、さらにはポインタ領域の書き込みが終了しないと、有効である旨の有効性情報は書き込まれない。すなわち、上述の場合においては、2つの論理ブロック%00と%02へのデータの書き込みが終了しなければ、第2の領域には、有効である旨の有効性情報は書き込まれない。

【0150】

その結果、2つの論理ブロック%00と%02へのデータの書き込みが行われている間にメモリコラプションが生じた場合には、第1の領域のみが有効な状態となつたままになり、論理ブロック%00と%02からは、有効である第1の領域を参照することで、その論理ブロック%00と%02にそれぞれ対応付けられている物理ブロック#00と#02に記憶されている古いデータが読み出されることになる。

10

【0151】

一方、2つの論理ブロック%00と%02へのデータの書き込みが終了し、第2の領域における最新性情報と有効性情報の更新が行われた場合には、論理ブロック%00と%02からは、有効であつて、且つ、最新である第2の領域を参照することで、その論理ブロック%00と%02にそれぞれ対応付けられている物理ブロック#10と#11に記憶されている新しいデータが読み出されることになる。

【0152】

従つて、この場合、論理ブロック%00と%02に記憶される、複数のデータとしての2つデータが関連性を有するものであるときには、メモリコラプションが生じても、シングルポインタ方式における場合と同様に、その2つのデータの間の整合性を維持することが可能となる。

20

【0153】

さらに、第1および第2の領域におけるポインタ領域の対象ブロックへのポインタの欄には、ポインタ領域の物理ブロックの3ブロック分の物理ブロック番号を記憶させることができる。また、ポインタ領域の1個の物理ブロックには、データ領域の物理ブロックの8ブロック分の物理ブロック番号を記憶させることができる。

【0154】

したがつて、最大で、データ領域の24(=3×8)個の物理ブロックを管理することができ、そのうちの最大で半分の12ブロックをデータ領域の更新ブロックとすることができます。すなわち、シングルポインタ方式よりも多いブロック数のデータの整合性を維持することができる。

30

【0155】

なお、図7および図8では、24ブロックのうちの8ブロックを更新ブロックとしてあり、8ブロック分のデータの整合性を維持することができる。

【0156】

図7および図8を参照して上述したメモリに対する読み書き方式は、データ領域への、いわば2段階のポインタを利用したものであるので、以下、適宜、ダブルポインタ方式と称する。

【0157】

次に、ダブルポインタ方式によるデータの読み書き処理について、図9および図10を参照してさらに説明する。

40

【0158】

[ダブルポインタ方式の読み出し処理]

図9は、読み出し処理を説明するフローチャートである。

【0159】

ステップS61において、第1および第2の領域の有効性情報を参照することで、その有効性が判定される。ここで、第1の領域としてのブロックのみが有効であると判定された場合、処理はステップS62に進められる。反対に、第2の領域としてのブロックのみが有効であると判定された場合、処理はステップS65に進められる。なお、第1および第2の領域としてのブロックの両方が無効であると判定された場合、I Cカード2のハ-

50

ドウェアとしての寿命（致命的エラー）であるので、読み出しあは行われずに処理は終了される（エラー処理が行われる）。

【0160】

第1の領域としてのブロックのみが有効であって、処理がステップS62に進むと、第1の領域におけるポインタ領域の対象ブロックへのポインタの欄を参照することで、読み出すべきデータの論理ブロック番号に対応する物理ブロック番号が記憶されている、ポインタ領域の物理ブロックの物理ブロック番号（ポインタ）が認識される。

【0161】

例えは、図7で説明したようにメモリが構成される場合において、読み出すべきデータの論理ブロック番号が%00乃至%07のときには、ポインタ領域の対象ブロックへのポインタの欄の1バイト目に記憶されている物理ブロック番号#18が認識される。また例えは、読み出すべきデータの論理ブロック番号が%08乃至%0Fのときには、ポインタ領域の対象ブロックへのポインタの欄の2バイト目に記憶されている物理ブロック番号#19が認識される。10

【0162】

ステップS63においては、ステップS62で認識された物理ブロック番号に対応するポインタ領域の物理ブロックを参照することで、読み出すべきデータの論理ブロック番号に対応する物理ブロック番号が認識される。

【0163】

すなわち、読み出すべきデータの論理ブロック番号が、ある基準値（図7の場合においては、%00または%08）から0乃至7だけオフセットした値である場合には、ステップS62で認識された物理ブロック番号に対応する物理ブロックの左から、そのオフセット分だけずれた位置に記憶された物理ブロック番号が、読み出すべきデータの論理ブロック番号に対応する物理ブロック番号として認識される。20

【0164】

ステップS64においては、ステップS63で認識された物理ブロック番号に対応する物理ブロックからデータが読み出されて、この読み出し処理が終了される。

【0165】

一方、第2の領域としてのブロックのみが有効であって、処理がステップS65に進むと、第2の領域におけるポインタ領域の対象ブロックへのポインタの欄を参照することで、読み2出すべきデータの論理ブロック番号に対応する物理ブロック番号が記憶されている、ポインタ領域の物理ブロックの物理ブロック番号（ポインタ）が認識される。30

【0166】

例えは、図8で説明したようにメモリが構成される場合において、読み出すべきデータの論理ブロック番号が%00乃至%07のときには、ポインタ領域の対象ブロックへのポインタの欄の1バイト目に記憶されている物理ブロック番号#1Bが認識される。また例えは、読み出すべきデータの論理ブロック番号が%08乃至%0Fのときには、ポインタ領域の対象ブロックへのポインタの欄の2バイト目に記憶されている物理ブロック番号#19が認識される。

【0167】

ステップS66においては、ステップS65で認識された物理ブロック番号に対応するポインタ領域の物理ブロックを参照することで、読み出すべきデータの論理ブロック番号に対応する物理ブロック番号が認識される。40

【0168】

すなわち、読み出すべきデータの論理ブロック番号が、ある基準値（図8の場合においては、%00または%08）から0乃至7だけオフセットした値である場合には、ステップS65で認識された物理ブロック番号に対応する物理ブロックの左から、そのオフセット分だけずれた位置に記憶された物理ブロック番号が、読み出すべきデータの論理ブロック番号に対応する物理ブロック番号として認識される。

【0169】

ステップS67においては、ステップS66で認識された物理ブロック番号に対応する物理ブロックからデータが読み出されて、この読み出し処理が終了される。

【0170】

[ダブルポインタ方式の書き込み処理]

次に、図10は、書き込み処理を説明するフローチャートである。

【0171】

ステップS31において、第1および第2の領域の有効性情報を参照することで、その有効性が判定される。ここで、第1の領域としてのブロックのみが有効であると判定された場合、処理はステップS32に進められる。反対に、第2の領域としてのブロックのみが有効であると判定された場合、処理はステップS40に進められる。なお、第1および第2の領域としてのブロックの両方が無効であると判定された場合、I Cカード2のハードウェアとしての寿命（致命的エラー）であるので、書き込みは行われずに処理は終了される（エラー処理が行われる）。

【0172】

第1の領域のみが有効であって、処理がステップS32に進むと、有効ではない第2の領域のデータが消去されているか否かが判定される。消去されていないと判定された場合、処理はステップS50に進められ、有効ではない第2の領域のデータが消去された後、処理がステップS33に進められる。反対に、消去されていると判定された場合、ステップS50はスキップされて、処理はステップS33に進められる。

【0173】

ステップS33においては、第1の領域におけるポインタ領域の対象ブロックへのポインタの欄が参照され、更新ブロックとができるデータ領域の物理ブロックの物理ブロック番号が記憶されたポインタ領域の物理ブロックの物理ブロック番号が認識される。すなわち、例えば、図7に示したようにメモリが構成される場合においては、第1の領域におけるポインタ領域の対象ブロックへのポインタの欄の3バイト目に記憶された物理ブロック番号#1Aが認識される。

【0174】

ステップS34では、ステップS33で認識された物理ブロック番号に対応する物理ブロック（図7の場合、物理ブロック#1A）に記憶された、データ領域の全ての更新ブロックの物理ブロック番号が参照されて、データ領域の更新ブロックとして認識される。

【0175】

図7の場合、物理ブロック#1Aに記録されている、物理ブロック番号#1A, #11, #12, #13, #14, #15, #16, #17がデータ領域の更新ブロックとして認識される。

【0176】

ステップS35において、データ領域の更新ブロックとして認識された全ての物理ブロックにそれぞれ記録されているデータが一括して消去される。ここで、データを一括して消去することにより、物理ブロックにデータを書き込み毎に、その都度、物理ブロックのデータを消去する場合に比較して、処理時間を短縮することができる。

【0177】

この後、ステップS36において、外部などから供給された、書き込むべきデータが、データが一括消去された後のデータ領域の更新ブロックに書き込まれる。

【0178】

なお、外部から、複数の論理ブロックに書き込むデータが供給された場合には、ステップS34で認識された物理ブロックに対し、例えば、左から右方向に順次データを書き込むようにする。

【0179】

この後、処理はステップS37に進められ、データの書き込みの対象とした論理ブロックの論理ブロック番号が対応付けられていた物理ブロック番号を記憶していた物理ブロックが、その一部が更新された状態でポインタ領域の更新ブロックにコピーされる。

10

20

30

40

50

【0180】

すなわち、例えば、図7および図8で説明した場合においては、データの書き込みの対象とした論理ブロックの論理ブロック番号%00,%02が対応付けられていた物理ブロック番号#00,#02を記憶していた物理ブロック#18が、記憶している物理ブロック番号#00,#02がそれぞれ#10,#11に更新された状態で、ポインタ領域の更新ブロック#1Bにコピーされる。

【0181】

さらに、ステップS37では、データ領域の更新ブロックへのポインタ（データ領域の更新ブロックの物理ブロック番号）を記憶している物理ブロックが、その一部が更新された状態でポインタ領域の更新ブロックにコピーされる。

10

【0182】

すなわち、例えば、図7および図8で説明した場合においては、データ領域の更新ブロックの物理ブロック番号#10乃至#17を記憶している物理ブロック#1Aが、記憶している物理ブロック番号#10,#11がそれぞれ#00,#02に更新された状態で、ポインタ領域の更新ブロック#1Cにコピーされる。

【0183】

そして、ステップS38に進み、第1の領域におけるポインタ領域の対象ブロックへのポインタの欄、およびポインタ領域の更新ブロックへのポインタの欄が更新され、さらに、その最新性情報を更新したものが、第2の領域にコピーされる。

20

【0184】

すなわち、例えば、図7および図8の場合を例にすれば、第1の領域におけるポインタ領域の対象ブロックへのポインタの欄については、外部から供給されたデータを書き込む論理ブロックに対応付けられていたデータ領域の物理ブロック#00,#02へのポインタのポインタとなっている1バイト目に記憶されていた物理ブロック番号#18が、実際にデータを書き込んだデータ領域の物理ブロック#10,#11へのポインタのポインタである物理ブロック番号#1Bに更新される。

【0185】

また、実際にデータを書き込んだデータ領域の物理ブロック#10,#11へのポインタのポインタとなっていた物理ブロック番号#1Aが、データを書き込む論理ブロックに対応付けられていたデータ領域の物理ブロック#00,#02へのポインタのポインタとなった物理ブロック番号#1Cに更新される。

30

【0186】

さらに、第1の領域におけるポインタ領域の更新ブロックへのポインタの欄については、物理ブロック番号の更新されたポインタ領域の物理ブロック（更新ブロックであった物理ブロック）の物理ブロック番号#1B,#1Cが、ポインタ領域の更新ブロックとなつた物理ブロックの物理ブロック番号#18,#1Aにそれぞれ更新される。

【0187】

またさらに、第1の領域における最新性情報を更新され、以上の更新結果が、第2の領域に書き込まれる。

【0188】

40

この後、処理はステップS39に進み、第1の領域の全てのデータが消去されて、書き込み処理が終了される。

【0189】

一方、ステップS31において、第2の領域が有効と判断された場合は、ステップS40乃至S47,S52において、第1または第2の領域についての処理が、逆に第2または第1の領域についてそれを行わることを除けば、ステップS32乃至S39,S50における場合とそれ同様の処理が行われて、書き込み処理が終了される。

【0190】

なお、上述した処理のうち、ステップS31乃至S35,S50の処理、およびステップS31,S40乃至S43,S52の処理（以下、コマンド非依存の処理と称する）に

50

については、書き込みコマンドの有無に拘わらず、予め実行しておくことが可能である。

【0191】

そこで、例えば、R/W12からポーリングを受けた直後にコマンド非依存の処理を実行しておけば、実際に書き込みコマンドが指示された場合、書き込み処理に要する処理時間を短縮することができる。

【0192】

ところで、メモリコラプションが発生している場合、ステップS31では、第1の領域と第2の領域の両方が有効と判定され、この場合、処理はステップS48に進められる。

【0193】

ステップS48においては、第1の領域と第2の領域との最新性が判定される。ここで、第1の領域の方が第2の領域より新しいと判定された場合、処理はステップS49に進められ、第2の領域よりも新しい第1の領域としてのブロックに、第2の領域の中のポインタ領域へのポインタが上書きされる。さらに、第1の領域の最新性情報および有効性情報が更新される。この後、処理はステップS50に進められて、それ以降、上述した場合と同様の処理が行われる。10

【0194】

反対に、第2の領域の方が第1の領域より新しいと判定された場合、処理はステップS89に進められ、第1の領域よりも新しい第2の領域としてのブロックに、第1の領域中のポインタ領域へのポインタが上書きされる。さらに、第2の領域の最新性情報および有効性情報が更新される。この後、処理はステップS52に進められて、それ以降、上述した場合と同様の処理が行われる。20

【0195】

以上説明した書き込み処理によれば、今度においてデータを書き込む可能性があるデータ領域の全ての更新ブロックを、一括して消去するようにしたので、更新ブロックにデータを書き込む都度、消去する場合に比較して、消去に要する処理時間を短くすることができる。よって、書き込み処理に要する一連の処理時間を短縮することができる。

【0196】

また、書き込み処理のうち、コマンド非依存の処理を事前に実行するようすれば、実際に書き込みコマンドが指示されてから書き込み処理を終了するまでに要する処理時間を短縮することができる。30

【0197】

以上、本発明を非接触通信システムに適用した場合について説明したが、本発明は、EEPROMに代表されるような、データを書き込む場合、データ領域を、その書き込みに先立って消去する必要のある記録媒体にデータを書き込むあらゆる装置に適用することができる。

【0198】

また、本発明は、特に、例えば、上述の非接触通信システムや、また、接触型であっても、ユーザがI/Cカードの抜き差しを自由に行うことができるシステムなどの、いわば、データの送受信が不安定な状態で行われるシステムについて、有用である。

【0199】

なお、本実施の形態では、有効判定ブロックまたはポインタ領域に、ポインタ領域またはデータ領域の、いわば空き領域としての、ポインタ領域の更新ブロックの物理ブロック番号、またはデータ領域の更新ブロックの物理ブロック番号をそれぞれ記憶させるようにした。しかしながら、これらの更新ブロックの物理ブロック番号は、必ずしも記憶させる必要はない。ただし、更新ブロックの物理ブロック番号を記憶させておかなければ、データの書き込みの際に、更新ブロックとなる物理ブロックを、EEPROM66の記憶内容をサーチすることにより検出する必要があるため、処理の高速化などの観点からは、更新ブロックの物理ブロック番号を記憶させておくようになるのが望ましい。40

【0200】

また、本実施の形態では、EEPROM66の所定の位置に、有効判定ブロックを記憶させる領域、ポインタ領域、およびデータ領域を割り当てるようにしたが、これらの領域を割り50

当てる位置は、特に限定されるものではない。さらに、これらの領域は、EEPROM 6 6 に連続した範囲の領域として確保する必要もない。すなわち、有効判定ブロックを記憶させる領域、ポインタ領域、またはデータ領域それぞれは、EEPROM上の不連続な位置に確保することが可能である。

【0201】

さらに、本実施の形態では、EEPROM 6 6 に、前回の記憶内容のみを保持しておくようにしたが、その他、例えば、前回と前々回の記憶内容を保持しておくようにすることなども可能である。ただし、この場合、より多くの記憶容量が必要となる。

【0202】

ところで、上述した一連の処理は、ハードウェアにより実行することもできるし、ソフトウェアにより実行することもできる。

【0203】

なお、コンピュータが実行するプログラムは、本明細書で説明する順序に沿って時系列に処理が行われるプログラムであってもよいし、並列に、あるいは呼び出しが行われたとき等の必要なタイミングで処理が行われるプログラムであってもよい。

【0204】

また、プログラムは、1台のコンピュータにより処理されるものであってもよいし、複数のコンピュータによって分散処理されるものであってもよい。さらに、プログラムは、遠方のコンピュータに転送されて実行されるものであってもよい。

【0205】

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【0206】

なお、本発明の実施の形態は、上述した実施の形態に限定されるものではなく、本発明の要旨を逸脱しない範囲において種々の変更が可能である。

【符号の説明】

【0207】

10 非接触通信システム， 11 コントローラ， 12 リーダ／ライタ， 13
I C カード， 64 演算部， 91 シーケンサ， 92 暗号／復号部， 93
パリティ演算部， 66 EEPROM

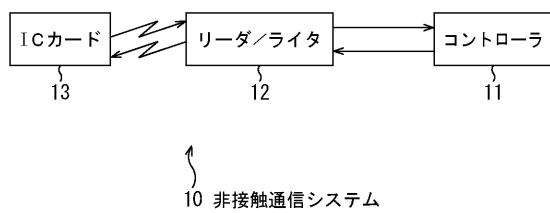
10

20

30

【図1】

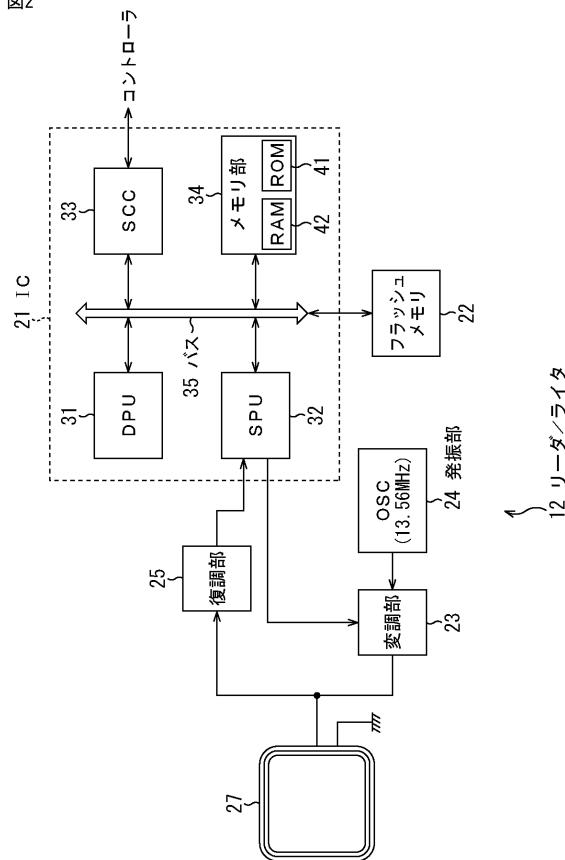
図1



10 非接触通信システム

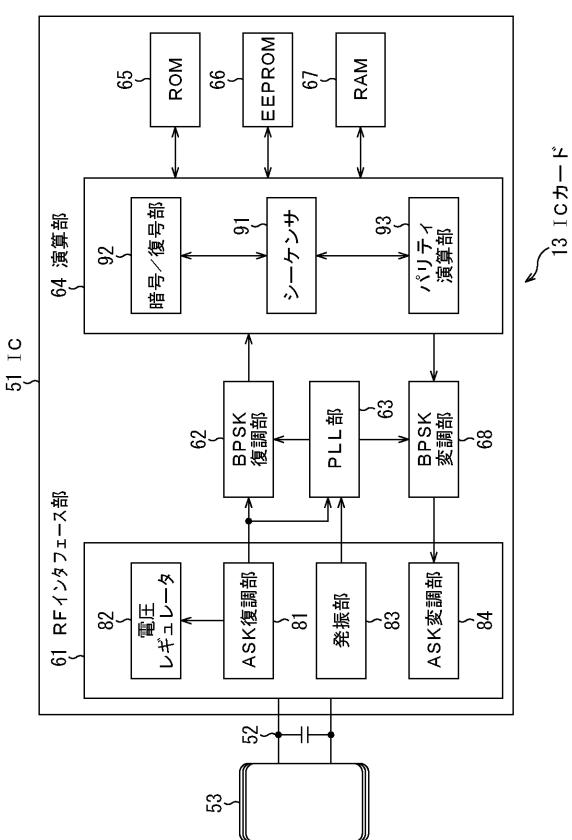
【図2】

図2



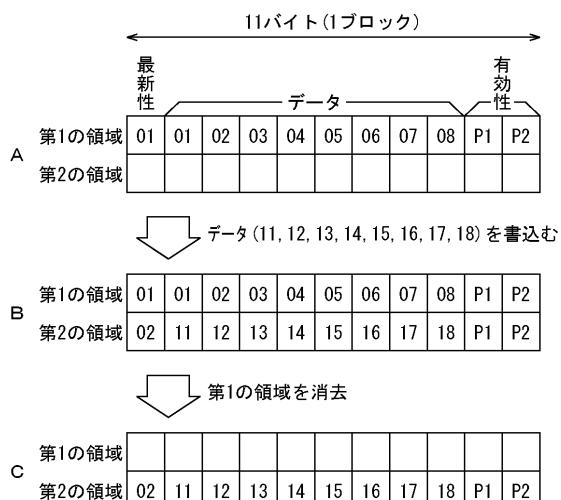
【図3】

図3



【図4】

図4



【図5】

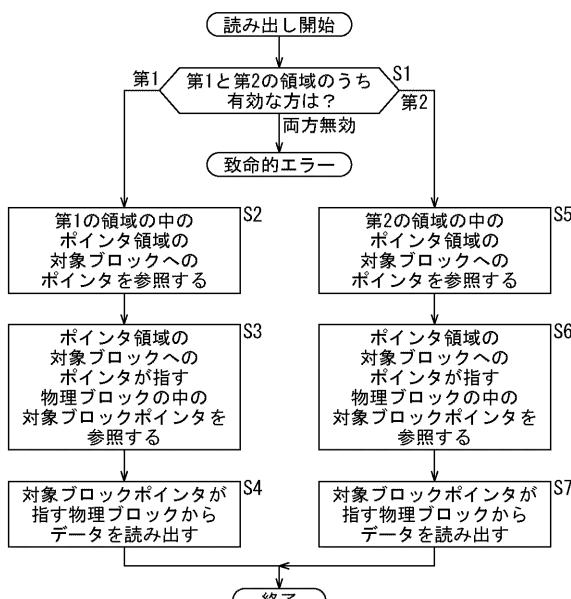
【図7】

【図6】

【図8】

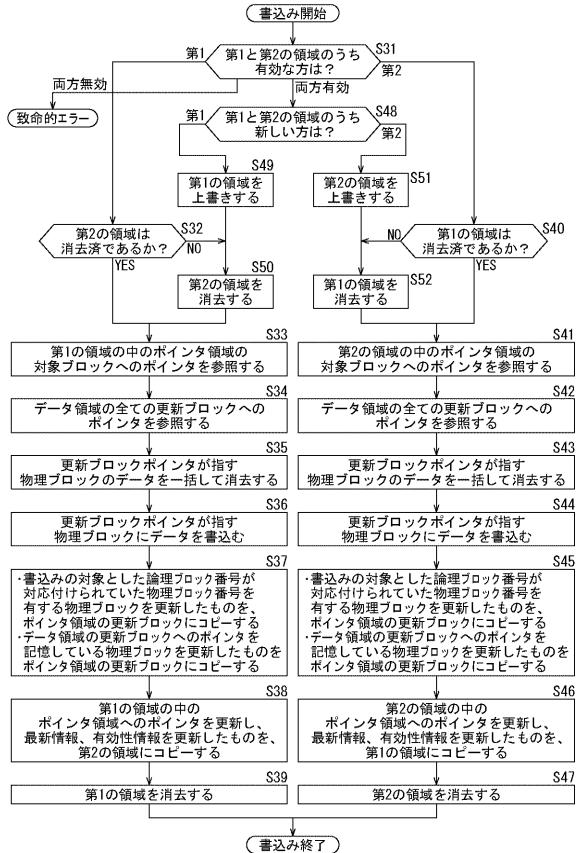
【図9】

図9



【図10】

図10



フロントページの続き

(56)参考文献 特開平11-025003(JP,A)
特開2002-343089(JP,A)
特開平09-073798(JP,A)
特開平10-240631(JP,A)
特開2005-235028(JP,A)
特開2003-150458(JP,A)

(58)調査した分野(Int.Cl., DB名)

G 06 F 12 / 16
G 06 K 17 / 00