

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号
特許第5832963号
(P5832963)

(45) 発行日 平成27年12月16日(2015.12.16)

(24) 登録日 平成27年11月6日(2015.11.6)

(51) Int.Cl.
G06K 19/07 (2006.01)

F I
G O 6 K 19/07

請求項の数 17 (全 56 頁)

(21) 出願番号	特願2012-147850 (P2012-147850)	(73) 特許権者	000003078
(22) 出願日	平成24年6月29日 (2012. 6. 29)		株式会社東芝
(65) 公開番号	特開2014-10708 (P2014-10708A)		東京都港区芝浦一丁目1番1号
(43) 公開日	平成26年1月20日 (2014. 1. 20)	(74) 代理人	100108855
審査請求日	平成26年9月10日 (2014. 9. 10)		弁理士 蔵田 昌俊
		(74) 代理人	100159651
			弁理士 高倉 成男
		(74) 代理人	100088683
			弁理士 中村 誠
		(74) 代理人	100109830
			弁理士 福原 淑弘
		(74) 代理人	100075672
			弁理士 峰 隆司
		(74) 代理人	100103034
			弁理士 野河 信久

最終頁に続く

(54) 【発明の名称】 メモリシステム

(57) 【特許請求の範囲】

【請求項 1】

不揮発性半導体記憶装置と、
前記不揮発性半導体記憶装置を制御する制御部と、
前記制御部に接続された作業エリアとしてのメモリと、
前記制御部により制御される拡張機能部と、
前記メモリ上に設けられ、前記拡張機能部の拡張機能を定義可能な一定のブロック長を有する拡張レジスタと、を具備し、
前記制御部は、前記拡張レジスタを介して前記拡張機能部を動作させるコマンドのヘッダデータを前記拡張機能部に書き込む第1のコマンドと、前記拡張レジスタを介して前記拡張機能部からレスポンスのヘッダデータを読み出す第2のコマンドとを処理することを特徴とするメモリシステム。

【請求項 2】

前記制御部は、前記拡張機能部に可変長のコマンドデータを書き込むための第3のコマンドと、前記拡張機能部から可変長のレスポンスデータを読み出す第4のコマンドとを処理することを特徴とする請求項1記載のメモリシステム。

【請求項 3】

前記第3のコマンドは、前記可変長のコマンドデータのブロックサイズとブロック数を指定するフィールドを有し、前記第4のコマンドは、前記可変長のレスポンスデータのブロックサイズとブロック数を指定するフィールドを有することを特徴とする請求項2記載

のメモリシステム。

【請求項 4】

前記コマンドのヘッダデータは、前記拡張機能部を動作させるためのオペレーションコードと、アークメントと、前記アークメントの長さ示すアークメント長と、前記可変長のコマンドデータの長さを示すコマンドデータ長とを含むことを特徴とする請求項 2 記載のメモリシステム。

【請求項 5】

前記レスポンスのヘッダデータは、レスポンスコードと、レスポンスデータと、前記レスポンスデータの長さ示すレスポンスデータ長とを含むことを特徴とする請求項 2 記載のメモリシステム。

10

【請求項 6】

前記拡張レジスタは、
前記コマンドのヘッダデータを転送するための第 1 のポートと、
前記レスポンスのヘッダを転送するための第 2 のポートと、
可変長のコマンドデータを転送するための第 3 のポートと、
可変長のレスポンスデータを転送するための第 4 のポートと
を有することを特徴とする請求項 1 記載のメモリシステム。

【請求項 7】

前記メモリシステムは、前記第 1 乃至第 4 のコマンドを受けるとレスポンスを返し、前記レスポンスは、前記拡張機能部にイベントが発生したことを示す第 1 のフラグを有することを特徴とする請求項 2 記載のメモリシステム。

20

【請求項 8】

前記拡張レジスタは、イベントが発生した前記拡張機能部を示す第 2 のフラグを有することを特徴とする請求項 7 記載のメモリシステム。

【請求項 9】

ホスト機器から転送される前記コマンドのヘッダデータと可変長のコマンドデータを分割する分割部と、

前記分割部により分割された前記コマンドのヘッダデータに対して前記第 1 のコマンドを発行し、前記可変長のコマンドデータに対して前記第 3 のコマンドと発行するコマンド発行部を有する変換器をさらに具備することを特徴とする請求項 2 記載のメモリシステム。

30

【請求項 10】

前記コマンド発行部は、前記第 2 のコマンドを発行して前記拡張機能部からレスポンスのヘッダデータを読み出し、前記第 4 のコマンドを発行して前記拡張機能部から前記可変長のレスポンスデータを読み出し、

前記変換器は、前記読み出されたレスポンスのヘッダデータと前記可変長のレスポンスデータを結合してホスト機器に転送する結合部をさらに具備することを特徴とする請求項 9 記載のメモリシステム。

【請求項 11】

ホスト機器とメモリデバイスとの間に設けられる電子装置であって、

40

前記メモリデバイスは、

不揮発性半導体記憶装置と、

前記不揮発性半導体記憶装置を制御する制御部と、

前記制御部に接続された作業エリアとしてのメモリと、

前記制御部により制御される拡張機能部と、

前記メモリ上に設けられ、前記拡張機能部の拡張機能を定義可能な一定のブロック長を有する拡張レジスタと、を具備し、

前記電子装置は、変換器を有し、

前記変換器は、

前記ホスト機器から転送されるコマンドのヘッダデータと可変長のコマンドデータを分

50

割する分割部と、

前記分割部により分割された前記コマンドのヘッダデータを前記拡張機能部書き込む第1のコマンドを発生し、前記拡張レジスタを介して前記拡張機能部からレスポンスのヘッダデータを読み出す第2のコマンドを発生するコマンド発行部と、

を具備することを特徴とする電子装置。

【請求項12】

前記コマンド発行部は、

前記拡張機能部に可変長のコマンドデータを書き込むための第3のコマンドと、前記拡張機能部から可変長のレスポンスデータを読み出す第4のコマンドとを処理することを特徴とする請求項11記載の電子装置。

【請求項13】

前記変換器は、前記拡張機能部から読み出されたレスポンスのヘッダデータと前記可変長のレスポンスデータを結合してホスト機器に転送する結合部をさらに具備することを特徴とする請求項12記載の電子装置。

【請求項14】

ホスト機器に接続される分配器と、

前記分配器に接続され、メモリデバイスに接続される変換器と、

前記分配器に接続され、ハードディスク装置に接続されるブリッジ回路と

を具備する電子装置であって、

前記メモリデバイスは、

不揮発性半導体記憶装置と、

前記不揮発性半導体記憶装置を制御する制御部と、

前記制御部に接続された作業エリアとしてのメモリと、

前記制御部により制御される拡張機能部と、

前記メモリ上に設けられ、前記拡張機能部の拡張機能を定義可能な一定のブロック長を有する拡張レジスタと、を具備し、

前記変換器は、

前記ホスト機器から転送されるコマンドのヘッダデータと可変長のコマンドデータを分割する分割部と、

前記分割部により分割された前記コマンドのヘッダデータを前記拡張機能部書き込む第1のコマンドを発生し、前記拡張レジスタを介して前記拡張機能部からレスポンスのヘッダデータを読み出す第2のコマンドを発生するコマンド発行部と

を具備することを特徴とする電子装置。

【請求項15】

前記コマンド発行部は、

前記拡張機能部に可変長のコマンドデータを書き込むための第3のコマンドと、前記拡張機能部から可変長のレスポンスデータを読み出す第4のコマンドとを処理することを特徴とする請求項14記載の電子装置。

【請求項16】

前記変換器は、前記拡張機能部から読み出されたレスポンスのヘッダデータと前記可変長のレスポンスデータを結合してホスト機器に転送する結合部をさらに具備することを特徴とする請求項15記載の電子装置。

【請求項17】

不揮発性半導体記憶装置と、

前記不揮発性半導体記憶装置を制御する制御部と、

前記制御部により制御される拡張機能部と、

前記拡張機能部の拡張機能を定義可能な一定のブロック長を有する拡張レジスタと、を具備し、

前記制御部は、前記拡張レジスタを介して前記拡張機能部を動作させるコマンドのヘッダデータを前記拡張機能部書き込む第1のコマンドと、前記拡張レジスタを介して前記

10

20

30

40

50

拡張機能部からレスポンスのヘッダデータを読み出す第2のコマンドとを処理することを特徴とするメモリシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明の実施形態は、例えば半導体不揮発性メモリを用いたメモリシステムに関する。

【背景技術】

【0002】

近時、メモリカードは、単なるメモリデバイスとしてだけでなく、付加価値を持たせるため様々な機能の追加が要望されている。また、追加機能をプラグ・アンド・プレイで使用可能にするために、汎用的な初期化手段が望まれている。

10

【先行技術文献】

【特許文献】

【0003】

【特許文献1】特開2004-46498号公報

【非特許文献】

【0004】

【非特許文献1】Part 1 Physical Layer Simplified Specification Ver3.01 May 18, 2010 (URL: <http://www.sdcard.org/home/>)

【非特許文献2】Part E1 SDIO Simplified Specification Ver2.00 Feb. 8, 2007 (URL: <http://www.sdcard.org/home/>)

20

【非特許文献3】Universal Serial Bus Mass Storage Class Bulk-Only Transport Revision 1.0 September 31, 1999 (URL: <http://www.usb.org/>)

【発明の概要】

【発明が解決しようとする課題】

【0005】

本実施形態は、拡張機能を容易に設定することが可能なメモリシステムを提供しようとするものである。

【課題を解決するための手段】

【0006】

30

本実施形態のメモリシステムによれば、不揮発性半導体記憶装置と、前記不揮発性半導体記憶装置を制御する制御部と、前記制御部に接続された作業エリアとしてのメモリと、前記制御部により制御される拡張機能部と、前記メモリ上に設けられ、前記拡張機能部の拡張機能を定義可能な一定のブロック長を有する拡張レジスタと、を具備し、前記制御部は、前記拡張レジスタを介して前記拡張機能部を動作させるコマンドのヘッダデータを前記拡張機能部に書き込む第1のコマンドと、前記拡張レジスタを介して前記拡張機能部からレスポンスのヘッダデータを読み出す第2のコマンドとを処理することを特徴とする。

【図面の簡単な説明】

【0007】

【図1】第1の実施形態に適用されるメモリシステムを概略的に示す構成図。

40

【図2】図1に示すメモリシステムのファームウェアの一例を示す構成図。

【図3】拡張レジスタのリードコマンドの一例を示す構成図。

【図4】リードコマンドによる拡張レジスタのリード動作を示すタイミング図。

【図5】リードコマンドによるデータポートのリード動作を示すタイミング図。

【図6】拡張レジスタのライトコマンドの一例を示す構成図。

【図7】マスクレジスタの動作を示す図。

【図8】ライトコマンドによる拡張レジスタのライト動作を示すタイミング図。

【図9】ライトコマンドによるデータポートのライト動作を示すタイミング図。

【図10】拡張レジスタの先頭ページに設定される汎用情報 (General Information) フィールドの一例を示す図。

50

- 【図 1 1】 リードコマンドに従ったメモリシステムの動作の一例を示すフローチャート。
- 【図 1 2】 ライトコマンドに従ったメモリシステムの動作の一例を示すフローチャート。
- 【図 1 3】 ホストドライバの動作の一例を示すフローチャート。
- 【図 1 4】 ホストドライバの動作の他の例を示すフローチャート。
- 【図 1 5】 S D I Oにおける拡張レジスタのアクセス動作を概略的に示す図。
- 【図 1 6】 レビジョン管理の一例を示す図。
- 【図 1 7】 第 2 の実施形態に係る拡張レジスタのリードコマンドの一例を示す図。
- 【図 1 8】 第 2 の実施形態に係る拡張レジスタのライトコマンドの一例を示す図。
- 【図 1 9】 リードコマンドによる拡張レジスタのリード動作を示すタイミング図。
- 【図 2 0】 リードコマンドによるデータポートのリード動作を示すタイミング図。 10
- 【図 2 1】 ライトコマンドによる拡張レジスタのライト動作を示すタイミング図。
- 【図 2 2】 ライトコマンドによるデータポートのライト動作を示すタイミング図。
- 【図 2 3】 拡張レジスタの先頭ページに設定される汎用情報フィールドの一例を示す図。
- 【図 2 4】 第 2 の実施形態に係るリードコマンドに従ったメモリシステムの動作の一例を示すフローチャート。
- 【図 2 5】 第 2 の実施形態に係るライトコマンドに従ったメモリシステムの動作の一例を示すフローチャート。
- 【図 2 6】 第 2 の実施形態に係る拡張レジスタのマルチブロックリードコマンドの一例を示す図。
- 【図 2 7】 第 2 の実施形態に係る拡張レジスタのマルチブロックライトコマンドの一例を示す図。 20
- 【図 2 8】 第 2 の実施形態に係る汎用情報の表示位置の一例を示す図。
- 【図 2 9】 第 2 の実施形態に係るメモリ空間と S D I O空間の関係の一例を示す図。
- 【図 3 0】 第 2 の実施形態に係る S D I Oの初期化の簡素化を説明するために示すフローチャート。
- 【図 3 1】 第 2 の実施形態に係るメモリデバイスとホストの機能インターフェースの関係を概略的に示す図。
- 【図 3 2】 第 2 の実施形態に係り、バッファの制御を説明するために示す概略構成図。
- 【図 3 3】 機能識別コードの一例を示す図。
- 【図 3 4】 第 3 の実施形態に係るホスト機器とメモリデバイスの関係を概略的に示す図。 30
- 【図 3 5】 第 3 の実施形態に係る拡張レジスタのリードコマンドの一例を示す構成図。
- 【図 3 6】 第 3 の実施形態に係る拡張レジスタのライトコマンドの一例を示す構成図。
- 【図 3 7】 図 3 7 (a) (b) (c) は、それぞれコマンドのパターンを示す図。
- 【図 3 8】 コマンドヘッダ、コマンドペイロード、レスポンスヘッダ、レスポンスペイロードのデータ構造の一例を示す図。
- 【図 3 9】 図 3 9 (a) (b) (c) は、それぞれコマンドヘッダ、ペイロード、及びレスポンスヘッダの一例を示す図。
- 【図 4 0】 図 4 0 (a) (b) (c) は、それぞれコマンドヘッダ、ペイロード、及びレスポンスヘッダの一例を示す図。
- 【図 4 1】 コマンドヘッダの“ OP Code ”の一例を示す図。 40
- 【図 4 2】 第 3 の実施形態に適用される拡張レジスタマップの一例を示す図。
- 【図 4 3】 図 3 7 (a) に対応するデータ転送の一例を示す図。
- 【図 4 4】 図 4 3 に示すデータ転送を説明するために示すタイミングチャート。
- 【図 4 5】 図 3 7 (b) に対応するデータ転送の一例を示す図。
- 【図 4 6】 図 4 5 に示すデータ転送を説明するために示すタイミングチャート。
- 【図 4 7】 図 3 7 (c) に対応するデータ転送の一例を示す図。
- 【図 4 8】 図 4 7 に示すデータ転送を説明するために示すタイミングチャート。
- 【図 4 9】 第 4 の実施形態に係わり、レスポンスのデータ構造の一例を示す図。
- 【図 5 0】 拡張機能のイベントを表示するフラグの一例を示す図。
- 【図 5 1】 ホスト機器に設けられたアプリケーション及び拡張機能の一例を示す構成図。 50

【図 5 2】図 5 1 に示すホスト機器とメモリデバイスの動作の一例を示すシーケンスチャート。

【図 5 3】第 5 の実施形態に係わり、変換器を用いた構成を概略的に示す図。

【図 5 4】第 5 の実施形態の動作を説明するために示すシーケンスチャート。

【図 5 5】第 5 の実施形態の別の動作を説明するために示すシーケンスチャート。

【図 5 6】図 5 3 の変形例を示す構成図。

【発明を実施するための形態】

【0008】

本実施形態を概略的に説明すると、次のようである。

【0009】

(機能拡張方法)

ホストドライバが、追加機能を制御する機能ドライバを探して、対応する機能ドライバがホストにインストールされていた場合、その機能ドライバに制御を渡す仕組みを取り入れることにより、機能拡張が容易に行えるようになる。機能固有の制御は、機能ドライバの中に隠蔽されるため、ホストドライバは最小限の情報のみで追加機能を実装可能となる。例えばファームウェアが管理する複数ページの拡張レジスタを有し、これら拡張レジスタのページ 0 に、特定のドライバを認識するための標準的な汎用情報(general information)フィールドを提供する。これによりホストシステムは、プラグ・アンド・プレイの実装が可能となる。また、マルチ機能カード/デバイスをサポートするために、個々の機能を指し示すことができるようにホストシステムが管理しておくことで、ホストソフトウェアの改変なしにマルチ機能カード/デバイスを使用可能にする。

【0010】

(SDメモリ又はSDIOホストコントローラの対応)

SDメモリ用のホストコントローラにおいても、追加機能の制御を効率良く行える拡張レジスタをアクセスするための専用コマンドを定義する。512 バイトの固定ブロック長の転送とすることにより、従来のSDメモリ用ホストコントローラからこの専用コマンドを発行できる。さらに、コマンドの引数として有効データ長の情報や、ライト時のマスク機能を持つことにより、リード・モディファイ・ライトを不要にすることが可能となる。

【0011】

SDIOカード対応のホストコントローラにおいては、SDIOのアクセスコマンドからも、拡張レジスタをアクセスできるようにすることにより、小さいブロック長転送とマルチブロック転送に対応可能となるため、さらに最適化したドライバを作ることが可能になる。

【0012】

データの転送用ポートとしてのデータポートをサポートすることで、拡張レジスタ空間の消費量が少ない実装が可能となる。また、データポートを用いることにより、拡張レジスタ以外のデバイスへのデータ転送も効率良く行うことが可能となる。複数ブロックによるバースト転送コマンドをサポートすることができる。データポートは、機能の実装時に拡張レジスタの任意のアドレスをデータポートとして定義できる。カードはアドレスを解読してデータポートか拡張レジスタかを判断する。

【0013】

(リロケータブルアドレスによる拡張レジスタ定義)

カードベンダーにより、拡張レジスタ上の任意の位置に追加機能を制御するレジスタを割り当てることを可能とし、実装したレジスタのアドレス情報を汎用情報フィールドから提供することにより、レジスタ配置をリロケータブルとすることを可能としている。このため、従来標準化が必要であったアドレス配置は不要となり、メモリデバイスを製造し易くなる。再配置が可能であるため、レジスタを拡張しても容易に対応ができる。

【0014】

以下、実施の形態について、図面を参照して説明する。

【0015】

10

20

30

40

50

(第1の実施形態)

図1は、本実施形態に係るメモリシステムを概略的に示している。

【0016】

メモリシステムは、例えばSDカードのようなメモリデバイス11と、ホスト機器20により構成される。

【0017】

メモリデバイス11は、ホスト機器20に接続されたときに電源供給を受けて動作し、ホスト機器20からのアクセスに応じた処理を行う。このメモリデバイス11は、コントローラ11aを有している。

【0018】

コントローラ11aは、例えばホストインタフェース(I/F)12、CPU13、ROM(Read only Memory)14、RAM(Random Access Memory)15、バッファ16、メモリインタフェース(I/F)17により構成されている。これらは、バスにより接続されている。メモリインタフェース17には、例えばNAND型フラッシュメモリ18と、拡張機能部としてのSDIO19が接続されている。拡張機能部は、例えば無線LAN装置などを適用することが可能である。

【0019】

ホストインタフェース12は、コントローラ11aとホスト機器20との間のインターフェース処理を行う。

【0020】

メモリインタフェース17は、コントローラ11aとNAND型フラッシュメモリ18、又はSDIO19との間のインターフェース処理を行う。

【0021】

CPU13は、メモリデバイス11全体の動作を司るものである。このCPU13を正制御するプログラムは、ROM14の中に格納されているファームウェア(制御プログラム等)を用いるかあるいは、RAM115上にロードして所定の処理を実行する。すなわち、CPU13は、各種のテーブルや後述する拡張レジスタをRAM18上に作成したり、ホスト機器20からライト(書き込み)コマンド、リード(読み出し)コマンド、イレース(消去)コマンドを受けてNAND型フラッシュメモリ18上の領域をアクセスしたり、バッファ16を介してデータ転送処理を制御したりする。

【0022】

ROM14は、CPU13により使用される制御プログラムなどのファームウェアを格納する。RAM15は、CPU13の作業エリアとして使用され、制御プログラムや各種のテーブルや後述する拡張レジスタを記憶する。

【0023】

バッファ16は、ホスト機器20から送られてくるデータを、例えばNAND型フラッシュメモリ18へ書き込む際、一定量のデータ(例えば1ページ分)を一時的に記憶したり、NAND型フラッシュメモリ18から読み出されたデータをホスト機器20へ送り出す際、一定量のデータを一時的に記憶したりする。またバッファを介することにより、SDバスインタフェースとバックエンドを非同期に制御することができる。

【0024】

NAND型フラッシュメモリ18は、例えば積層ゲート構造のメモリセル、又はMONOS構造のメモリセルにより構成されている。

【0025】

SDIO19は、例えばデジタルカメラやPHSなどの周辺機器やインターフェースとしての機能を有している。例えば、SDIO19として無線LAN装置を適用することで、無線通信機能を有さないデジタルカメラでも外部サーバ、外部PC等との間で無線によるデータ通信を行うことが可能となる。

【0026】

ホスト機器20は、例えばデジタルカメラやPHSなどが適用可能である。ホスト機器

10

20

30

40

50

20は、ホストコントローラ21、CPU22、ROM23、RAM24、例えばハードディスク25(SSDを含む)により構成されている。これらはバスにより接続されている。

【0027】

CPU22は、ホスト全体を制御する。ROM23は、CPU22の動作に必要なファームウェアを記憶している。RAM24は、例えばCPU22の作業領域として使用されるが、CPU22が実行可能なプログラムもここにロードされ実行される。ハードディスク25は、各種データを保持する。ホストコントローラ21は、メモリデバイス11が接続された状態において、メモリデバイス11とのインターフェース処理を行う。さらに、CPU22の指示に従って、後述する各種コマンドを発行する。

10

【0028】

(ファームウェアの構成)

図2は、メモリデバイス11のROM14に記憶されたファームウェアの機能構成の一例を示している。これらの機能はコントローラ11aを構成するCPU13等の各ハードウェアとの組み合わせにより実現されるものである。ファームウェアは、例えばコマンド処理部14a、フラッシュメモリ制御部14b、拡張レジスタ処理部14c、機能処理プログラム14dにより構成されている。拡張レジスタ処理部14cは、メモリデバイス11が起動された際、RAM15内に拡張レジスタ31を生成する。この拡張レジスタ31は、仮想レジスタであり、拡張機能を定義可能とされている。

20

【0029】

(拡張レジスタの構成)

図2に示すように、拡張レジスタ31は、例えば8ページにより構成されている、1ページは、512バイトにより構成されている。512バイトの拡張レジスタをバイト単位にアクセスするため、最低9ビットのアドレスが必要となり、8ページアクセスするために、最低3ビットのアドレスが必要となる。合計12ビットのアドレスにより、拡張レジスタの全空間がアクセス可能となる。512バイトは殆どのホストがサポート可能なアクセス単位であるが、512バイトに限定されず大きくしても良い。長いビット長のアドレスフィールドで構成される場合は、下位何ビットかがアクセス単位として使用され、残りの上位のビットは、複数ページのひとつを選択するために使用される。

30

【0030】

512バイト単位とする理由は、多数のメモリカードホストコントローラが、1ブロック=512バイトを単位としてリード/ライト転送を行う構成になっているためである。SDIO対応のホストコントローラであれば、1バイト単位のリード/ライトが可能であるが、全てのホストコントローラがこれをサポートしているわけではない。大多数のホストコントローラで拡張機能を制御できるようにするためには、512バイト単位のアクセスが行えると都合が良い。

【0031】

8ページ(ページ0~ページ7)の内、ページ0は、拡張機能のプラグ・アンド・プレイを行うために汎用情報フィールドを記録しておくための領域である。汎用情報フィールドの詳細については後述する。ページ1~ページ7は、拡張機能を制御するためのレジスタが定義される。ページ0は位置が特定し易いために、汎用情報フィールドを記録しておく場所としては適切であるが、必ずしもページ0である必要はなく、特定のページ位置を汎用情報フィールドの記載する場所として定義することもできる。

40

【0032】

拡張レジスタのリード/ライトは、以下に定義される専用のリード/ライトコマンドが用いられる。これらのコマンドは、拡張レジスタをリード/ライトする第1の動作モードと、データポートを構成する第2の動作モードを有している。

【0033】

(拡張レジスタのリードコマンド(CMD48))

図3は、拡張レジスタのリードコマンド(CMD48)のフィールド構成の一例を示し

50

ている。“S”は、コマンドのスタートビットを示し、“T”は転送方向を示すビットであり、“index”は、コマンド番号を示している。“RS”(レジスタセレクト)は拡張レジスタ31内のページを示し、“OFS”は選択されたページ内におけるデータの位置(ページの先頭からのオフセット)を示している。3ビットの“RS”と、9ビットの“OFS”で、512バイトの拡張レジスタ8ページ分の空間をバイト単位に指定することができる。具体的には、選択された拡張レジスタ内のリード開始位置が“RS”と“OFS”により指定される。

【0034】

“LEN”はデータ長を示している。9ビットのLENフィールドにより、512バイトの拡張レジスタ内の読み出しに必要な有効なデータ長が指定される。

10

【0035】

“CRC7”は、巡回冗長検査(cyclic redundancy check)コードを示し、“E”は、コマンドのエンドビットを示している。“rsv”は、予備のビットを示している。

【0036】

(拡張レジスタのリードコマンド、第1の動作モード)

図4は、第1の動作モードによる拡張レジスタのリード動作の例を示している。

【0037】

図4に示すように、メモリデバイス11は、ホスト機器20からコマンド(CMD48)を受け取ると、レスポンス(R1)をホスト機器20に返し、その後、拡張レジスタ31から512バイトのデータブロックを読み出す。

20

【0038】

具其他的には、コマンド(CMD48)の引数で、拡張レジスタのページと、ページ内の読み出すべきデータの位置が、“RS”と“OFS”で指定され、データ長が“LEN”で指定される。このようにして指定された拡張レジスタ内のデータが、512バイトのデータブロックの先頭にセットされ、読み出される。512バイトのデータブロックのうち、“LEN”で指定されたデータ長を超えるデータは、無効データとなる。データブロックの最後にはCRCコードが付加され、正しくデータが受け取れたかをチェックすることが可能とされている(無効データを含めてチェックを行う)。有効データが先頭から配置されているため、ホスト機器20は、有効データを探すために、データシフトなどの操作を行う必要がない。

30

【0039】

(拡張レジスタのリードコマンド、第2の動作モード)

図5は、第2の動作モードによるデータポートリードの動作の例を示している。

【0040】

メモリデバイス11は、このコマンド(CMD48)を受け取ると、レスポンス(R1)を返し、その後に512バイトのデータブロックを返す。

【0041】

コマンドの引数“RS”、“OFS”により、拡張レジスタの選択されたページ内の位置が指定される。図5ではレングスが1の場合のデータポート例が示されている。すなわち、データポートは、拡張レジスタマップ上において、1バイトのアドレスを占有するだけで良い。データポートであるかどうかをアドレスのデコードによって識別できれば良く、実際に1バイト幅のポートを通してデータが伝送される必要はないので、データ伝送性能には影響しない。このデータポートに割り当てられたデバイスから1ブロック(512バイト単位)のデータをリードすることができる。すなわち、1回当たり、1ブロック(512バイト単位)のデータをリードすることができる。この読み出されたデータは、例えばバッファ16に保持され、ホスト機器20によって読み出される。

40

【0042】

続いて同じデータポートをリードすると、続きの512バイトのデータを読み出すことができる。データポートから読み出すデータを何処から取ってくるかは、拡張機能の仕様によって自由に定義ができる。データポート制御は、例えば、拡張レジスタ上に制御レジ

50

スタを定義して制御することができる。512バイトのデータブロックの最後にCRCコードが付加され、正しくデータが受け取れたか否かがチェック可能とされている。

【0043】

(拡張レジスタのライトコマンド (CMD49))

図6は、拡張レジスタのライトコマンドの一例を示している。ライトコマンド (CMD49) において、リードコマンド (CMD48) と同一部分には同一符号を付している。ライトコマンドとリードコマンドは、“index”により区別される。3ビットの“RS”と、9ビットの“OFS”により、拡張レジスタのページと選択されたページ内のデータの位置が指定される。9ビットの“LEN”フィールドにより、512バイトの拡張レジスタに書き込むデータ長が指定される。したがって、512バイト内の任意のデータ長(バイト単位)のデータを拡張レジスタの任意のページと場所に書き込むことが可能である。

10

【0044】

ライトコマンド (CMD49) は、コマンドの引数の中にマスクレジスタが設けられている。すなわち、“Mask”は、8ビット長のマスクレジスタを示している。このマスクレジスタにより、1バイトのデータのライトにおいて、ビット単位のオペレーションが可能となり、特定のビットにのみデータを書き込むことが可能となる。このため、1バイト内のビットオペレーションであれば、リード・モディファイ・ライトを行う必要がない。マスクレジスタは、データ長が1バイトのとき、すなわち、“LEN=0”(レングス1)のとき有効となる。マスクレジスタ“Mask”のデータが“1”のビットは、データが書き込まれ、マスクレジスタ“Mask”のデータが“0”のビットは、既にセットされた値が保存される。

20

【0045】

すなわち、図7(a)に示すようなデータを保持している拡張レジスタを仮定した場合において、マスクレジスタのデータが、図7(b)に示すような場合、ライトコマンドが実行されることにより、図7(c)に示すように、マスクレジスタのデータが“1”のビットはデータが書き込まれ、データが“0”のビットは、元のデータが保持される。このため、リード・モディファイ・ライトを行うことなく、所要のビットのみデータを書き換えることが可能となる。“x”で示す部分が、新しいデータが書き込まれたビットを示す。

30

【0046】

また、より長いマスクデータを別の手段によって供給できる場合、LEN>1でもマスクライトが可能であるが、図7に示す例では、コマンド引数にマスクデータを割り当てているため、8ビットマスクとしている。

【0047】

(拡張レジスタのライトコマンド、第1の動作モード)

図8は、第1の動作モードによる拡張レジスタのライト動作の例を示している。

【0048】

メモリデバイス11は、このコマンド (CMD49) を受け取ると、レスポンス (R1) を返し、その後、512バイトのデータブロックを受け取る。

40

【0049】

メモリデバイス11は、データブロックが正しく受け取れたかどうかを示すCRCコードをホスト機器20に返す。その後、このコマンドの処理が終了するまでビジーを返し、ホスト機器20が次のコマンドを発行できるタイミングを知らせる。データブロックは、バッファ16に保持されている。

【0050】

コマンド処理において、コマンドの引数“RS”、“OFS”により、拡張レジスタ内のページと位置が指定され、“LEN”によりデータ長が指定される。バッファ16に保持されたデータブロックのうち、先頭から“LEN”で指定した長さのデータが拡張レジスタに書き込まれる。“LEN”で指定されたデータ長を超えるデータブロック中のデー

50

タは無効データとして破棄される。

【0051】

有効データをデータブロックの先頭から配置することにより、ホストシステムは有効データをデータブロックの途中に配置する操作が不要となる。

【0052】

(拡張レジスタのライトコマンド、第2の動作モード)

図9は、第2の動作モードによるライトデータポートの動作の例を示している。

【0053】

メモリデバイス11は、このコマンド(CMD49)を受け取ると、レスポンス(R1)を返し、その後、512バイトのデータブロックを受け取る。

10

【0054】

メモリデバイス11は、データブロックが正しく受け取れたかどうかを示すCRCコードをホストに返す。その後、このコマンド処理が終わるまでビジーを返し、ホスト機器20が次のコマンドを発行できるタイミングを知らせる。データブロックは、バッファ16に保持されている。

【0055】

コマンド処理において、コマンドの引数“RS”、“OFS”により、拡張レジスタ内のページと位置が指定され、データポートが指定される。データポートは、拡張レジスタマップ上において、1バイトのアドレスを占有するだけでよい。このデータポートに、バッファ16に保持された1ブロック(512バイト単位)のデータがある割り当てたデバイスにライトすることができる。すなわち、1回当たり、1ブロックのデータをライトすることができる。

20

【0056】

続いて同じデータポートをライトすると、続く512バイトのデータを割り当てたデバイスに書き込むことができる。データポートのデータを何処に渡すかは、拡張機能の仕様によって自由に定義ができる。データポート制御は、例えば、拡張レジスタ上に制御レジスタを定義して制御することができる。

【0057】

(汎用情報フィールドの使用例)

図10は、拡張レジスタ31のページ0に示された汎用情報フィールドの例を示している。この汎用情報フィールドにより、ホスト機器20が拡張機能を制御するドライバを特定できるようにすることにより、拡張機能を追加した場合において、ホストシステムが容易に拡張機能を使えることができ、プラグ・アンド・プレイを実現することができる。

30

【0058】

図10を参照して、標準のホストドライバが処理すべきシーケンス例を説明する。

【0059】

(ストラクチャレビジョン)

ストラクチャレビジョンは、拡張レジスタ31のページ0のフォーマットを定義するレビジョンである。汎用情報フィールドに新しい情報を追加した場合、ストラクチャレビジョンを更新することにより、どのバージョンの汎用情報フィールドを保持しているかを示す。以前のバージョンの機能ホストドライバは、新しいフィールドを無視する。

40

【0060】

(データ長)

データ長は、ページ0に記録されている有効データ長を示している。

【0061】

(拡張機能数(=N))

拡張機能数は、デバイスが何個の拡張機能をサポートしているかを示している。ホストドライバは、起動時に、サポートしている機能数だけ繰り返し、各拡張機能用のドライバがインストールされているかどうかを調べる。

【0062】

50

(デバイス 1 機能識別コード)

デバイス 1 機能識別コードに、コードが設定してある場合、標準ドライバを用いることができることを示す。OS が標準ドライバをサポートしている場合、専用ドライバをインストールすることなく、このデバイスが使用できる。専用ドライバがインストールされている場合は、そちらの使用を優先する。非標準の機能の場合、このフィールドに“0”が設定される。この場合は、専用ドライバによってのみこの機能は制御される。

【0063】

(デバイス 1 製造者識別情報、デバイス 1 機能識別情報)

デバイス 1 製造者識別情報、デバイス 1 機能識別情報は、それぞれ専用ドライバを特定するための情報であり、これらのフィールドには、例えば ASCII 文字列により製造者名や販売者名、又は拡張機能の識別情報が記載される。ホストドライバは、これらの情報をもとにデバイス 1 の専用ドライバがインストールされているかどうかを探す。

10

【0064】

機能識別情報には、例えば ASCII 文字列によりデバイスの型番、レビジョンなどが記載される。

【0065】

(次デバイスの先頭アドレス)

次デバイスの先頭アドレスは、次のデバイス情報が記載されているページ 0 内のアドレスを示している。ホストシステムがこのデバイスをサポートしていない場合、このデバイスは使用できないため、次のデバイスがチェックされる。これ以降のフィールドは可変長のため、この位置に定義している。

20

【0066】

(デバイス 1 アドレスポインタ 1 ~ X、レングスフィールド 1 ~ X)

デバイス 1 アドレスポインタ 1 ~ X、レングスフィールド 1 ~ X は、ひとつの機能に複数の拡張レジスタ領域を定義できることを示している。それぞれのアドレスとレングスを下記に列挙する。

【0067】

(デバイス 1 アドレスポインタ 1 (開始アドレス)、レングス 1)

デバイス 1 が使用する拡張レジスタの第 1 領域。拡張レジスタのページ 1 ~ 7 の空間内の先頭アドレスと、使用する拡張レジスタ領域の大きさを示している。

30

【0068】

すなわち、1 つのデバイスに、1 つ又は複数の拡張レジスタ領域を割り付けることができ、アドレスポインタは、ページ 0 以外の任意の拡張領域の場所 (開始アドレス) を示している。レングスは、ポインタを先頭アドレスとした拡張レジスタを占有する大きさを示している。

【0069】

(デバイス 1 アドレスポインタ 2 (開始アドレス)、レングス 2)

デバイス 1 に割り当てられた拡張レジスタ内の第 2 領域の位置と領域の大きさを示している。これにより、例えば標準ドライバは、第 1 領域のみで制御するが、専用ドライバは、第 1 領域と第 2 領域を用いて効率良く制御することを可能にするなどの応用が可能となる。

40

【0070】

(デバイス 1 アドレスポインタ X (開始アドレス)、レングス X)

デバイス 1 に割り当てられた第 X 領域の位置と領域の大きさを示している。

【0071】

このように、拡張レジスタ内に複数の領域を定義できる。各領域はオーバーラップしないように配置される。レングス情報によりオーバーラップがあるかどうかをチェックすることができる。

【0072】

追加フィールドが必要になった場合、これ以降に追加分定義していく。新しいフィールド

50

が認識できないホストは、認識可能なフィールドまで読み出し、追加フィールドは無視する。上記の（次デバイスの先頭アドレス）フィールドによりスキップすることができる。

【0073】

（リードコマンド（CMD48）の動作）

図11は、上記リードコマンド（CMD48）に対応するメモリデバイス11内のコントローラ11aの動作を示している。

【0074】

リードコマンドを受けると、CPU13により、コマンドの引数“RS”、“OFS”が解析され、リードコマンドがデータポートかどうか判別される（ST11）。すなわち、拡張レジスタ内のページ“RS”と、ページ内におけるデータの位置が判別される。この結果、コマンドが拡張レジスタのリードであると判別された場合、第1の動作モードにより、拡張レジスタ31の選択されたページの“OFS”で示された位置からデータ長“LEN”のデータが取得される（ST12）。この取得されたデータは、バッファ16にセットされる（ST13）。

【0075】

一方、ステップST11において、リードコマンドがデータポートであると判別された場合、第2の動作モードにより、拡張レジスタ31の選択されたページの“OFS”で示された位置のデータポートを介して、例えばSDIO19の特定のファンクションから512バイトのデータが取得される（ST14）。この取得されたデータは、バッファ16にセットされる（ST15）。

【0076】

（ライトコマンド（CMD49）の動作）

図12は、上記ライトコマンド（CMD49）に対応するメモリデバイス11内のコントローラ11aの動作を示している。

【0077】

ライトコマンドを受けると、CPU13（コマンド処理部14a）により、コマンドの引数“RS”、“OFS”が解析され、ライトコマンドがデータポートかどうか判別される（ST21）。すなわち、拡張レジスタ内のページ“RS”と、ページ内におけるデータの位置が判別される。この結果、ライトコマンドがデータポート以外と判別された場合、コマンドの引数“LEN=0”（レングス1）であるかどうか、すなわち、マスクが有効かどうか判別される（ST22）。この判別の結果、“LEN=0”ではないと判別された場合（レングスが1より大きい）、拡張レジスタ処理部14cにより拡張レジスタのライト処理が行われる。すなわち、バッファ16から“LEN”により指定された長さのデータが取得される（ST23）。この取得されたデータは、拡張レジスタの“RS”で選択されたページの“OFS”で指定された位置にセットされる。

【0078】

一方、上記ステップST22において、“LEN=0”であり（レングスが1）、マスクが有効であると判別された場合、拡張レジスタ処理部14cによりバッファ16から1バイトのデータと、1バイトのマスクが取得される（ST25）。この1バイトのデータと、1バイトのマスクにより、図7A、7B、7Cに示すマスク動作が実行され、拡張レジスタの“RS”で選択されたページの“OFS”で指定された位置のデータの一部が書き換えられる（ST26）。

【0079】

また、上記ステップ21において、データポートであると判別された場合、バッファ16から512バイトのデータが取得される（ST27）。この取得されたデータは、拡張レジスタ31の選択されたページの“OFS”で示された位置のデータポートを介して、例えばSDIO19の特定のファンクションに転送される（ST28）。

【0080】

（ホストドライバ処理）

図13は、ホスト機器20の処理を示している。ホスト機器20に、メモリデバイス1

10

20

30

40

50

1 が接続されると、メモリデバイス 11 が起動され、メモリデバイス 11 の RAM 15 に拡張レジスタ 31 が展開される。メモリデバイス 11 は、ホストドライバによって先ず、リードコマンド (CM 48) を発行して、拡張レジスタ 31 のページ 0 のデータを取得する (ST 31)。次に、取得したページ 0 のストラクチャレビジョンが確認され、どのバージョンの汎用情報フィールドを保持しているかが確認される (ST 32)。この後、サポート機能数 N と、デバイス情報の先頭アドレスが取得される (ST 33、ST 34)。

【0081】

次いで、ホスト機器 20 内に、取得した拡張機能に対応する専用機能ドライバがインストールされているかどうか検索される (ST 35、ST 36)。この結果、専用機能ドライバが無い場合、拡張レジスタのページ 0 に記載された機能識別コードが “0” であるかどうか判別される (ST 37)。この結果、機能識別コードが “0” である場合、この拡張機能はサポートされていないため、このデバイスは使用できないと認識され、次のデバイスに対するドライバの検索に移る (ST 34)。

10

【0082】

また、ステップ ST 37 の判別の結果、機能識別コードが “0” でない場合、ホスト機器 20 にインストールされている標準機能ドライバが検索される (ST 38、ST 39)。この結果、標準機能ドライバが無い場合、この拡張機能はサポートされていないため、デバイスは使用できないと認識され、次のデバイスに対するドライバの検索に移る (ST 34)。

【0083】

20

また、ステップ ST 35、ST 36 の検索の結果、標準機能ドライバが有った場合、及びステップ ST 35、ST 36 の検索の結果、専用機能ドライバが有った場合、ページ 0 に記載されたデバイスのアドレス及びレンジ数が取得される (ST 40)。この動作がアドレス及びレンジ数だけ実行される (ST 41)。

【0084】

この後、検索された専用機能ドライバ、又は標準機能ドライバがホスト機器 20 の例えばハードディスク 25 から RAM 24 にロードされ、ページ 0 に記載された 1 つ又は複数の拡張領域のアドレスポインタ (開始アドレス) が機能ドライバに渡され、その拡張機能が初期化される (ST 42)。上記アドレス、レンジ情報は、前記 RAM 24 にロードした機能ドライバを実行するときに渡される。標準機能ドライバと専用機能ドライバでは、受け渡しできるアドレス、レンジ情報の数が異なる可能性があるが、ページ 0 に登録された順番に受け渡しできる数だけ受け渡される。したがって、最初に登録されたアドレス、レンジ領域は、共通的な機能レジスタとして働き、後ろに登録されているアドレス、レンジ領域は、オプション的な役割を果たすことができる。

30

【0085】

初期化は機能ドライバが行う。すなわち、機能ドライバは、ホストドライバから渡された開始アドレスを基に、その機能に割り当てられた拡張レジスタをアクセスしてデバイスを初期化する。初期化においては、デバイスの消費電力を考慮する必要がある。ホストが供給可能な電力の範囲でデバイスを使う必要があるためである。デバイスが複数のパワーモードを持っている場合、ホストが供給可能なデバイスパワー以下のパワーモードを選択する必要がある。ホストシステムは、別な手段によってホストシステムが供給可能な電力を機能ドライバに伝えることで、パワーモードの選択が可能となる。

40

【0086】

上記ステップ ST 34 ~ ST 43 の動作が、サポート機能数 N に達するまで繰り返される (ST 43)。

【0087】

尚、ページ 0 に例えば新たなフィールドが追加された場合、新しいフィールドの処理がステップ ST 40、ST 41 の間に追加される。新しいフィールドを識別できないホストドライバは、そのフィールドをスキップするように構成される。

【0088】

50

上記のように、ホスト機器 20 は、拡張レジスタ 31 のページ 0 の情報を取得し、この情報に基づき、ドライバを検索することにより、プラグ・アンド・プレイを実現することができる。また、従来のように拡張レジスタの固定位置を決める必要がなく、デバイスベンダーは任意の拡張レジスタ位置に機能を定義することができるため、機能拡張を容易に実装することができる。

【0089】

図 14 は、図 13 の変形例を示すものであり、図 13 と同一部分には同一符号を付し、異なる部分についてのみ説明する。

【0090】

図 14 は、専用機能ドライバと、標準機能ドライバの検索処理が異なっている。すなわち、ステップ S T 34 において、デバイス情報の先頭アドレスが取得された後、先ず、機能識別コードが“0”であるか否かが判別される（S T 51）。この結果、“0”でない場合、すなわち、標準機能である場合、専用ドライバを使用するかどうかを判別される（S T 53）。この結果、専用ドライバを使用しない場合、標準機能ドライバが検索される（S T 54、S T 55）。この検索の結果、標準機能ドライバが無い場合、及びステップ S T 53 において、専用機能ドライバを使用すると判別された場合、専用機能ドライバが検索される（S T 52、S T 56）。この検索の結果、専用機能ドライバがあった場合、及びステップ S T 55 において、標準機能ドライバがあった場合、前述したように、アドレス、レンジ数が取得される（S T 40）。

【0091】

上記動作によっても、図 13 と同様に、プラグ・アンド・プレイを実現することができる。

【0092】

尚、上記説明において、拡張機能用のドライバは、ホスト機器 20 内にインストールされ、ホスト機器 20 内を検索すると説明した。しかし、これに限定されるものではなく、拡張機能用のドライバは、メモ리카ード 11 に格納されていてもよい。この場合、メモ리카ード 11 も拡張機能用のドライバの検索対象とされる。

【0093】

図 33 は、カードにオプション設定があり、そのオプションによって機能ドライバが異なる場合、機能ドライバを特定するための情報を示している。図 33 に示すように、機能特定コードは、オプションコードとファンクションコードの 2 種類の情報を示している。ファンクションコードは特定の標準化された機能仕様を示し、オプションの種類もこの機能仕様で定義される。オプションコードは、カードが機能ドライバに影響するオプションを実装しているかどうかを示す情報である。この例は、C M D 48 / 49 をサポートしているかどうかの情報と、C M D 52 / 53 をサポートしているかどうかの情報を示している。オプションコードが 1 バイトであるとする、C M D 48 / 49 を用いるドライバは、“01h”（“h”は 16 進数を示す）で表され、C M D 52 / 53 を用いるドライバは、“02h”で表される。機能ドライバをホストシステムにインストールするとき、機能ドライバ実装コードとして登録される。二つのドライバがインストールされたホストシステムの場合、“01h”と“02h”のオプションコードを持っている。

【0094】

C M D 48 / 49 を用いるように設計されたカードは、オプションコードに“01h”が表示されている。ホストシステムは、オプションコードに基づき C M D 48 / 49 のドライバを選択する。また、C M D 52 / 53 を用いるように設計されたカードは、オプションコードに“02h”が表示されている。ホストシステムは、オプションコードに基づき C M D 52 / 53 のドライバを選択することができる。

【0095】

ここで重要なのは、ホストドライバは、オプション情報に関する知識が不要であり、汎用的なホストドライバが作れる点にある。オプション情報に関する知識は、機能ドライバがホストシステムにインストールされたときに与えられる。ホストドライバにオプション

情報に関する知識は不要であるため、新しいカードをインストールした場合でも、ホストドライバを更新する必要がない。機能仕様は自由にオプションの内容を決めることができ、またこの方法により、ホストシステムにオプションの組み合わせに対応した複数の機能ドライバをインストールしておき、カードのサポート状況に応じて最適な機能ドライバを選択することができる。

【0096】

(S D I Oにおける拡張レジスタアクセス)

図15は、S D I Oにおける拡張レジスタアクセスを示している。

【0097】

S Dメモリカードに対応するホストコントローラは、コマンドC M D 4 8、C M D 4 9を用いて拡張レジスタをアクセスして、拡張機能を制御することができる。すなわち、固定長ブロック、シングルブロック転送をサポートしている。

【0098】

これに対して、S D I Oカードに対応するホストコントローラでもコマンドC M D 4 8、C M D 4 9を用いて拡張レジスタをアクセスできるようにするとともに、拡張レジスタをS D I Oの各ファンクション領域にマッピングすることで、S D I OコマンドC M D 5 2 (ライトコマンド)、C M D 5 3 (データ転送コマンド)からもアクセスすることが可能となる。S D I Oコマンドを用いると、可変ブロック長、マルチブロック転送をサポートでき、ドライバの最適化を図ることができる。コマンドC M D 4 8、C M D 4 9でアクセスする場合、上記S D I O空間上のマッピングに関係なく、拡張レジスタをアクセスすることができる。

【0099】

具体的には、拡張レジスタをS D I Oで使用する場合、拡張レジスタの各ページは、各ファンクション領域にマッピングされる。図15に示す例の場合、拡張レジスタのページ0は、ファンクション領域61のファンクション0にマッピングされ、ページ1、ページ2はファンクション1にマッピングされ、ページ3はファンクション2にマッピングされている。ファンクション0は、各ページの機能レジスタがS D I Oマップのどこに配置されているかを示すアドレス情報を保持している。このため、このアドレス情報を用いることにより、コマンドC M D 4 8、C M D 4 9を用いたドライバだけでなく、コマンドC M D 5 2、C M D 5 3を用いたドライバにより、拡張レジスタの各ページをアクセスすることが可能である。

【0100】

尚、ホスト機器20は、拡張レジスタの第1ページに記載された汎用情報フィールドから、拡張機能に割り当てられた拡張レジスタの位置情報をドライバに渡す。これにより、拡張機能が任意の位置に配置されていても制御可能となる。

【0101】

また、ホスト機器20は、前述したプラグ・アンド・プレイにより、メモリデバイス11とデータ転送が可能となった状態において、前記コマンドC M D 4 8、C M D 4 9、C M D 5 2、C M D 5 3を用いて、拡張機能部としてのS D I Oとデータ伝送が可能となる。

【0102】

上記実施形態によれば、メモリデバイス11のR A M 1 5内に複数ページの拡張レジスタ31を設け、これら拡張レジスタ31のページ0に、特定のドライバを認識するための標準的な汎用情報フィールドを設定している。このため、ホスト機器20は、拡張レジスタ31のページ0の汎用情報フィールドを参照してドライバを設定することにより、プラグ・アンド・プレイを実現することができる。

【0103】

また、拡張レジスタをアクセスするための専用のコマンドC M D 4 8、C M D 4 9を定義することにより、メモリ用のホストコントローラにおいても、追加機能を効率良く制御することができる。

【0104】

しかも、データの転送は、512バイトの固定ブロック長の転送としているため、従来のメモリ用ホストコントローラから拡張レジスタをアクセスするための専用コマンドを発行できる。

【0105】

さらに、コマンドの引数として有効データ長の情報や、ライト時のマスク機能を設定しているため、データの一部を書き換える際に、リード・モディファイ・ライトが不要であり、容易にデータの一部を書き換えることができる。

【0106】

また、SDIOカード対応のホストコントローラにおいては、データポートをサポートしているため、ある特定デバイスに対するデータ転送が行え、かつ拡張レジスタ空間の消費量を低減できる実装が可能となる。

10

【0107】

また、データポートを用いることにより、SDIOにおいて複数ブロックによるバースト転送コマンドをサポートすることができ、メモリ以外のデバイスのデータ転送を効率良く行うことができる。(本実施形態では記載されていないが、メモリにおいても複数ブロックによるバースト転送コマンドを定義すれば複数ブロック転送は可能になる。)

さらに、SDIOカード対応のホストコントローラにおいては、SDIOのアクセスコマンドを用いて、拡張レジスタをアクセスすることにより、小さいブロック長転送とマルチブロック転送とに対応可能となる。このため、さらに最適化したドライバを作ることが

20

【0108】

また、カードベンダーにより、拡張レジスタ上の任意の位置に追加機能を制御するレジスタを割り当てることを可能とし、実装したレジスタのアドレス情報をページ0の汎用情報フィールドから提供している。このため、定義された機能レジスタをリロケータブルに配置することが可能である。このため、従来標準化が必要であったアドレス割り当てを決める作業が不要であり、デバイスの製造を容易化できる。

【0109】

尚、拡張レジスタは、複数ページに限定されるものではなく、1ページとし、1ページ内で、上記ページ0とページ1～7に対応する領域を設定することも可能である。

30

【0110】

(レビジョン確認による使用可能な機能の決定)

上記各機能は、その機能で定義した拡張レジスタ・セットにレビジョンを示すレジスタを具備する。また、機能ドライバは対応するレビジョンをドライバ自身が知っている。ある機能をレビジョンアップで拡張する場合、従来の機能と互換性を維持しつつ機能を拡張することにより互換性を維持することができる。リムーバブルカードを用いる場合、カードの機能レビジョンと、ホストシステムにインストールされている機能ドライバのレビジョンの組み合わせによって使用可能な機能が決定される。

【0111】

図16は、レビジョン管理の例を示している。図16は、カードと機能ドライバのレビジョンに応じて、利用可能な機能の例を示している。例えば3つのレビジョン(A<B<C)がある場合を説明する。この場合、CはBの機能を包含し、BはAの機能を包含するような拡張が行われている。レビジョン管理は、機能ドライバによって行われる。機能ドライバ自身は自分のレビジョンを知っている。利用可能な機能は、図16に示すような組み合わせで決定される。全ての機能ドライバレビジョンで、レビジョンAの機能は使用可能であり、レビジョンBの機能を使うためには、機能ドライバレビジョンがB以上である必要がある。

40

【0112】

(第2の実施形態)

図17、図18は、第2の実施形態に係るリードコマンドCMD48、及びライトコマ

50

ンドCMD49のフィールド構成の一例を示している。尚、図17、図18において、図3、図6と同一部分には同一符号を付し、説明は省略する。

【0113】

図17、図18に示すCMD48、CMD49は、図3、図6に示すCMD48、CMD49において、“RS”と“OFS”の12ビットで構成していたアドレスフィールドを“FNO”、“Addr”により構成される20ビットに拡張し、SDIOとの親和性・互換性を考慮している。

【0114】

“MIO”フィールドは、メモリ空間と、SDIO空間を分離するビットであり、相互に独立して拡張レジスタを定義できる。このため、拡張レジスタを定義するときには双方の干渉を防ぐことができる。“MIO”=0のとき、メモリ用の拡張レジスタをアクセスでき、“MIO”=1のとき、SDIO用の拡張レジスタをアクセスできる。

10

【0115】

“FNO/FID”フィールドは、“MIO”フィールドの値により、“FNO”と“FID”の一方に設定される。“MIO”=1の場合“FNO”は、機能番号を示す3ビットのフィールドであり、“MIO”=0の場合“FID”は、機能識別情報を示す4ビットのフィールドである。ビット数が異なるため別なシンボルで表記している。前記汎用情報フィールドを読む場合は、“FNO/FID”=0を設定する。ホストドライバは、このフィールドを0にセットしておけば良い。“FID”は、メモリ空間では使用しないが、“FNO”はSDIO空間において8個の機能空間を区別するために使用される。

20

【0116】

すなわち、“FNO/FID”（4ビット）は、“MIO”=1のとき、ビット38～36は、“FNO”を示し、ビット35は、常に“0”とされる。

【0117】

また、“FNO/FID”は、“MIO”=0のとき、ビット38～36は、“FID”を示す。“FID”は、メモリ空間を増加せず、機能を識別するために使用される。

【0118】

（メモリ空間を“FID”によって増やしても良く、これは制約ではない。）

カードに機能を実装するとき、“FID/FNO”にユニークな値が割り当てられ、後述するように、汎用情報のフィールド定義に表示される。このため、機能ドライバがデータポートへのコマンド発行時、引数に“FID/FNO”を設定することにより、カードは、コマンドが指定した機能に対するコマンドであることが確認できる。したがって、間違ったデータポートの指定により、誤書き込みによるデータ破壊、誤動作などを防止でき、安全性が担保される。

30

【0119】

また、ホストはアドレス情報から機能を特定しようとするアドレス情報をデコードしなければならないが、“FID/FNO”だけでも機能識別が可能となる。したがって、ホストドライバの制御を簡素化できる。すなわち、同じコマンドが複数の機能により混在して使用されるため、ホストおよびカードにおいて、各機能を識別することが可能なように、“FID/FNO”が設定されている。

40

【0120】

“Addr”フィールド（17ビット）は、アドレスであり、128KBの空間をアクセスできる。“Addr”の上位8ビットは、ページ番号として用いられ、8ビットにより0～7ページのうちの1ページが選択される。下位9ビットで選択されたページ内の512バイトのブロックがアクセスされる。すなわち、“MIO”、“FNO”（“MIO”=1）、“Addr”を用いることにより、拡張レジスタの位置が指定される。

【0121】

図17に示す“Len”フィールド（8ビット）は、有効データ長を示している。

【0122】

また、図18に示すライトコマンド（CMD49）において、“MW”は、マスクライ

50

トモードを指定するビットである。“MW”=0のとき、マスクがディスエーブルとされ、“MW”=1のとき、マスクがイネーブルとされる。

【0123】

また、“Len/Mask”フィールドは、マスクがディスエーブル(“MW”=0)のとき、データ長が16-08ビットの9ビットに設定される。また、マスクがイネーブル(“MW”=1)のとき、データ長は1に設定され、16-08ビットのうちの下位8ビットにより書き込み動作が上述したように制御される。すなわち、8ビットの各ビットは、“1”のとき、レジスタのデータが書き込まれ、“0”のとき、レジスタのビットは変化されず、既にセットされた値が保存される。

【0124】

第2の実施形態では、SDIOコマンドCMD52、CMD53がアクセス可能な空間と、CMD48、CMD49がアクセス可能なSDIO空間を一致させることができる。すなわち、どちらのコマンドを用いても同じ拡張レジスタ・セットにアクセスすることが可能になる。

【0125】

(拡張レジスタのリードコマンド、第1の動作モード)

図19は、拡張レジスタのリードコマンド(CMD48)の第1の動作モードによる拡張レジスタのリード動作の例を示している。

【0126】

図19に示すように、メモリデバイス11は、ホスト機器20からコマンド(CMD48)を受け取ると、レスポンス(R1)をホスト機器20に返し、その後、拡張レジスタ31から512バイトのデータブロックを読み出す。

【0127】

具体的には、コマンド(CMD48)の引数としての“FNO”(MIO=1)、“Addr”により、ページ内の読み出すべきデータの位置が指定され、読み出すべき有効データ長が“Len”で指定される。このようにして指定された拡張レジスタ内のデータが、512バイトのデータブロックの先頭にセットされ、読み出される。512バイトのデータブロックのうち、“Len”で指定されたデータ長を超えるデータは、無効データとなる。データブロックの最後にはCRCコードが付加され、正しくデータが受け取れたかをチェックすることが可能とされている(無効データを含めてチェックを行う)。

【0128】

図20は、第2の動作モードによるデータポートリードの動作の例を示している。

【0129】

メモリデバイス11は、このコマンド(CMD48)を受け取ると、レスポンス(R1)を返し、その後512バイトのデータブロックを返す。

【0130】

メモリデバイス11は、コマンドの引数の“FID/FNO”が割り当てられた拡張レジスタ・セットに一致するかどうかを検証する。拡張レジスタ・セットは、“FNO”(“MIO”=1)と“Addr”によって特定される。これらが一致している場合、コマンドの引数“Addr”により、拡張レジスタの選択されたページ内の位置が指定される。データポートは、拡張レジスタマップ上において、1バイトのアドレスを占有するだけで良い。データポートであるかどうかはアドレスのデコードによって識別すれば良く、実際に1バイト幅のポートを通してデータが伝送される必要はないので、データ伝送性能には影響しない。このデータポートに割り当てられたデバイスから1ブロック(512バイト単位)のデータをリードすることができる。すなわち、1回当たり、1ブロック(512バイト単位)のデータをリードすることができる。この読み出されたデータは、例えばバッファ16に保持され、ホスト機器20によって読み出される。

【0131】

続いて同じデータポートをリードすると、続きの512バイトのデータを読み出すことができる。データポートから読み出すデータを何処から取ってくるかは、拡張機能の仕様

10

20

30

40

50

によって自由に定義ができる。データポート制御は、例えば、拡張レジスタ上に制御レジスタを定義して制御することができる。512バイトのデータブロックの最後にCRCコードが付加され、正しくデータが受け取れたか否かがチェック可能とされている。

【0132】

また、上記検証の結果、“FID/FNO”が機能に割り当てられた値と不一致である場合、データ転送動作は実行されず、データブロックは転送されない。

【0133】

(拡張レジスタのライトコマンド、第1の動作モード)

図21は、拡張レジスタのライトコマンドの一例を示している。

【0134】

メモリデバイス11は、このコマンド(CMD49)を受け取ると、レスポンス(R1)を返し、その後、512バイトのデータブロックを受け取る。

【0135】

メモリデバイス11は、データブロックが正しく受け取れたかどうかを示すCRCコードをホスト機器20に返す。その後、このコマンドの処理が終了するまでビジーを返し、ホスト機器20が次のコマンドを発行できるタイミングを知らせる。データブロックは、バッファ16に保持されている。

【0136】

ライトコマンド(CMD49)において、リードコマンド(CMD48)と同一部分には同一符号を付している。ライトコマンドとリードコマンドは、“index”により区別される。“FNO”(“MIO”=1)と17ビットの“Addr”により、拡張レジスタのページと選択されたページ内のデータの位置が指定される。さらに、9ビットの“Len”フィールドにより、512バイトの拡張レジスタに書き込むデータ長が指定される。したがって、512バイト内の任意のデータ長(バイト単位)のデータを拡張レジスタの任意のページと場所に書き込むことが可能である。

【0137】

ライトコマンド(CMD49)は、上記のように、コマンドの引数の中にマスクレジスタが設けられている。すなわち、“Mask”は、8ビット長のマスクレジスタを示している。このマスクレジスタにより、1バイトのデータのライトにおいて、ビット単位のオペレーションが可能となり、特定のビットにのみデータを書き込むことが可能となる。このため、1バイト内のビットオペレーションであれば、リード・モディファイ・ライトを行う必要がない。マスクレジスタは、データ長が1バイトのとき、すなわち、“Mask”の上位1ビットが“1”のとき有効となる。

【0138】

(拡張レジスタのライトコマンド、第2の動作モード)

図22は、第2の動作モードによるライトデータポートの動作の例を示している。

【0139】

メモリデバイス11は、このコマンド(CMD49)を受け取ると、レスポンス(R1)を返す。その後、メモリデバイス11は、コマンドの引数の“FID/FNO”が拡張レジスタ・セットに一致するかどうかを検証する。拡張レジスタ・セットは、“FNO”(“MIO”=1)と“Addr”によって特定される。これらが一致している場合、コマンドの引数“Addr”により、拡張レジスタの選択されたページ内の位置を指定し、512バイトのデータブロックを受け取る。

【0140】

次いで、メモリデバイス11は、データブロックが正しく受け取れたかどうかを示すCRCコードをホストに返す。その後、このコマンド処理が終わるまでビジーを返し、ホスト機器20が次のコマンドを発行できるタイミングを知らせる。データブロックは、バッファ16に保持されている。

【0141】

コマンド処理において、コマンドの引数“Addr”により、拡張レジスタ内のページ

10

20

30

40

50

と位置が指定され、データポートが指定される。データポートは、拡張レジスタマップ上において、1バイトのアドレスを占有するだけでよい。このデータポートに、バッファ16に保持された1ブロック（512バイト単位）のデータがある割り当てたデバイスにライトすることができる。すなわち、1回当たり、1ブロックのデータをライトすることができる。

【0142】

続いて同じデータポートをライトすると、続く512バイトのデータを割り当てたデバイスに書き込むことができる。データポートのデータを何処に渡すかは、拡張機能の仕様によって自由に定義ができる。データポート制御は、例えば、拡張レジスタ上に制御レジスタを定義して制御することができる。

10

【0143】

また、上記検証の結果、“FID/FNO”が機能に割り当てられた値と不一致である場合、データ転送動作は実行されず、データブロックは破棄される。

【0144】

（汎用情報フィールドの使用例）

図23は、第2の実施形態に係るFIDの指定に関する例を示した図である。汎用情報フィールドの意味は、図10と同じである。異なる点は、拡張アドレス、長さフィールドのフォーマットで、“FID/FNO”の値を設定するために4ビットフィールドが確保されていることにある。各機能毎にユニークな“FID/FNO”が設定される。カードに実装された各機能は自分の“FID/FNO”を認識している。

20

【0145】

（リードコマンド（CMD48）の動作）

図24は、図19、図20に示すリードコマンド（CMD48）に対応するメモリデバイス11内のコントローラ11aの動作を示している。

【0146】

リードコマンドを受けると、CPU13により、コマンドの引数“FID/FNO”が割り当てられた拡張レジスタ・セットに一致するかどうかを検証される（ST51）。拡張レジスタ・セットは、“FNO”（“MIO”=1）と“Addr”によって特定される。検証の結果、両者が一致している場合、コマンドの引数の“Addr”が解析され、リードコマンドがデータポートかどうか判別される（ST52）。すなわち、“FNO”（“MIO”=1）と“Addr”でデータポートとして定義されたアドレスであるかどうか判別される。

30

【0147】

この結果、アドレスがデータポートとしてのアドレスではなく、コマンドが拡張レジスタのリードであると判別された場合、第1の動作モードにより、位置“Addr”に基づき、拡張レジスタ31の選択されたページからデータ長“Len”のデータが取得される（ST53）。この取得されたデータ長“Len”のデータは、バッファ16の512バイトのデータブロックにセットされる（ST54）。

【0148】

一方、ステップST52において、リードコマンドがデータポートであると判別された場合、第2の動作モードにより、拡張レジスタ31の選択されたページの予め設定された位置のデータポートを介して、例えばSDIO19の特定の機能（ファンクション）から512バイトのデータが取得される（ST55）。この取得されたデータは、バッファ16の512バイトのデータブロックにセットされる（ST56）。

40

【0149】

上記ステップST51の判別の結果、データポートのコマンドでは無い場合、処理が終了される。

【0150】

（ライトコマンド（CMD49）の動作）

図25は、上記ライトコマンド（CMD49）に対応するメモリデバイス11内のコン

50

トローラ 11a の動作を示している。

【0151】

ライトコマンドを受けると、CPU 13（コマンド処理部 14a）により、コマンドの引数“FID/FNO”が割り当てられた拡張レジスタ・セットに一致するかどうか検証する（ST 61）。拡張レジスタ・セットは、“FNO”（“MIO”=1）と“Addr”によって特定される。検証の結果、両者が一致している場合、コマンドの引数“Addr”が解析され、ライトコマンドがデータポートかどうか判別される（ST 62）。すなわち、“FNO”（“MIO”=1）と“Addr”で予め設定されたデータポートの位置かどうか判別される。

【0152】

この結果、ライトコマンドがデータポート以外と判別された場合、コマンドの引数“MW”が“1”かどうか、すなわち、マスクライトかどうか判別される（ST 63）。 10

【0153】

この判別の結果、マスクライトではないと判別された場合、拡張レジスタ処理部 14c により拡張レジスタのライト処理が行われる。すなわち、バッファ 16 のデータブロックから“Len”により指定された長さのデータが取得される（ST 64）。この取得されたデータは、“Addr”に基づき、拡張レジスタの選択されたページの指定された位置にセットされる（ST 65）。

【0154】

一方、上記ステップ ST 63 において、“MW”=“1”であり、マスクライトであると判別された場合、拡張レジスタ処理部 14c によりバッファ 16 のデータブロックから 1 バイトのデータが取得されるとともに、引数から 1 バイトのマスクが取得される（ST 66）。 20

【0155】

次いで、1 バイトのデータと、1 バイトのマスクにより、図 7A、7B、7C に示すマスク動作が実行され、“Addr”で指定された拡張レジスタの所定のページ、所定の位置に 1 バイトのマスク動作が実行されたデータがセットされる（ST 67）。

【0156】

また、上記ステップ ST 62 において、ライトコマンドがデータポートであると判別された場合、バッファ 16 のデータブロックから 512 バイトのデータが取得される（ST 68）。この取得されたデータは、拡張レジスタ 31 の指定されたページの位置のデータポートを介して、例えば S D I O 19 の特定の機能に送られる（ST 69）。 30

【0157】

上記ステップ ST 61 の判別の結果、データポートのコマンドでは無い場合、処理が終了される。

【0158】

（CMD 58、CMD 59）

図 26、図 27 は、データの転送効率を良くするためのマルチブロック転送コマンドを示すものであり、図 26 は、CMD 58（Multi-Block Read）、図 27 は、CMD 59（Multi-Block Write）を示している。 40

【0159】

CMD 58、CMD 59 の引数は、CMD 48、CMD 49 と類似しているが一部定義が異なる。また、CMD 58 には、CMD 48 の引数“Len”がなく、CMD 59 には、CMD 49 の引数“MW”及び“Len/Mask”がない。これは、マルチブロック転送がデータポートに対する転送を想定しているためである。CMD 58、CMD 59 はオプションコマンドであり、データポートは、複数のシングルブロック転送コマンド CMD 48、CMD 49 で代替できるようになっている。

【0160】

マルチブロック転送は、データポートを経由したデータ転送を想定している。したがって、このコマンドのアドレスが、拡張レジスタ空間の中のデータポートとして定義された 50

アドレスと一致した場合のみ、このコマンドが有効になる。このため、このコマンドが通常の拡張レジスタに対して実行された場合、エラーとなりデータ転送は実行されない。

【0161】

“FID/FNO”フィールド（4ビット）には、発行されたコマンドがどの機能で用いるコマンドであるかを識別するためのコードが設定される。このため、“FID”フィールドを用いることにより、この値で機能を識別することができ、実装が容易になる。アドレス“Addr”フィールドでもファンクションは識別可能である。しかし、カードによって割り当てられるアドレスが異なるため、ホストドライバがアドレスから機能を特定することは管理が難しいという問題がある。このため、ホストドライバは、“FID”フィールドの値に基づき、機能毎にホストシステムに実装されているデータバッファなどを切り替え制御に用いることができる。

10

【0162】

CMD 58/59の引数には、データ長を指定する“Len”フィールドがない。この理由は、長いデータ伝送は、長いブロック数を指定する必要があるため、この情報はリード/ライトコマンドの引数には指定しきれないためである。したがって、CMD 58/59を発行する前にデータ転送に必要なブロック数を指定しておく必要がある。このため、例えば拡張レジスタにブロック数を設定するレジスタ定義し、CMD 49で設定する方法や、ブロック数を設定するコマンド（CMD 23）を、CMD 58/59を発行する直前に発行する、などの方法が用いられる。

【0163】

20

拡張レジスタにデータを設定する場合、各機能で独立して設定することができ、他の機能から影響されない。共通のブロック数コマンド（CMD 23）を用いる場合、メモリマルチブロックコマンドに対するブロック数の設定と区別が必要である。このため、各CMD 58/59の直前に発行する必要があるため、ホストドライバは、CMD 23の直後に他のコマンドが発行されないように順番を管理する必要がある。

【0164】

ホストは、複数機能カード/デバイスのひとつの機能を特定するためには、初期化で得られるカード固有アドレス（RCA:Relative Card Address）やデバイスID、前記“MIO”と前記“FID/FNO”情報が必要である。

【0165】

30

図28は、第2の実施形態に係る汎用情報フィールドの表示位置を示している。図28（a）に示すメモリ空間では、拡張レジスタのページ0に配置され、図28（b）に示すSDIO空間では、従来のレジスタとコンフリクトしない特定の位置に配置される。例えば図28（b）において、SDIO用汎用情報は、機能（ファンクション）0の“008FFh”～“00800h”（512バイト）（“h”は16進数を示す）0に配置されている。

【0166】

図29は、第2の実施形態に係るメモリ空間とSDIO空間の対応関係の一例を示している。図29において、図15と同一部分には同一符号を付している。

【0167】

40

メモリ拡張レジスタは、CMD 48/49によりアクセスが可能である。具体的には、512バイトの固定ブロック長で、シングルブロック転送が行われる。さらにデータポートの場合、CMD 58/59でマルチブロック転送を行うことができる。SDIO拡張レジスタは、CMD 48/49だけでなく、CMD 52/53によってもアクセスが可能である。CMD 53は可変長コマンドのため、データポートでなくてもSDIO拡張レジスタのアクセスに用いることができる。

【0168】

（機能ドライバインストール）

SDIO機能（CMD 52/53）が使用可能かどうかは、ホストシステムがサポートする機能によって決まる。SDIOをサポートしないホストは、CMD 48/49、CM

50

D 5 8 / 5 9 を用いた機能ドライバをインストールする。S D I O をサポートするホストシステムは、さらに C M D 5 2 / 5 3 を用いた機能ドライバもインストールすることが可能となる。

【0169】

尚、C M D 5 3 は、例えば可変ブロック長、及び複数ブロック転送をサポートするリード又はライトが可能なコマンドであり、C M D 5 2 は、例えばデータを持たず、引数とレスポンスで1バイトのデータのリード、ライトを可能とするコマンドである。

【0170】

C M D 4 8 / 4 9 の S D I O 拡張レジスタ空間は、C M D 5 2 / 5 3 の空間と等価である。C M D 5 3 が可変ブロック長とマルチブロック転送をサポートするため、データ転送は最適化された S D I O ドライバを使用することにより一層効率的に実行される。

10

【0171】

C I S (Card Information Structure) を参照することなく、C M D 4 8 / 4 9 をサポートするホストが、その情報を参照するように、S D I O の汎用情報は、前記した機能0の特定の位置から見られる。

【0172】

(機能ドライバの選択)

S D I O 対応カードに関して、C M D 5 2 / 5 3 を用いた機能ドライバがインストールされている場合、その機能ドライバが使用され、インストールされていない場合、C M D 4 8 / 4 9、C M D 5 8 / 5 9 を用いた機能ドライバが使用される。

20

【0173】

S D I O 非対応カードに関して、C M D 4 8 / 4 9、C M D 5 8 / 5 9 を用いた機能ドライバが使用される。

【0174】

(S D I O の初期化動作)

図30は、コンボカードにおけるS D I O の初期化動作を概略的に示している。

【0175】

従来、S D I O の初期化シーケンスは、最初にS D I O 初期化コマンド (C M D 5) を実行しないとS D I O 機能が有効にならないという定義になっている。このため、コンボカードでメモリを使用している場合でも、S D I O を使用する時点で再初期化が必要になるため、ホストにとって使いにくい仕様となっていた。

30

【0176】

通常、I / O 機能は、その機能を使う直前に初期化するのがシステムリソースを無駄にしないためや、消費電力を無駄に消費しないために好ましい。機能が初期化されるタイミングとしては、機能を用いるアプリケーションが起動したタイミングで行うのが良い。

【0177】

また、再初期化では、カード固有アドレス (R C A) の変更が行われるため、メモリのアクセス方法にも影響を与えてしまう。メモリ制御に影響を与えず、S D I O 機能を有効にするため、メモリ初期化シーケンスを基本として、S D I O 機能は後で追加できることが望ましい。

40

【0178】

そのため、図30に示すように、メモリデバイス11が起動され、初期化されると (S T 7 1)、コマンド (C M D 3) が発行され、カード固有アドレス (R C A) が取得される (S T 7 2)。この後、コマンド (C M D 7) が発行され (S T 7 3)、メモリデバイス11が転送状態、すなわち、メモリが使用可能な状態に設定される (S T 7 4)。この状態にて、S D I O の初期化コマンド (C M D 5) が発行される (S T 7 5)。これにより、S D I O が初期化され、C M D 5 2 と C M D 5 3 の受付が可能とされる (S T 7 6)。

【0179】

尚、カード固有アドレス (R C A) はメモリの初期化で取得した値を維持する。これは

50

初期化方法の追加であり、従来のS D I O初期化シーケンスでも初期化は可能であり互換性がある。

【0180】

上記構成によれば、機能を用いるアプリケーションが起動したタイミングにおいて、機能が初期化されるため、メモリ制御に影響を与えることなく、各機能を初期化することができる。

【0181】

(機能ドライバインターフェース)

従来、S D I Oは、共通レジスタに必要な制御ビットを割り当てて制御していた。これを行うためハードウェアを実装する必要がある。専用の機能ドライバで機能を制御する場合、実装は自由に行うことができる。このため、必ずしも共通レジスタで制御を行う必要はない。従来共通レジスタで行っていた制御を機能ドライバのA P I (Application Program Interface)として定義すれば、制御をソフトウェア化することができる。A P Iレベルを標準化することにより、ハードウェアによる標準化は不要であり、実装を容易化することができる。

10

【0182】

A P Iの例を以下に示す。

【0183】

(1) Initialize Function

機能を初期化するために、ホストドライバから呼び出す。

20

【0184】

(2) Abort/Reset Function

機能のアボート、又はリセット。

【0185】

(3) Get Function Information

機能レビジョンの読み出し。

【0186】

機能情報の読み出し(サポート情報など)。

【0187】

割り込み情報の読み出し (ポーリング)。

30

【0188】

(4) Power Consumption Control

機能が実装しているパワーモード情報。

【0189】

(5) Power Off Notification

電源を切っても良いタイミングを通知する。

【0190】

(6) Application Interface

アプリケーションとの間の制御インターフェース。

【0191】

40

特に、複数機能が実装されているカードにおいて、カードの電源を切る場合、各機能が電源を停止できる状態としてから、ホストはカード電源を切る必要がある。Power Off Notificationはこの制御に用いられるA P Iである。

【0192】

図31は、メモリデバイス11としてのS Dカードとホスト機器20の機能インターフェースの関係を概略的に示している。

【0193】

ホスト機器20は、ホストコントローラ21、ホストドライバ71、ファイルシステム72、メモリアプリケーション73、機能ドライバ74、機能アプリケーション75により構成されている。また、メモリデバイス11としてのS Dカードは、拡張レジスタ31

50

、例えばS D I Oからなる機能ハードウェア19を含んでいる。

【0194】

ホスト機器20において、ホストドライバ71は、機能ドライバ74を見つけ、ロードする機能がサポートしている。すなわち、ホストドライバ71は、拡張レジスタの汎用情報フィールドを参照して、機能ドライバ74を検出し、その機能ドライバを実行することにより、拡張機能を使用することができる。また、拡張機能レジスタ31を制御する機能ドライバ74と機能アプリケーション74との間は、機能仕様により定義されたA P Iによりコミュニケーションされる。

【0195】

S Dカードは、標準化のため、機能仕様により定義された拡張機能レジスタ31と、ホストドライバ71が機能ドライバ74を見つけてロードすることができるように、上述した汎用情報フィールドを有している。

【0196】

ホストコントローラ21とメモリデバイス11は、上述したコマンドC M D 4 8 / 4 9等を用いてコミュニケーションされる。

【0197】

上記構成によれば、従来共通レジスタで行っていた制御を機能ドライバのA P Iとして定義することにより、制御をソフトウェア化することができる。また、A P Iレベルを標準化することにより、ハードウェアによる標準化が不要であり、実装を容易化することができる。

【0198】

また、ホストドライバ71は、拡張レジスタの汎用情報フィールドを参照して、機能ドライバ74を検出し、その機能ドライバを実行することにより、拡張機能を使用することができる。したがって、ホスト機器20は、容易に拡張機能を使用することが可能である。

【0199】

(“ F I D ”によるデータバッファの制御)

メモリデバイス11は、アドレス情報を識別することにより、どの機能に対するコマンドであるかを判別することが可能ではある。しかし、アドレス範囲は機能によって異なるため、ホスト機器20はアドレスから機能を識別するのが難しい。

【0200】

このため、上述したように、“ F I D / F N O ”を用いることにより、ホスト機器20は簡単に機能を識別することが可能である。

【0201】

また、“ F I D / F N O ”を用いることにより、例えばホスト機器20の複数のバッファを制御することが可能である。

【0202】

図32に示すように、ホスト機器20は、メモリデバイス11としてのS Dカードの複数の機能に対してデータ転送を行う場合、各機能に対応して独立したバッファ81、82を持っている。これらバッファ81、82は、マルチプレクサ(M U X) 83を介してホストコントローラ21に接続されている。このマルチプレクサ83を“ F I D / F N O ”により制御することにより、各機能に対応したバッファ81、82を選択することができる。

【0203】

すなわち、ホスト機器20は、C M D 5 8 / 5 9に設定した“ F I D / F N O ”に応じて、対応するバッファ81、82をマルチプレクサ83により選択することができる。

【0204】

ホストコントローラ21から例えばリードコマンドC M D 5 8が発行された場合、メモリデバイス11の対応する機能の拡張レジスタから読み出されたデータは、ホストコントローラ21を介してマルチプレクサ83に供給される。マルチプレクサ83は、“ F I D

10

20

30

40

50

／FNO”に基づき、受け取ったデータをバッファ81、82のうちの一方に供給する。

【0205】

また、ホストコントローラ21から例えばライトデータコマンドCMD59が発行された場合、マルチプレクサ83は、“FID／FNO”に基づき、バッファ81、82のうちの一方から選択されたデータをホストコントローラ21に供給し、ホストコントローラ21は、メモリデバイス11にデータを転送する。メモリデバイス11は、“FID／FNO”に基づき、対応する機能の拡張レジスタにデータを供給する。

【0206】

このように、“FID／FNO”を用いて、マルチプレクサ83を制御することにより、各機能に対応したバッファ82、83を確実に選択することが可能である。

10

【0207】

(第3の実施形態)

上記のように、拡張レジスタをメモリデバイスとしてのSDカードの内部に設け、ホスト機器から供給されるコマンドCMD48、CMD49、CMD58、CMD59によって拡張レジスタにデータを書き込んだり、拡張レジスタからデータを読み出したりすることが可能されている。

【0208】

第3の実施形態では、ホスト機器20から拡張機能部19に必要なコマンドや、メッセージを転送したり、拡張機能部19からのレスポンスをホスト機器20に転送したりする具体的な方法について説明する。

20

【0209】

第3の実施形態は、拡張機能の動作に必要なコマンドとレスポンスを定義するとともに、これらコマンドとレスポンスを受け渡す場所を定めている。さらに、ホスト機器20と拡張機能部19との間で、長いデータを転送する場合、必要に応じて例えばコマンドペイロードと称するデータを追加し、データを転送する方法を定義する。

【0210】

図34は、第3の実施形態を示すものであり、ホスト機器20とメモリデバイス11としてのSDカードとの関係を、ファームウェアレベルで概略的に示している。図34において、図1と同一部分には、同一符号を付している。

【0211】

ホスト機器20は、拡張機能アプリケーションを20-1、カードコマンド発行部20-1を有している。

30

【0212】

メモリデバイス11は、コントローラ11a、NANDフラッシュメモリ、拡張機能部19を有している。コントローラ11aは、カードコマンド制御部11-1、拡張レジスタ31、メモリ制御部11-2を有している。

【0213】

ホスト機器20からメモリデバイス11の拡張機能部19にコマンドやメッセージなどのデータを送受信する場合、カードコマンド発行部20-2は、CMD48／49／58／59のいずれかと、コマンドヘッダ20a、レスポンスヘッダ20c、及び必要に応じてコマンドペイロード20b、レスポンスペイロード20dを発行する。

40

【0214】

すなわち、ホスト機器20からメモリデバイス11にデータを転送する場合、コマンドヘッダ20aと、必要に応じてコマンドペイロード20bが用いられる。コマンドヘッダ20aは、CMD49によりメモリデバイス11に供給され、コマンドペイロード20bは、CMD59によりメモリデバイス11に供給される。

【0215】

一方、ホスト機器20により、メモリデバイス11の拡張機能からのレスポンスを受け取る場合、レスポンスヘッダ20cと、必要に応じてレスポンスペイロード20dが用いられる。レスポンスヘッダ20cは、CMD48を用いて受け取られ、レスポンスペイロ

50

ード 20 d は、C M D 5 8 (Multi-Block Read) を用いて受け取られる。

【0 2 1 6】

(コマンド C M D 5 8、5 9)

図 3 5 は、リードコマンドとしての C M D 5 8 を示し、図 3 6 は、ライトコマンドとしての C M D 5 9 を示している。図 3 4、図 3 5 において、C M D 5 8 / 5 9 の引数は、図 2 6、図 2 7 と類似する。このため、図 2 6、図 2 7 と異なる部分についてのみ説明する。

【0 2 1 7】

図 2 6、図 2 7 に示す例の場合、長いデータを転送する際、C M D 5 8 / 5 9 を発行する直前において、ブロック数を設定する C M D 2 3 を発行した。

10

【0 2 1 8】

これに対して、第 3 の実施形態において、C M D 5 8 / 5 9 は、C M D 2 3 を用いることなく、長いデータを転送可能としている。

【0 2 1 9】

図 3 5、図 3 6 において、図 2 6、図 2 7 と異なる部分は、引数 “ B U S ” (Block Unit Select) と “ B U C ” (Block Unit Count) である。“ F N O ” は、“ F I D / F N O ” と実質的に同一であり、“ M I O ” により選択されたメモリ空間、又は S D I O 区間における機能空間を区別するために使用される。

【0 2 2 0】

“ B U S ” は、ブロックユニットのサイズを指定するフィールドである。“ B U S ” = “ 0 ” の場合、ブロックユニットサイズは、5 1 2 バイトであり、“ B U S ” = “ 1 ” の場合、ブロックユニットサイズは、3 2 K バイトである。3 2 K バイトは、6 4 ブロックデータ (6 4 × 5 1 2 バイト) が 1 ブロックユニットとして扱われることを示している。

20

【0 2 2 1】

“ B U C ” は、ブロックユニット数を指定するフィールドである。このフィールドで指定するブロックユニットサイズは、“ B U S ” で指定されたブロックユニットサイズによって計算して転送するデータの合計のサイズを計算する。ただし、データ転送時のデータサイズの単位となるブロックは B U S に関らず、5 1 2 バイト固定である。

【0 2 2 2】

また、後述するように、“ A D D R ” (アドレスフィールド) において指定されたアドレスにより、拡張レジスタのポートが指定される。

30

【0 2 2 3】

(コマンドの送受信パターン)

図 3 7 は、ホスト機器 2 0 とメモリデバイス 1 1 との間において送受信されるコマンドのパターンを示している。コマンドの送受信パターンとして、例えば 3 つのタイプがある。

【0 2 2 4】

図 3 7 (a) に示すタイプ 1 は、C M D 4 9 によりコマンドヘッダ 2 0 a をホスト機器 2 0 からメモリデバイス 1 1 に送信し、C M D 4 8 によりメモリデバイス 1 1 からのレスポンスヘッダをホスト機器 2 0 で受信する場合を示している。

40

【0 2 2 5】

このタイプ 1 のパターンは、短いデータの送受信に適している。

【0 2 2 6】

図 3 7 (b) に示すタイプ 2 は、C M D 4 9 によりコマンドヘッダ 2 0 a をホスト機器 2 0 からメモリデバイス 1 1 に送信し、この後、C M D 5 9 によりコマンドペイロード 2 0 b をホスト機器 2 0 からメモリデバイス 1 1 に送信する。この後、C M D 4 8 によりメモリデバイス 1 1 からのレスポンスヘッダ 2 0 c をホスト機器 2 0 で受信する場合を示している。

【0 2 2 7】

コマンドヘッダ 2 0 a の送信とコマンドペイロード 2 0 b の送信の順番は、この限りで

50

はなく、先ず、コマンドペイロード 20 b を送信し、次いで、コマンドヘッダ 20 a を送信してもよい。

【0228】

このタイプ 2 のパターンは、512 バイトより長いデータを拡張機能に送信する場合に適している。

【0229】

図 37 (c) に示すタイプ 3 は、CMD 49 によりコマンドヘッダ 20 a をホスト機器 20 からメモリデバイス 11 に送信し、この後、CMD 48 によりメモリデバイス 11 からのレスポンスヘッダ 20 c をホスト機器 20 で受信し、次いで、CMD 58 によりメモリデバイス 11 からのレスポンスペイロード 20 d をホスト機器 20 で受信する場合を示している。

10

【0230】

ここで、レスポンスヘッダ 20 c の受信とレスポンスペイロード 20 d の受信の順番は、この限りではなく、先ず、レスポンスペイロード 20 d を受信し、次いで、レスポンスヘッダ 20 c を受信してもよい。

【0231】

このタイプ 3 のパターンは、512 バイトより長いデータを拡張機能から受信する場合に適している。

【0232】

コマンドペイロード 20 b やレスポンスペイロード 20 d を必要とするかどうかは、後述するように、コマンドヘッダ 20 a により指定される。

20

【0233】

図 38 は、コマンドヘッダ 20 a、コマンドペイロード 20 b、レスポンスヘッダ 20 c、レスポンスペイロード 20 d のデータ構造の例を示している。

【0234】

コマンドヘッダ 20 a は、512 バイトにより構成され、例えば“オペレーションコード(OPコード)”、“rsv”、“アークメント長”、“ペイロード長”、“アークメント”、“パディング”を含んでいる。コマンドヘッダ 20 a のデータ構造は、後述するように、“OPコード”の内容により、“アークメント”の内容が変更される。

【0235】

“OPコード”は、拡張機能を動作させるためのコマンドのコードを示している。さらに、後述するように、コマンドペイロード 20 b やレスポンスペイロード 20 d を必要とするかどうかは、“OPコード”内において指定される。

30

【0236】

“アークメント長”は、“アークメント”のデータ長を示し、“ペイロード長”は、ペイロードのデータ長を示している。

【0237】

“パディング”は、コマンドヘッダの長さを機器のデータ転送処理単位(512 バイト)に揃えるためのデータであり、データそのものに意味はない。コマンドペイロード 20 b、レスポンスヘッダ 20 c、レスポンスペイロード 20 d の“パディング”も、コマンドヘッダ 20 a の“パディング”と同様である。

40

【0238】

コマンドペイロード 20 b は、512 バイト×N (N は、1 以上の自然数) のデータ長を有し、このデータ長は、コマンドヘッダ 20 a のペイロード長により管理される。コマンドペイロード 20 b は、例えば実データとしての“コマンドペイロード”と“パディング”とにより構成されている。“コマンドペイロード”の内容は、“OPコード”に応じて変わるものである。

【0239】

例えば“OPコード”が拡張処理部 19 を介して NAND フラッシュメモリ 18 にデータを書き込む処理を指示する場合、コマンドペイロードは、書き込みデータである。

50

【0240】

また、拡張処理部19が認証処理機能を有する場合において、“OPコード”が暗号化キープロックのデータを書き込む指示である場合、“コマンドペイロード”は、暗号化キープロックのデータとなる。

【0241】

レスポンスヘッダ20cは、512バイトにより構成されている。レスポンスヘッダ20cは、例えば“レスポンスコード”、“レスポンスデータ長”、“レスポンスデータ”、“パディング”を含んでいる。

【0242】

レスポンスペイロード20dは、“レスポンスペイロード”と“パディング”により構成されている。“レスポンスペイロード”の内容は、レスポンスの内容に従って変化する。レスポンスが、拡張機能を介してNANDフラッシュメモリのデータを読み出した結果である場合、“レスポンスペイロード”の内容は、読み出しデータである。

10

【0243】

(コマンドヘッダ、ペイロード、及びレスポンスヘッダの例)

図39、図40は、コマンドヘッダ、ペイロード、及びレスポンスヘッダの例を示している。

【0244】

図39は、拡張機能がメモリデバイス11の認証処理で使用される暗号化キーを処理する機能を有する場合の例を示している。

20

【0245】

メモリデバイス11の認証処理で使用される暗号化キープロック(Encrypted Key Block: EKB)と称するデータをメモリデバイス11に書き込む場合において、EKBをメモリデバイスに書き込むため、“Write EKB”と称するコマンドを送信する例を示している。

【0246】

図39(a)は、この場合のコマンドヘッダ20aの一例を示すものである。

【0247】

“OP Code”(OPコード)は“80h”であり、“Write EKB”コマンドを示している。

30

【0248】

“Reserved”は、将来の拡張目的や機器がデータ処理し易いように、データの位置調整のために設定されている。

【0249】

“Argument Length”は、このコマンドのアーギュメントの長さを示し、この例では“04h”が設定されている。

【0250】

“Payload Length”は、ペイロードの長さを示し、“N”が設定されている。

【0251】

“Block Number”から“Padding data”まではアーギュメントとして扱われる。“Block Number”は記録するEKBの番号を示し、“n”は、複数のEKBのうちの1つを区別するための識別番号である。

40

【0252】

“Reserved”及び“Padding Data”は、前述した通りである。

【0253】

図39(b)は、この例におけるコマンドペイロード20bの一例を示している。“Write EKB”の場合、“コマンドペイロード”として、EKBデータそのものを転送する。コマンドペイロード20bの長さは、前述したコマンドヘッダ20aの“Payload Length”で規定される。メモリデバイス11は、コマンドペイロード20bを受け取る前に、コマンドヘッダaにより“ペイロード長”が通知されることにより、記録のための事前準備を

50

行うことができる。

【0254】

図39(c)は、この例におけるレスポンスヘッダ20cの一例を示している。“Write EKB”の場合、レスポンスヘッダ20cは、“Response Code”と“Padding Data”により構成される。“Response Code”には、EKBデータの書き込み処理結果が示される。例えば書き込みが成功した場合、“Response Code”として“00h”が設定される。

【0255】

図40は、拡張機能がメモリデバイス11の認証処理で使用されるEKBデータをメモリデバイスから読み出す場合に使用する“Read EKB”と称するコマンドを送信する例を示している。

10

【0256】

図40(a)は、この場合のコマンドヘッダ20aの一例を示すものである。“OP Code”、“Reserved”、“Argument Length”、“Payload Length”の意味は、図39に示す“Write EKB”と同様である。

【0257】

“OP Code”は“81h”であり、“Read EKB”コマンドを示している。

【0258】

“Argument Length”は、この例の場合、“0Ch”に設定されている。

【0259】

“Payload Length”は、“N”が設定されている。

20

【0260】

“Block Number”から“Padding data”まではアーギュメントとして扱われる。“Block Number”は、記録されたEKBの番号“n”を示している。

【0261】

“EKB Offset”は、読み出すEKBの読み出し開始アドレスを示している。

【0262】

“EKB Length”は、読み出すEKBの長さを示している。

【0263】

図40(b)は、この例におけるレスポンスペイロード20bの一例を示している。“Read EKB”の場合、レスポンスペイロード20bは、コマンドヘッダ20aで指定された“Block Number”、“EKB Offset”、“EKB Length”に基づき、拡張機能部19から読み出されたEKBデータをホスト機器20に転送する。このため、“レスポンスペイロード”は、EKBデータそのものである。

30

【0264】

レスポンスヘッダ20cは、“Read EKB”の場合の“レスポンスヘッダ”は、“Response Code”と“Padding Data”により構成される。“Response Code”には読み出し処理の結果が示される。例えば読み出し処理が成功した場合、“00h”が設定される。

【0265】

この例において、“Response Data”は使用されなかったが、EKBデータを転送するとき、EKBデータのダイジェストデータを“Response Data”に記録してデータ転送することもできる。このダイジェストデータは、例えば暗号のハッシュ関数SHA1等を利用することが可能であり、メモリのデータ転送と計算処理の値を分けてメモリデバイス11からホスト機器20に転送することが可能である。

40

【0266】

図41は、コマンドヘッダ20aの“OP Code”の一例を示している。

【0267】

“OP Code”は、拡張機能の処理コマンドを1から順番に割り振ることも可能である。また“OP Code”内にコマンドペイロード20bや、レスポンスペイロード20dの有無を示すビットを割り当てることも可能である。

【0268】

50

図4-1において、“OP Code”の6ビット目は、例えば“コマンドペイロードビット”と呼ばれ、コマンドペイロードを必要とするかどうかを示すビットである。“コマンドペイロードビット”が“1”である場合、このコマンドはコマンドペイロードを送ることを示す。

【0269】

また、“コマンドペイロードビット”が“0”である場合、このコマンドはコマンドペイロードを送らず、コマンドヘッダのみ、コマンド処理を実行することを示す。

【0270】

“OP Code”の5ビット目は、“レスポンスペイロードビット”と呼ばれ、レスポンスペイロードを必要とするかどうかを示すビットである。“レスポンスペイロードビット”が“1”である場合、このコマンドはレスポンスペイロードをメモリデバイスが送信することを示す。

10

【0271】

また、“レスポンスペイロードビット”が“0”である場合、メモリデバイスはレスポンスペイロードを送信せず、コマンド処理の実行結果として、レスポンスヘッダのみ送信することを示す。

【0272】

図4-2は、第3の実施形態に適用される拡張レジスタ31のマップの一例を示している。このマップにおいて、オフセットは、拡張レジスタの先頭アドレスからの相対アドレスを示しており、このオフセットは、CMD58、CMD59の“ADDR”フィールドにより指定される。

20

【0273】

オフセット“4”は、“拡張レジスタステータス”と称するレジスタを示している。この“拡張レジスタステータス”は、拡張レジスタがアイドル状態であるか、送信状態であるか、受信状態であるかを示す読み出し専用のレジスタである。

【0274】

オフセット“8”は、“拡張レジスタオペレーション”と称するレジスタを示している。この拡張レジスタオペレーション”は、CMD58、CMD59により使用される書き込み、又は読み出し可能なレジスタである。このレジスタは、リセット可能とされている。

30

【0275】

オフセット“12”は、“拡張コマンドポート”と称するレジスタであり、拡張コマンドの書き込み専用ポートである。

【0276】

オフセット“13”は、“拡張コマンドペイロードポート”と称するレジスタであり、コマンドペイロードの書き込み専用ポートである。

【0277】

オフセット“14”は、“拡張レスポンスペイロードポート”と称するレジスタであり、拡張レスポンスペイロードの読み出し専用ポートである。

40

【0278】

オフセット“15”は、“拡張レスポンスポート”と称するレジスタであり、拡張レスポンスの読み出し専用ポートである。

【0279】

(トランザクションデータ(タイプ1))

図4-3は、図3-7(a)に示すCMD49によりコマンドヘッダ20aをホスト機器20からメモリデバイス11に送信し、CMD48によりメモリデバイス11からのレスポンスヘッダをホスト機器20で受信するタイプ1の場合のトランザクションデータを示している。

【0280】

ホスト機器20よりCMD49が発行されてコマンドヘッダ20aが転送されると、図

50

18に示すC M D 49の“ A d d r ”により、拡張レジスタ内のページと位置が指定され、“拡張コマンドポート”が指定される。この“拡張コマンドポート”を介して、拡張機能部19にO Pコード等が転送される。

【0281】

また、ホスト機器20よりC M D 48が発行されると、図17に示すC M D 49の“ A d d r ”により、拡張レジスタ内のページと位置が指定され、“拡張レスポンスポート”が指定される。この“拡張レスポンスポート”を介して、拡張機能部19からレスポンスヘッダ20cが読み出され、ホスト機器20に転送される。

【0282】

図44は、トランザクションデータ(タイプ1)のタイミングチャートを示している。

10

【0283】

図44に示すように、メモリデバイス11は、C M D 49を受け取ると、レスポンスR1を返し、その後、512バイトのデータブロックとしてのコマンドヘッダ20aを受け取る。

【0284】

メモリデバイス11は、コマンドヘッダ20aが正しく受け取れたかどうかを示すC R Cコードをホスト機器20に返す。その後、コマンドヘッダ20aに記載されたコマンド処理が終わるまでビジーを返し、ホスト機器20が次のコマンドを発行できるタイミングを知らせる。コマンドヘッダ20aは、バッファ16に保持されている。

【0285】

20

コマンド処理において、C M D 49の引数“ A d d r ”により、拡張レジスタ内のページと位置が指定され、データポートとしての“拡張コマンドポート”が指定される。この“拡張コマンドポート”を介してバッファ16に保持された1ブロック(512バイト)のコマンドヘッダが拡張機能部19の割り当てられたデバイスに書き込まれる。

【0286】

この後、メモリデバイス11は、C M D 48を受け取ると、レスポンスR1を返す。

【0287】

コマンド処理において、メモリデバイス11は、C M D 49の引数“ A d d r ”により、拡張レジスタ内のページと位置を指定し、データポートとしての“拡張レスポンスポート”が指定される。この“拡張レスポンスポート”を介して拡張機能部19の割り当てられたデバイスから、512バイトのデータブロックとしてのレスポンスヘッダ20cが読み出される。この読み出されたレスポンスヘッダ20cは、ホスト機器20に転送される。

30

【0288】

(トランザクションデータ(タイプ2))

図45は、図37(b)に示すC M D 49によりコマンドヘッダ20aをホスト機器20からメモリデバイス11に送信した後、C M D 59によりコマンドペイロード20bを送信し、C M D 48によりメモリデバイス11からレスポンスヘッダを読み出し、ホスト機器20で受信するタイプ1の場合のトランザクションデータを示している。

【0289】

40

タイプ1と同様に、ホスト機器20よりC M D 49が発行されてコマンドヘッダ20aが転送されると、C M D 49の“ A d d r ”により、拡張レジスタ内のページと位置が指定され、“拡張コマンドポート”が指定される。この“拡張コマンドポート”を介して、拡張機能部19にO Pコード等が転送される。

【0290】

次いで、ホスト機器20よりC M D 59が発行されてコマンドペイロード20bがメモリデバイス11に転送され、拡張レジスタの“拡張コマンドペイロードポート”を介して、拡張機能部19に複数のペイロードブロックが転送される。

【0291】

この後、ホスト機器20よりC M D 48が発行されると、拡張レジスタの“拡張レスポ

50

ンスポート”を介して、拡張機能部19からレスポンスヘッダ20cが読み出され、ホスト機器20に転送される。

【0292】

図46は、トランザクションデータ(タイプ2)のタイミングチャートを示している。

【0293】

図46に示すように、メモリデバイス11は、CMD49を受け取ると、レスポンスR1を返し、その後、512バイトのコマンドヘッダ20aを受け取る。

【0294】

メモリデバイス11は、コマンドヘッダ20aが正しく受け取れたかどうかを示すCRCコードをホスト機器20に返す。その後、コマンドヘッダ20aに記載されたコマンド処理が終わるまでビジーを返し、ホスト機器20が次のコマンドを発行できるタイミングを知らせる。コマンドヘッダ20aは、バッファ16に保持されている。コマンド処理は、トランザクションデータ(タイプ1)と同様であるため、説明は省略する。

【0295】

次いで、メモリデバイス11は、CMD59を受け取ると、レスポンスR1を返す。

【0296】

コマンド処理において、メモリデバイス11は、CMD59の引数“ADDR”により、拡張レジスタ内のページと位置を指定し、データポートとしての“拡張コマンドペイロードポート”が指定される。この“拡張コマンドペイロードポート”を介して拡張機能部19の割り当てられたデバイスに、コマンドペイロード20bを構成する512バイトのデータブロックとしてのレスポンスヘッダ20cが書き込まれる。

【0297】

具体的には、メモリデバイス11は、先ず、CMD59の“BUS”に指定されたサイズ(512バイト/32kバイト)の1ブロックのデータを受け取る。次いで、メモリデバイス11は、512バイトのデータが正しく受け取れたかどうかを示すCRCコードをホスト機器20に返し、データの受信処理が終わるまでビジーを出力する。

【0298】

この後、CMD59の“BUC”に指定されたブロック数のデータの受信が完了するまで、上記動作が繰り返される。

【0299】

尚、コマンドペイロード20bによりメモリデバイス11に送られたデータは、バッファ16に保持される。このため、メモリデバイス11は、ホスト機器20の制御無しで、バッファ16に保持されたデータを、1ブロック単位で拡張機能部19に転送することが可能であり、所謂DMA(Direct Memory Access)転送が可能となる。

【0300】

この後、メモリデバイス11は、CMD48を受け取ると、レスポンスR1を返し、その後、ホスト機器20は、512バイトのデータブロックとしてのレスポンスヘッダ20cを“拡張レスポンスポート”を介して受け取る。コマンド処理は、トランザクションデータ(タイプ1)と同様であるため、説明は省略する。

【0301】

(トランザクションデータ(タイプ3))

図47は、図37(c)に示すCMD49によりコマンドヘッダ20aをホスト機器20からメモリデバイス11に送信した後、CMD48によりメモリデバイス11からレスポンスヘッダを読み出し、ホスト機器20で受信し、次いで、CMD58によりレスポンスペイロード20bを受信するタイプ3の場合のトランザクションデータを示している。

【0302】

トランザクションデータ(タイプ1)、(タイプ2)と同様に、ホスト機器20よりCMD49が発行されてコマンドヘッダ20aが転送されると、メモリデバイス11は、CMD49の“Addr”により、拡張レジスタ内のページと位置が指定され、“拡張コマンドポート”を指定する。この“拡張コマンドポート”を介して、拡張機能部19にコマ

10

20

30

40

50

ンドヘッダ 20 a が転送される。

【0303】

次いで、ホスト機器 20 より CMD 58 が発行されると、拡張レジスタの“拡張レスポンスペイロードポート”を介して、拡張機能からレスポンスペイロード 20 d が読み出され、ホスト機器 20 に転送される。

【0304】

この後、ホスト機器 20 より CMD 48 が発行されると、拡張レジスタの“拡張レスポンスポート”を介して、拡張機能からレスポンスヘッダ 20 c が読み出され、ホスト機器 20 に転送される。

【0305】

図 48 は、トランザクションデータ（タイプ 3）のタイミングチャートを示している。

【0306】

図 48 に示すように、メモリデバイス 11 は、CMD 49 を受け取ると、レスポンス R1 を返し、その後、512 バイトのコマンドヘッダ 20 a を受け取る。

【0307】

メモリデバイス 11 は、コマンドヘッダ 20 a が正しく受け取れたかどうかを示す CRC コードをホスト機器 20 に返す。その後、コマンドヘッダ 20 a に記載されたコマンド処理が終わるまでビジーを返し、ホスト機器 20 が次のコマンドを発行できるタイミングを知らせる。コマンドヘッダ 20 a は、バッファ 16 に保持されている。コマンド処理は、トランザクションデータ（タイプ 1）と同様であるため、説明は省略する。

【0308】

次いで、メモリデバイス 11 は、CMD 58 を受け取ると、レスポンス R1 を返し、その後、レスポンスペイロード 20 d を構成する 512 バイトのデータを受け取る。

【0309】

コマンド処理において、メモリデバイス 11 は、CMD 58 の“ADDR”に基づき、“拡張レスポンスペイロードポート”を指定し、この拡張レスポンスペイロードポート”を介して、拡張機能部 19 から CMD 58 の“BUS”に指定されたサイズ（512 バイト／32k バイト）の 1 ブロックのデータを受け取る。この受け取ったデータはホスト機器 20 に転送される。次いで、メモリデバイス 11 は、512 バイトのデータが正しく受け取れたかどうかを示す CRC コードをホスト機器 20 に送る。

【0310】

この後、CMD 58 の“BUC”に指定されたブロック数のデータの受信が完了するまで、上記動作が繰り返される。

【0311】

この後、メモリデバイス 11 は、CMD 48 を受け取ると、レスポンス R1 を返し、その後、ホスト機器 20 は、512 バイトのデータブロックとしてのレスポンスヘッダ 20 c を“拡張レスポンスポート”を介して受け取る。コマンド処理は、トランザクションデータ（タイプ 1）と同様であるため、説明は省略する。

【0312】

尚、トランザクションデータ（タイプ 2）において、コマンドヘッダ 20 a は、マンドペイロード 20 b より先に転送したが、コマンドペイロード 20 b を転送した後、コマンドヘッダ 20 を転送してもよい。

【0313】

また、トランザクションデータ（タイプ 3）において、レスポンスヘッダ 20 c は、レスポンスペイロード 20 d の後に転送したが、レスポンスヘッダ 20 c を転送した後、レスポンスペイロード 20 d を転送してもよい。

【0314】

上記第 3 の実施形態によれば、ホスト機器 20 とメモリデバイス 11 の拡張機能部 19 との間のデータの送受信を、CMD 48／49／58／59 と、コマンドヘッダ及びレスポンスヘッダと、コマンドペイロード及びレスポンスペイロードとに分けている。このた

10

20

30

40

50

め、コマンド処理の後、コマンドヘッダ20aに基づき、次のコマンドペイロード20bにより転送されるデータを拡張機能部19への転送するための準備を行うことができる。このため、コマンドペイロードにより送られてきたデータを効率的に拡張機能部19に転送することが可能である。したがって、例えばコマンドペイロードにより送られて来たデータは、DMAにより転送することができる。

【0315】

また、レスポンスをレスポンスヘッダ20cとレスポンスペイロード20dに分けている。このため、コマンド処理の後、レスポンスヘッダ20cに基づき、レスポンスの準備をした後、拡張機能部19から送られたデータをレスポンスペイロード20dとして効率的にホスト機器20に転送することが可能となる。

10

【0316】

また、ホスト機器20は、CMD49により、拡張機能部19の動作を指示するOPコードとアギュメントを含むコマンドヘッダ20aを拡張レジスタの拡張コマンドポートを介して拡張機能部19に転送した後、CMD48により、拡張レジスタの拡張レスポンスポートを介して拡張機能部19から供給されるレスポンスコードとレスポンスデータを含むレスポンスヘッダ20cを受けている。このため、拡張機能部19に必要なコマンドや、メッセージを転送したり、拡張機能部19からのレスポンスをホスト機器20に転送したりすることが可能である。

【0317】

また、長いデータを拡張レジスタに転送する場合、CMD59により複数のブロックにより構成されたコマンドペイロード20bを拡張レジスタの拡張コマンドペイロードポートを介して拡張機能部19に転送することができ、CMD58により複数のブロックにより構成された長いデータを含むレスポンスペイロードを拡張レジスタの拡張レスポンスペイロードポートを介して読み出すことができる。したがって、長いデータを拡張機能部19に対して書き込み、読み出すことが可能である。

20

【0318】

しかも、CMD58／59は、引数としてブロックのサイズを指定する“BUS”と、ブロック数を指定する“BUC”を有している。したがって、ブロック数を指定するCMD23を用いる必要がないため、長いデータを容易に転送することが可能である。

【0319】

(第4の実施形態)

第1乃至第3の実施形態において、メモリデバイス11は、CMD48／49を受けるとレスポンスR1をホスト機器20に返す。

30

【0320】

しかし、メモリデバイス11内部の拡張機能部19は、コマンドに対する処理の完了や、エラーなどのイベントをホスト機器20に通知する手段がなかった。拡張レジスタを介してこれらのイベントを表示することは可能であるが、ホスト機器20はイベントが起こったかどうかをポーリングする必要があり非効率であった。

【0321】

そこで、第4の実施形態は、拡張レジスタをアクセスするコマンドを含む全てのコマンドに対して返信されるレスポンスR1に、拡張レジスタのイベントを通知するフラグを設け、ホスト機器20が拡張レジスタをアクセスすることなく、ホスト機器20にイベントを通知可能とする。

40

【0322】

図49は、レスポンスR1のデータ構造の一例を示している。レスポンスR1は、例えば32ビットにより構成されている。このうち、例えば第18ビットが拡張機能のイベントを示すフラグEF__EVENT(Extension Function Event)に設定されている。

【0323】

メモ리카ード11内の例えば複数の拡張機能のうちの少なくとも1つにイベントが発生すると、フラグEF__EVENTが“1”に設定される。このフラグEF__EVENTの

50

ビットは、以降のレスポンスにおいて、“0”に戻してもよいし、次に説明するフラグが読まれたとき、“0”に戻してもよい。

【0324】

図50は、例えば拡張レジスタ内に設けられた拡張機能イベントフラグ(Extension Function Event Flag)の一例を示している。

【0325】

この拡張機能イベントフラグは、例えば17ビットにより構成されており、各ビットに対応して拡張機能の番号が対応されている。

【0326】

拡張機能において、イベントが発生した場合、該当する拡張機能の番号のビットが“1”に設定される。イベントの種別は各拡張機能の拡張レジスタ内で通知できるように定義しても良い。このビットは、このビットの値が読み出された時点で“0”に戻っても良いし、各拡張機能に“0”に戻るタイミングを定義しても良い。

【0327】

図51は、ホスト機器20にアプリケーションと、拡張機能アプリケーション1、拡張機能アプリケーション2、及びこれらに対応したファイルシステム、拡張機能ドライバ1、及び拡張機能ドライバ2が設けられている場合の例を示している。このような構成のホスト機器20により、メモリデバイス11をアクセスする場合の動作について説明する。

【0328】

図52は、上記構成のホスト機器20により、メモリデバイス11をアクセスする場合の動作のシーケンスチャートの例を示している。

【0329】

先ず、ホスト機器20のアプリケーション、又はファイルシステムから書き込み、又は読み出し要求が発生すると(S81)、カードドライバ、又はホストコントローラからCMDXXが発生される(S82)。このCMDXXは、CMD48/49/58/59のいずれかである。

【0330】

メモリデバイス11は、CMDXXを受けるとレスポンスR1をホスト機器20に返す(S83)。

【0331】

このレスポンスR1に含まれるフラグEF_EVENTが“1”に設定されている場合、カードドライバ、又はホストコントローラは、CMD48を発行し(S85)、拡張レジスタの拡張機能イベントフラグを読み出す(S86)。

【0332】

カードドライバ、又はホストコントローラは、拡張機能イベントフラグを解析し、例えば拡張機能ドライバ1にイベントが発生したことを検出すると、割り込み、又はカードドライバのポーリングにより、拡張機能ドライバ1に通知する(S87)。

【0333】

この後、機能拡張ドライバ1は、カードドライバ、又はホストコントローラに対してイベントの内容を読み出すためCMD48を発行し(S88)、カードドライバ、又はホストコントローラは、メモリデバイス11にCMD48を発行する(S89)。

【0334】

メモリデバイス11は、CMD48を受けると、レスポンスR1をホスト機器20に返す(S90)。

【0335】

この後、コマンド処理により、拡張機能部からイベントの内容がホスト機器20に転送される(S91)。このイベントの内容は、さらに機能拡張ドライバ1に転送される(S92)。

【0336】

上記第4の実施形態によれば、CMD48/49/58/59のレスポンスR1に拡張

10

20

30

40

50

機能のいずれかにイベントが発生したことを示すフラグE F _ E v e n t を設けている。このため、ホスト機器20は、レスポンスR1を受けることにより、拡張機能のいずれかにイベントが発生したことを知ることができる。したがって、ポーリングにより、イベントの発生を検出する必要がないため、効率良くイベントを検出することが可能である。

【0337】

また、拡張レジスタ内に拡張機能イベントフラグを設け、この拡張機能イベントフラグにより、拡張機能部19の何れの機能によりイベントが発生したこと特定可能としている。このため、ホスト機器20は、フラグE F _ E v e n t によりイベントの発生が検出された場合、拡張機能イベントフラグを読み出すことにより、イベントが発生した拡張機能を容易に特定することができる。

10

【0338】

(第5の実施形態)

第1乃至第4の実施形態は、ホスト機器20によりメモリデバイス11の拡張機能をアクセスする場合について説明した。

【0339】

これに対して、第5の実施形態は、ホスト機器20とメモリデバイス11との間に変換機器を設け、この変換機器を介してメモリデバイス11の拡張機能をアクセスする場合について説明する。具体的には、例えばパーソナルコンピュータとしてのホスト機器20にメモリデバイス11を接続するためのアダプタが接続される場合がある。第5の実施形態は、このようなアダプタが接続された場合においても、ホスト機器20とメモリデバイス11の拡張機能部19とのデータ転送を可能とするものである。

20

【0340】

図53は、ホスト機器20とメモリデバイス11との間に、例えばUniversal Serial Bus (USB) タイプの接続機器としてのアダプタ100が接続されている場合の例を概略的に示している。

【0341】

アダプタ100は、変換器101を有している。変換器101は、コマンド分割部102、レスポンス結合部103、及びコマンド発行部104を有している。

【0342】

図54、図55は、図53で示す例の処理動作を示している。図54は、データの書き込み動作の例を示し、図55は、データの読み出し動作の例を示している。

30

【0343】

USBタイプのアダプタ100を用いてメモリデバイス11がホスト機器20に接続される場合、データをホスト機器20からアダプタ100に対して、先ずCommand Block Wrapper (CBW) と呼ばれるメッセージを送り、データの送信もしくは受信を行う。最後にCommand Status Wrapper (CSW) と呼ばれるメッセージをアダプタ100からホスト機器20へ送信してデータの送受信を制御する。CBWにおいて、データの送受信の方向、送受信のデータの長さを指定できる。

【0344】

ホスト機器20は、メモリデバイス11の拡張機能部19に長いデータを転送する場合、前述したコマンドヘッダ20a、及びコマンドペイロード20bを使用し、拡張機能部19からホスト機器20に長いデータを転送する場合、レスポンスヘッダ20c、及びレスポンスペイロード20dが使用される。

40

【0345】

図54に示すように、データの書き込み時、ホスト機器20はCBWを変換器101へ送る(S101)。変換器101は、CSWを解析し、この後にデータが書き込まれることを、送られてくるデータの長さと共に知ることができる(S102)。この時、データの長さに基づきコマンドペイロードが含まれるかどうかを判断できる。

【0346】

変換器101は、ホスト機器20から転送されたコマンドヘッダ20a、コマンドペイ

50

ロード20bを受けると(S103)、コマンド分割部102により、コマンドヘッダ20a、コマンドペイロード20bを分離する。すなわち、コマンド分割部102は、転送されたデータのうち、最初の512バイトをコマンドヘッダ20aとみなし、これ以降のデータをコマンドペイロード20bとみなして、これらを分割する。長さが512バイトを超えない場合、変換器101はコマンドペイロードが無いものとして処理する。

【0347】

コマンド発行部104は、コマンド分割部102から供給されたコマンドヘッダ20aに対してCMD49を発行し(S104)、メモリデバイス11からレスポンスR1を受けた後(S105)、CMD49によりコマンドヘッダ20aをメモリデバイス11に転送する(S106)。メモリデバイス11の内部動作(S107)は、上述した通りである。

10

【0348】

また、コマンド発行部104は、コマンド分割部102から供給されたコマンドペイロード20bに対してCMD59を発行し(S108)、メモリデバイス11からレスポンスR1を受けた後(S109)、コマンドペイロード20bをメモリデバイス11に転送する(S110)。メモリデバイス11は、コマンドペイロードを処理する(S111)。

【0349】

CMD59の発行時、コマンドペイロードの長さと同じかそれ以上の長さになるように“BUS”、“BUC”を設定する。コマンドペイロードの長さ以上に“BUS”、“BUC”を設定した場合、コマンドペイロードを送信した後、“BUS”、“BUC”で設定した長さの任意のデータを続けて送信しても良いし、送信を中止するコマンドを発行して次の作業に進んでも良い。

20

【0350】

変換器101は、データをメモリデバイス11に送信した後、CSWをホスト機器20に送信する(S112)。これによりホスト機器20は、変換器101の処理が終了したことを知ることができる。

【0351】

一方、図55に示すように、メモリデバイス11から長いデータを読み出す場合、ホスト機器20から変換機101へCBWが送られる(S121)。変換器101は、CBWを解析し、この後にデータ読み出しが行われることを、読み出すデータの長さと共に知ることができる(S122)。この時、データの長さからレスポンスペイロードが必要かどうか判断できる。もし指定されたデータの長さが512バイトを超えていた場合、レスポンスペイロードの読み出しが必要であると判断する。

30

【0352】

変換器101のコマンド発行部104は、レスポンスペイロードの読み出しが必要な場合、CMD58を発行する(S123)。CMD58発行時、レスポンスペイロードの長さと同じかそれ以上の長さになるように“BUS”、“BUC”を設定する。レスポンスペイロードの長さ以上に“BUS”、“BUC”を設定した場合、レスポンスペイロードを受け終わった後、“BUS”、“BUC”で設定した長さになる任意のデータを続けて読み出ししても良いし、送信を中止するコマンドを発行して次の作業に進んでも良い。

40

【0353】

メモリデバイス11からレスポンスR1を受けた後(S124)、コマンド発行部104は、メモリデバイス11から受け取ったブロックをレスポンス結合部103に供給する(S125)。

【0354】

この後、コマンド発行部104は、CMD48を発行する(S126)。コマンド発行部104は、メモリデバイス11からレスポンスR1を受けた後(S127)、メモリデバイス11から供給された512バイトのデータをレスポンスヘッダ20cとしてレスポンス結合部103に供給する(S128)。

50

【0355】

レスポンス結合部103は、レスポンスヘッダ20cと、複数のブロックにより構成されたレスポンスペイロード20dを結合してホスト機器20に転送する(S129)。

【0356】

最後に変換器101はCSWをホスト機器20に送信する(S130)。ホスト機器は、これにより変換器101が処理を終了したことを知ることができる。

【0357】

コマンド分割部102、及びレスポンス結合部103は、処理を行うのに十分なサイズの作業領域を持っていない場合、コマンド発行部104、及びホスト機器20へそれまで作業領域に蓄積されたデータをその都度それぞれに送り出して逐次処理することも可能である。

10

【0358】

尚、上記説明は、長いデータを転送する場合であったが、コマンドペイロード20b、及びレスポンスペイロード20dを使用しないデータ転送の場合、変換器101のコマンド分割部102は、単にコマンドヘッダ20aを出力し、レスポンス結合部103は、単にレスポンスヘッダ20cを出力する。

【0359】

また、メモリデバイス11は、必要なデータを送った後にタイムアウトする(例えば250msec)。この後、レスポンスペイロードが送り終わったことをレスポンスR1のイベントを用いて示しても良い。

20

【0360】

図56は、第5の実施形態の変形例を示すものであり、アダプタの他の例を示している。

【0361】

図56において、アダプタ110は、ホスト機器20に対してメモリデバイス11と例えばハードディスク113を接続可能としている。

【0362】

アダプタ110は、分配器111と、変換器101、及びブリッジ回路112を有している。分配器111の第1の端子は、USBを介してホスト機器20に接続されている。分配器111の第2の端子は、変換器101を介してメモリデバイス11に接続されている。変換器101の構成は、上述した通りである。分配器111の第3の端子は、ブリッジ回路112、SATA(serial ATA)ケーブル114を介してハードディスク113に接続されている。ブリッジ回路112は、例えば図示せぬメモリインターフェースモジュールと、メモリ制御モジュールなどを含んでいる。

30

【0363】

このような構成の場合においても、変換器101を用いることにより、ホスト機器20とメモリデバイス11の拡張機能部19との間でデータを転送することが可能である。

【0364】

上記第5の実施形態によれば、ホスト機器20とメモリデバイス11との間にアダプタ100が設けられている場合においても、アダプタ100に変換器101を設けることにより、ホスト機器20とメモリデバイス11の拡張機能部19との間でデータを転送することが可能である。

40

【0365】

尚、上記実施形態において、拡張機能の一例としては、コントローラに実装されるセキュリティ機能や、メモリ、例えばNANDフラッシュメモリに実装されるセキュリティ機能がある。

【0366】

コントローラに実装されるセキュリティ機能とは、例えば鍵情報及びコントローラ固有の識別情報等をコントローラに格納し、これらの情報を基にホスト機器との間でセキュアな通信チャンネルを構成するための機能等である。

50

【0367】

また、メモリに実装されるセキュリティ機能とは、例えば鍵情報、メモリ固有の識別情報、及びこの識別情報を暗号化して生成された暗号化識別情報等をメモリに格納し、これらの情報を基にコントローラを介してホスト機器との間で認証処理を実施するための機能等である。

【0368】

その他、本発明は上記各実施形態そのままに限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で構成要素を変形して具体化できる。また、上記各実施形態に開示されている複数の構成要素の適宜な組み合わせにより、種々の発明を形成できる。例えば、実施形態に示される全構成要素から幾つかの構成要素を削除してもよい。さらに、異なる実施形態にわたる構成要素を適宜組み合わせてもよい。

10

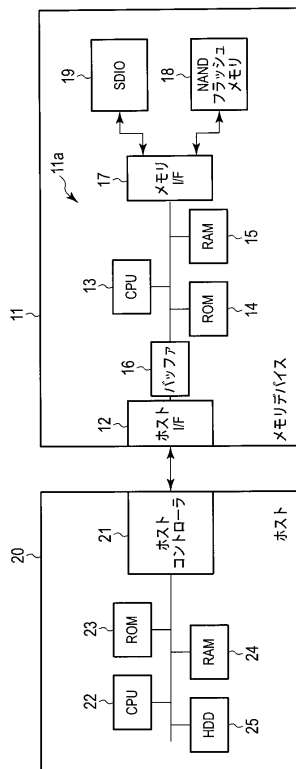
【符号の説明】

【0369】

11…メモリデバイス、11a…カードコントローラ、18…NANDフラッシュメモリ、19…拡張機能部、20…ホスト機器、31…拡張レジスタ、101…変換器、102…コマンド分割部、103…レスポンス結合部。

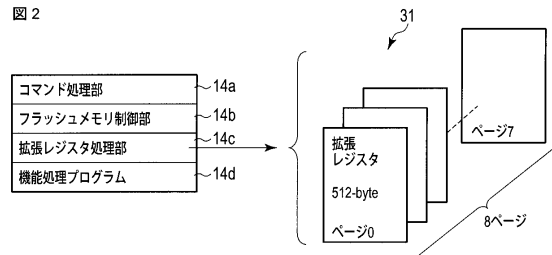
【図1】

図1



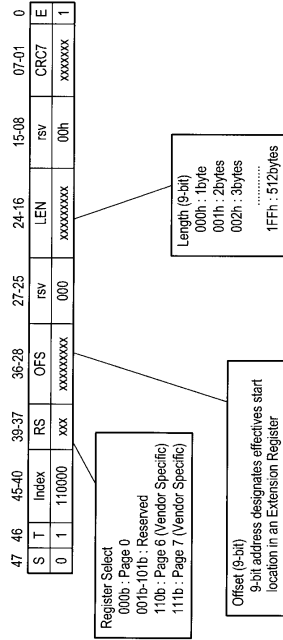
【図2】

図2



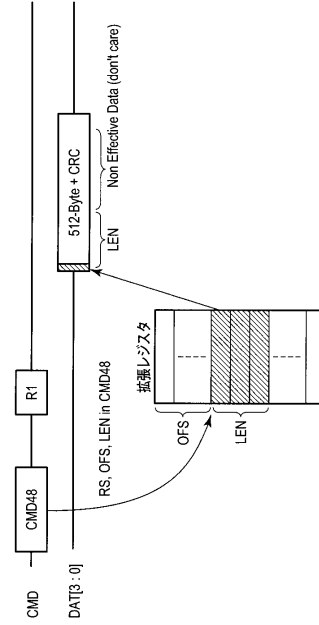
【図 3】

図 3



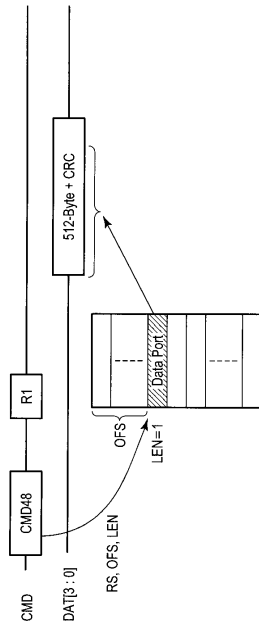
【図 4】

図 4



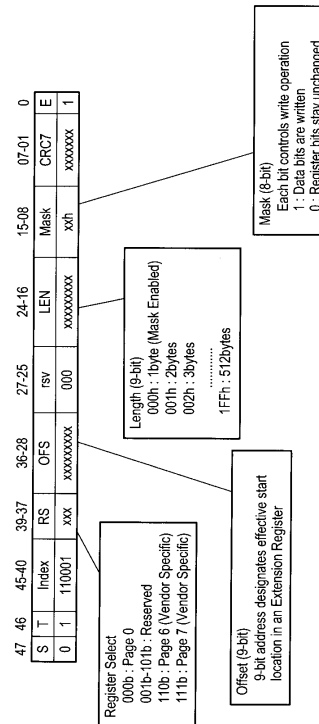
【図 5】

図 5



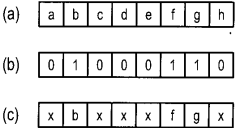
【図 6】

図 6



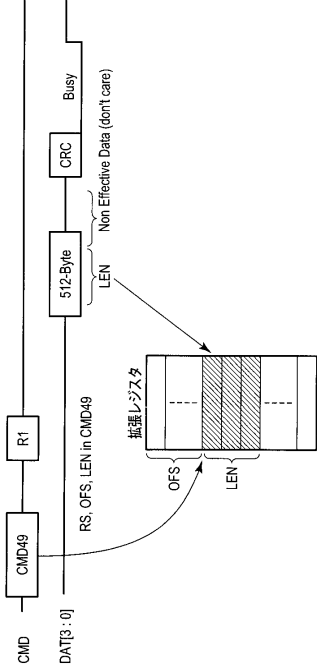
【図 7】

図 7



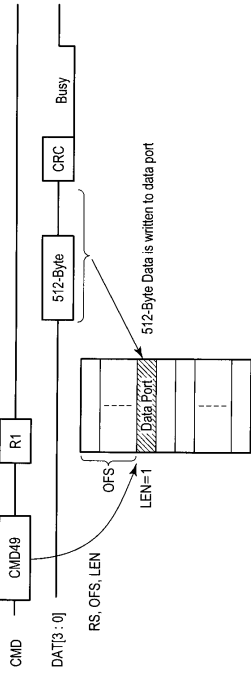
【図 8】

図 8



【図 9】

図 9

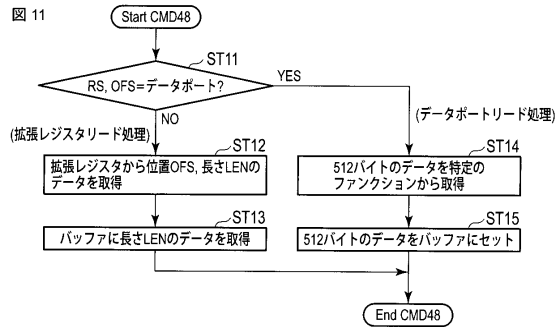


【図 10】

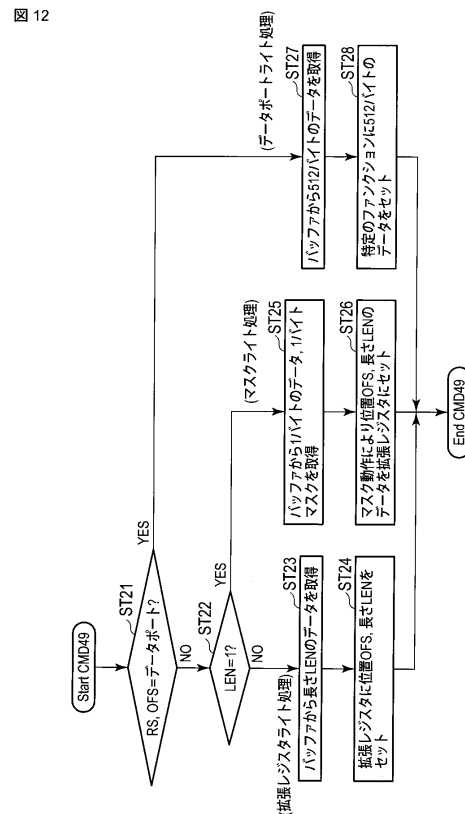
図 10

ストラクチャレビジョン	2/バイト	Page 07フォーマットを定義するレビジョン
データ長	2/バイト	Page 01に記録されている有効なデータ長
拡張機能数 (=N)	1/バイト	デバイスがサポートしている拡張機能の数
予備	1/バイト	（フィールドが埋め込まれたままにしておく）
デバイス1機能識別コード	4/バイト	標準デバイス識別コードの拡張機能は標準化されたデバイスドライバを用いる。非標準の機能は0を設定する
デバイス1製造者識別情報	16/バイト	文字列で製造者、または販売者を記録
デバイス1機能識別情報	16/バイト	拡張機能は専用のドライバがインストールされているかどうかを確認する情報を提供する。文字列で型番、レビジョンなどを記録
次デバイスの先頭アドレス (Page0)	2/バイト	デバイス1を使用しないとき、デバイス2のチェックに移行する
デバイス1アドレス・レンジ数 (=X)	2/バイト	下記に設定されているアドレス・レンジフィールドの数
デバイス1 開始アドレス1、レンジ1	各2/バイト	デバイスが使用する拡張レジスタの第1領域。Page 1~7の拡張レジスタ空間内の先頭アドレスと、使用する拡張レジスタ領域の大きさを示す
デバイス1 開始アドレス2、レンジ2	各2/バイト	デバイス1が使用する拡張レジスタの第2領域
...	...	...
デバイス1 開始アドレスX、レンジX	各2/バイト	デバイス1が使用する拡張レジスタの第X領域
デバイス2機能識別コード	4/バイト	デバイス2の機能識別コード。内容はデバイス1と同様
...	...	...
デバイス2 開始アドレスY、レンジY	4/バイト	デバイス2のアドレス・レンジ情報終了を示す
...	...	...
デバイスN機能識別コード	4/バイト	デバイスNの機能識別コード。内容はデバイス1と同様
...	...	...
デバイスN 開始アドレスZ、レンジZ	4/バイト	デバイスNのアドレス・レンジ情報終了を示す未使用領域

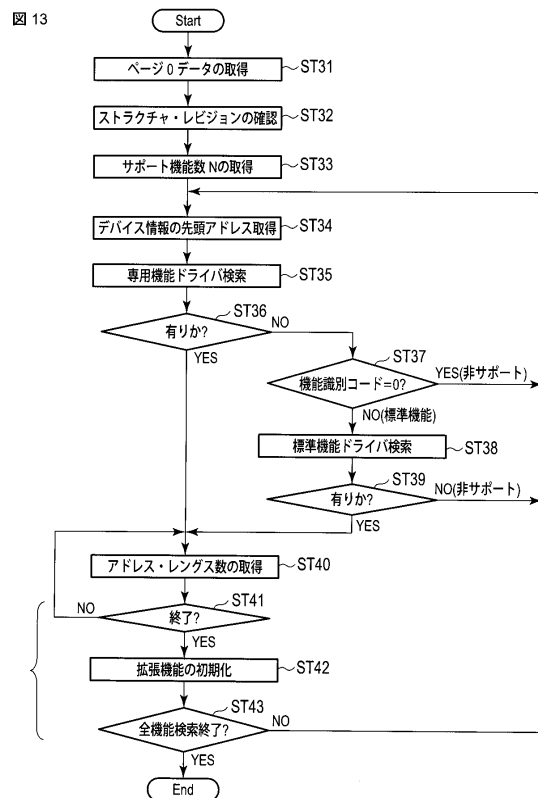
【図 1 1】



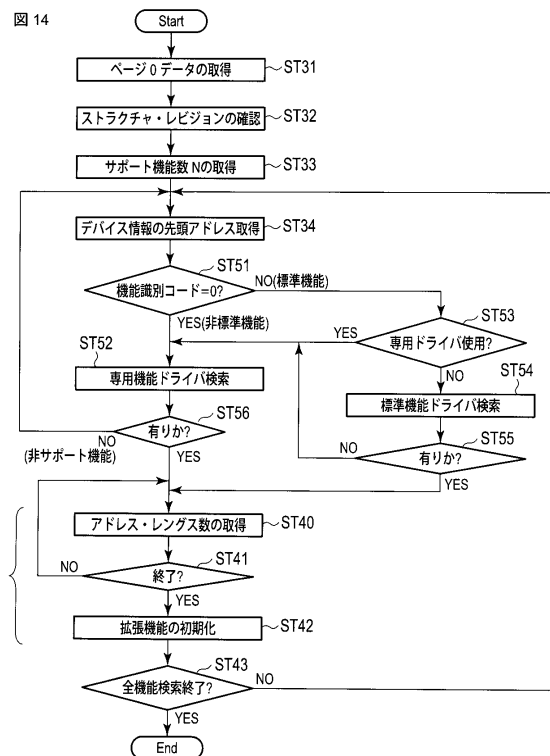
【図 1 2】



【図 1 3】

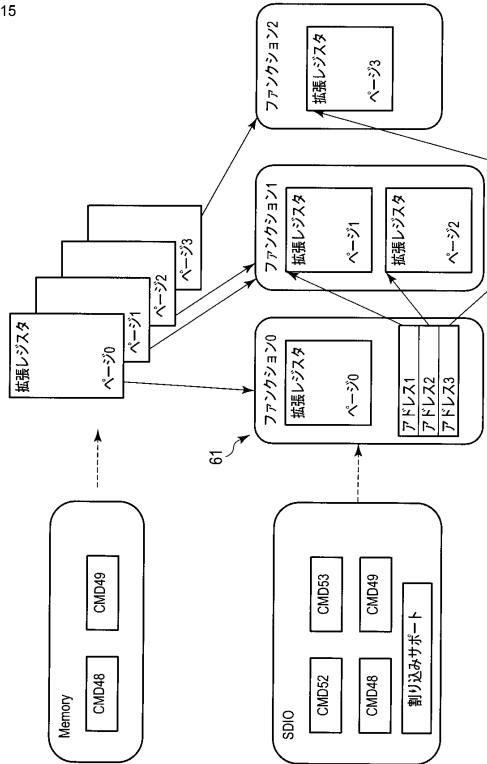


【図 1 4】



【図 15】

図 15



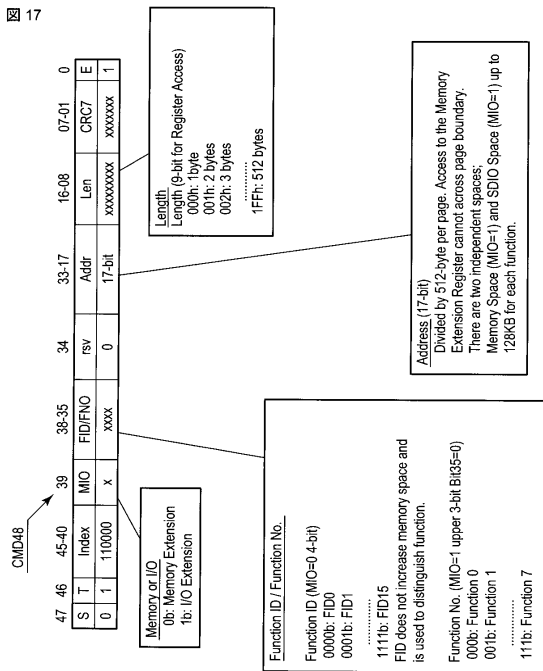
【図 16】

図 16

機能ドライバレビジョン	カードの機能レビジョン		
	A	B	C
A	レビジョン A	レビジョン A	レビジョン A
B	レビジョン A	レビジョン B	レビジョン B
C	レビジョン A	レビジョン B	レビジョン C

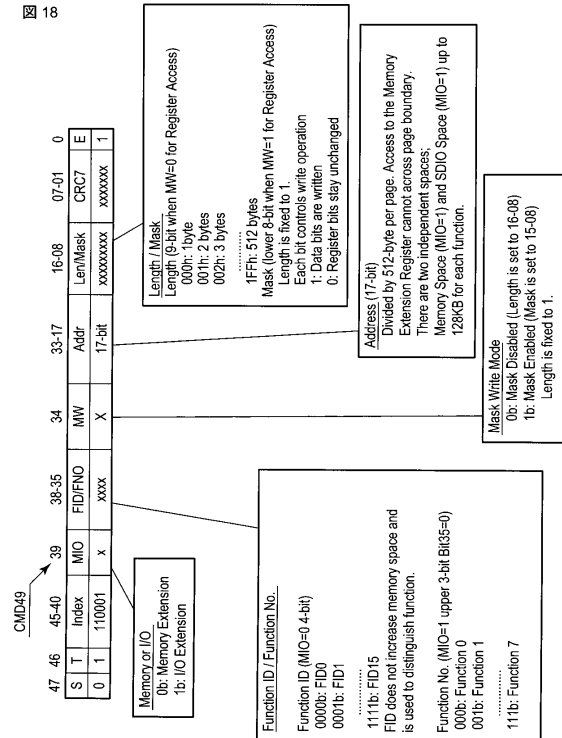
【図 17】

図 17



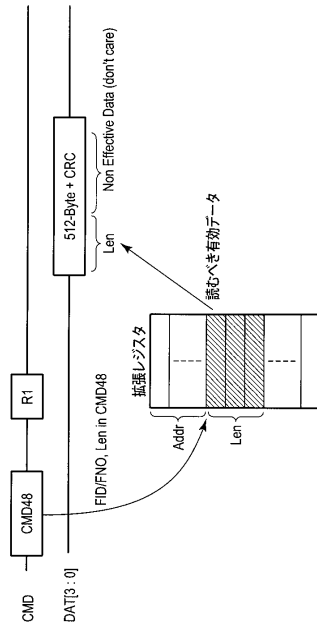
【図 18】

図 18



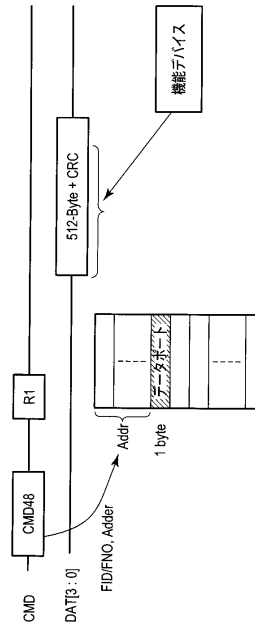
【図 19】

図 19



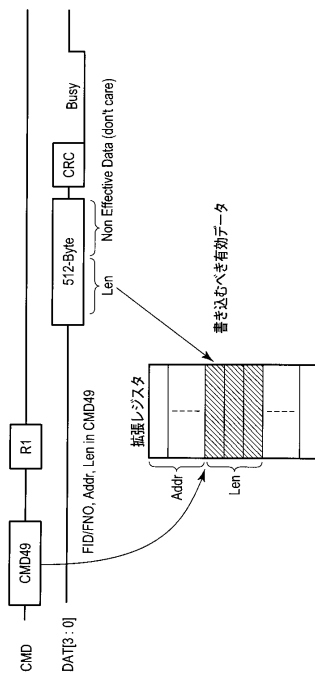
【図 20】

図 20



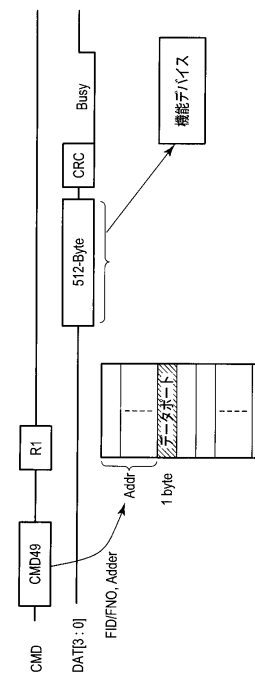
【図 21】

図 21



【図 22】

図 22



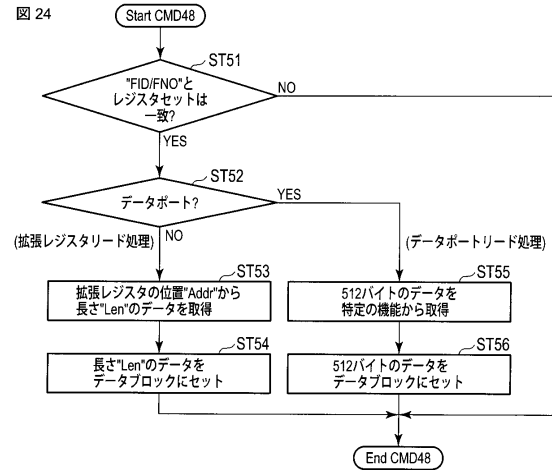
【図 23】

図 23

ストラチャレビジョン	21-bit	ページのフォーマット定義
ページの有効長	21-bit	ページに記録された有効データ長
拡張された機能の数	1-bit	デハイスはサポートされた拡張機能の数
予備	4-bit	00hにセット(偶数バイトのための追加)
拡張1機能識別コード	16-bit	特定機能ドライバ(を使用するための機能識別コード)非標準機能はこのフィールドを0にセットする
拡張1製造者識別情報	16-bit	ASCIIにより製造者名や販売者名を記載(SDAIにより管理されない)
拡張1機能識別情報	16-bit	ASCIIにより機能情報を記載(SDAIにより管理されない)
次の拡張へのポイント	2-bit	FNO=000b, アドレスビット17は常に0として扱われる
レジスタセットの数(=x)	1-bit	下記のアドレス/長さ/長さ1
拡張1アドレス1, 長さ1	4-bit	00hにセット(偶数バイトのための調整)
拡張1アドレス2, 長さ2	4-bit	拡張1の第2レジスタセット
拡張1アドレスx, 長さx	4-bit	拡張1の第xレジスタセット
拡張2機能識別コード	4-bit	拡張2機能のスタート
拡張2アドレス1, 長さ1	4-bit	拡張2の第1レジスタセット
拡張2機能識別コード	4-bit	拡張2機能のスタート
拡張Nアドレス1, 長さ1	4-bit	拡張Nの第1レジスタセット
拡張Nアドレス2, 長さ2	4-bit	拡張Nの第2レジスタセット
拡張Nアドレスx, 長さx	4-bit	拡張Nの第xレジスタセット
未使用領域	16-bit	オール0にセット

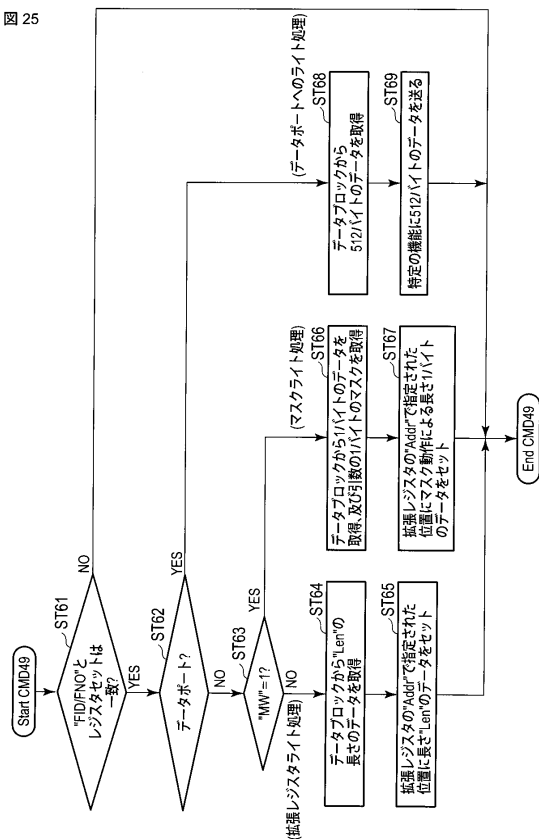
【図 24】

図 24



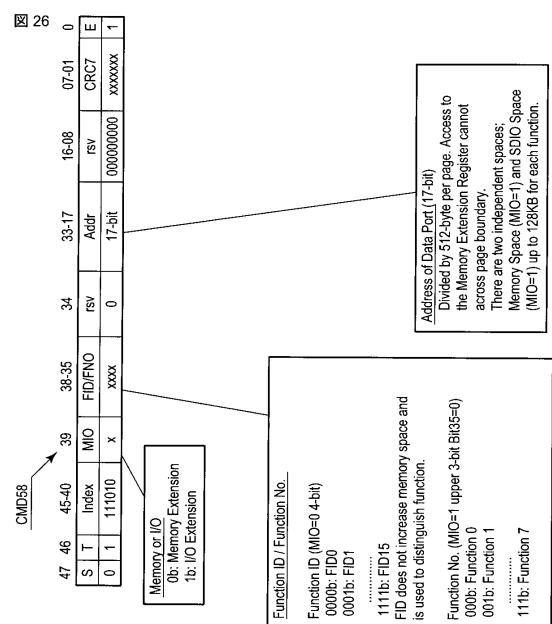
【図 25】

図 25

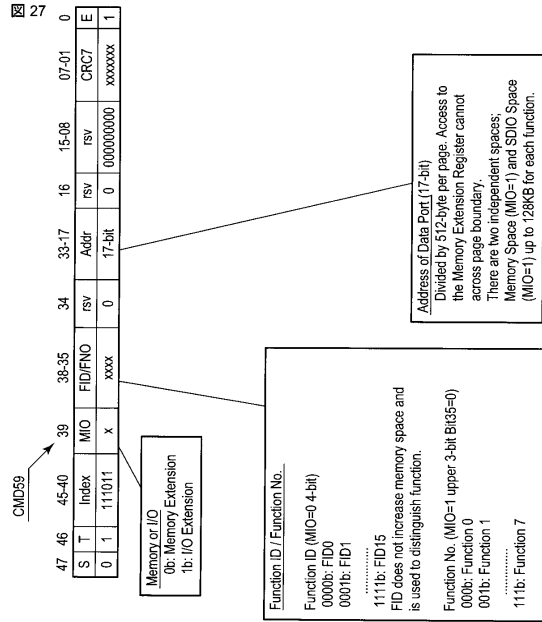


【図 26】

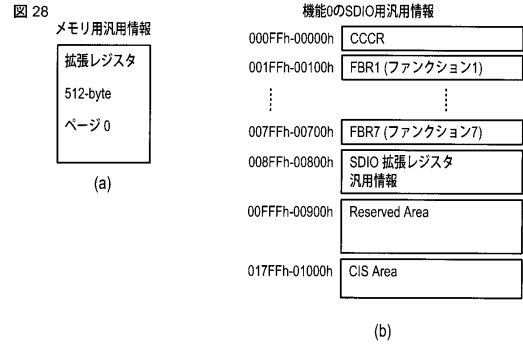
図 26



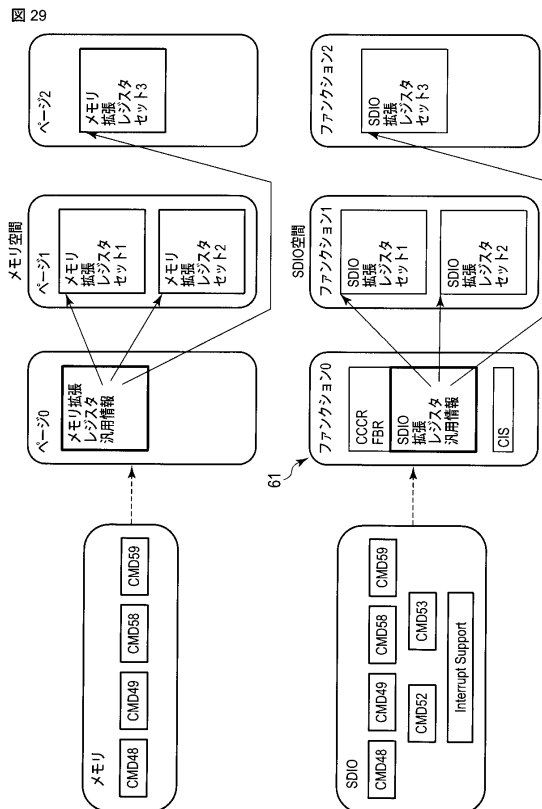
【図 27】



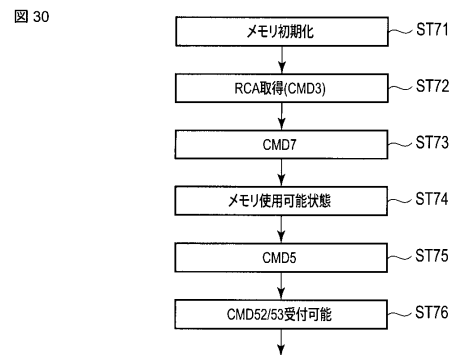
【図 28】



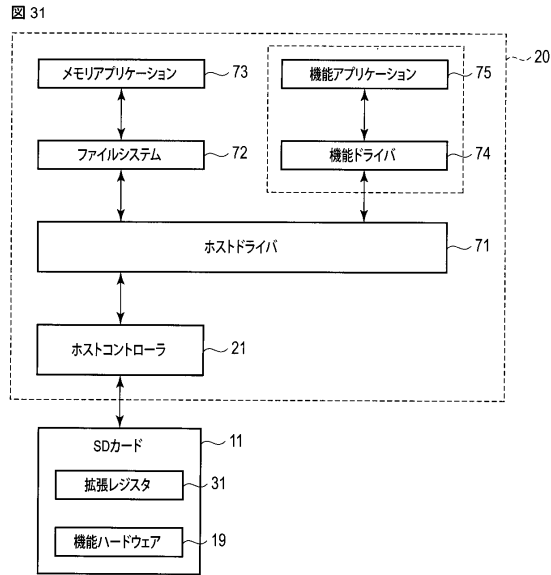
【図 29】



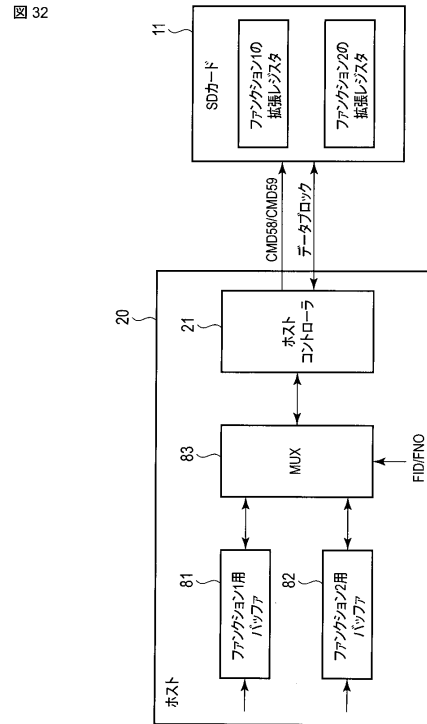
【図 30】



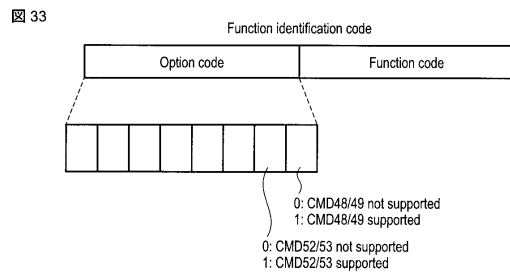
【図 3 1】



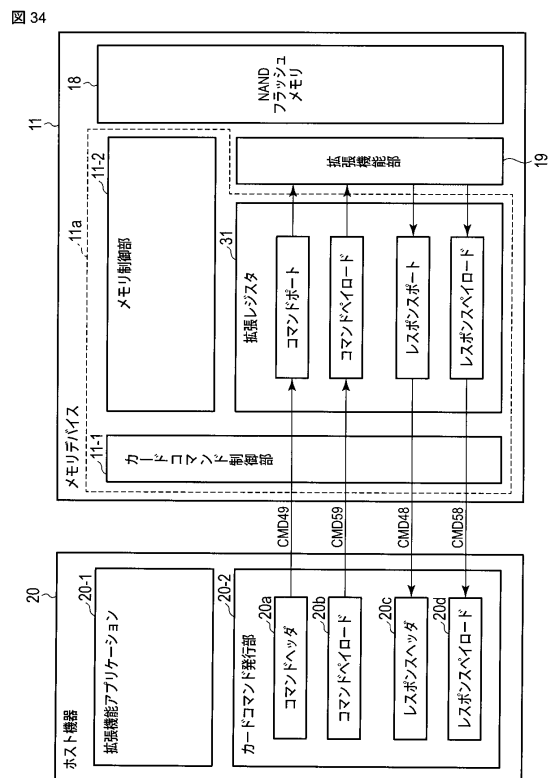
【図 3 2】



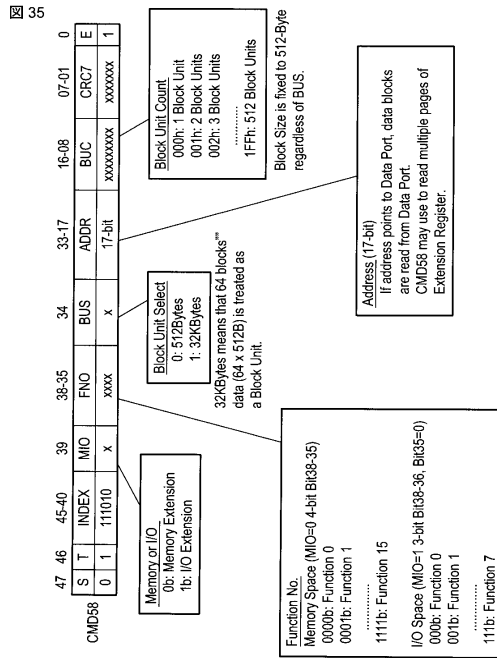
【図 3 3】



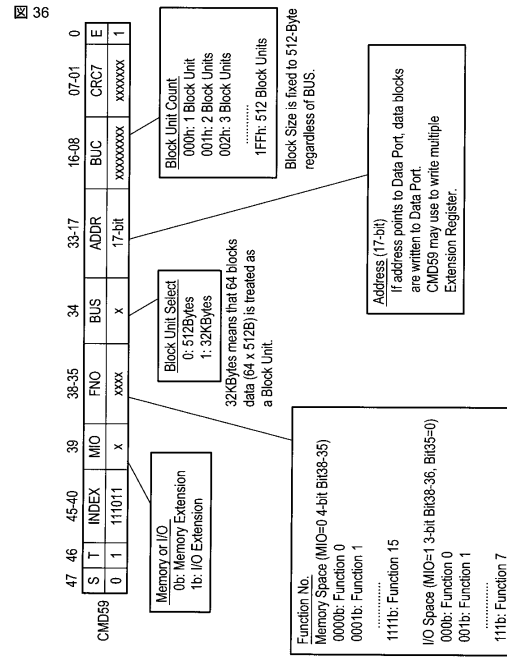
【図 3 4】



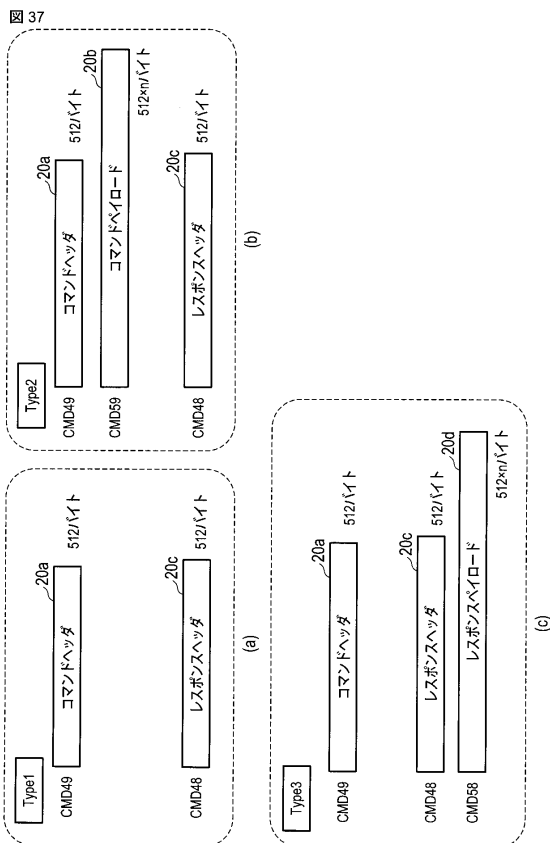
【図 35】



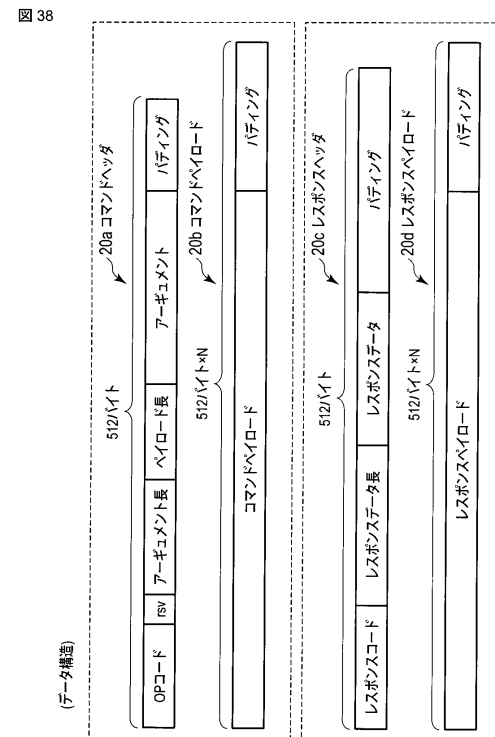
【図 36】



【図 37】



【図 38】



【図 39】

図 39

Byte Position	Name	Description
0	OP Code	80h
1	Reserved	00h
2-3	Argument Length	04h
4-7	Payload Length	N
8	Block Number	n
9-11	Reserved	000000h
Xxx-511	Padding data	Padding data

20a コマンドヘッダ

Byte Position	Name	Description
0-511	Encrypted Key Block #1	暗号化された鍵ブロック
:	:	:
-(512×n-1)	Encrypted Key Block #n	

20b コマンドペイロード

Byte Position	Name	Description
0	Response Code	00h
1-511	Padding data	Padding data

20c レスポンスヘッダ

【図 40】

図 40

Byte Position	Name	Description
0	OP Code	81h
1	Reserved	00h
2-3	Argument Length	0Ch
4-7	Payload Length	N
8	Block Number	n
9-11	Reserved	000000h
12-15	EKB Offset	EKBの読出開始場所
16-19	EKB Length	読みだすEKBの長さ
20-511	Padding data	Padding data

20a コマンドヘッダ

Byte Position	Name	Description
0-511	Encrypted Key Block #1	暗号化された鍵ブロック
:	:	:
-(512×n-1)	Encrypted Key Block #n	

20b コマンドペイロード

Byte Position	Name	Description
0	Response Code	00h
1-511	Padding data	Padding data

20c レスポンスヘッダ

【図 41】

図 41

7	6	5	4	3	2	1	0
	コマンドペイロードビット	レスポンスペイロードビット	コマンドコード				

OP Code

【図 42】

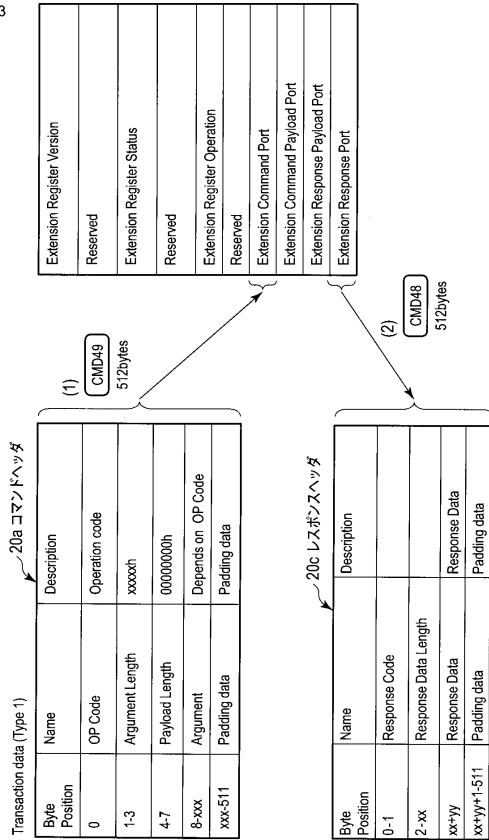
図 42

Offset	Register Name	Type	Description
0	Extension Register Version	Read only register	Version number
1-3	Reserved	-	
4	Extension Register Status	Read only register	Register Status. Ex. Idle, Send, Receive
5-7	Reserved	-	
8	Extension Register Operation	Read and write register	Register can be reset
9-11	Reserved	-	
12	Extension Command Port	Write only port	Extension Command
13	Extension Command Payload Port	Write only port	Command payload
14	Extension Response Payload Port	Read only port	Long response payload
15	Extension Response Port	Read only port	Extension response

拡張レジスタマップの例

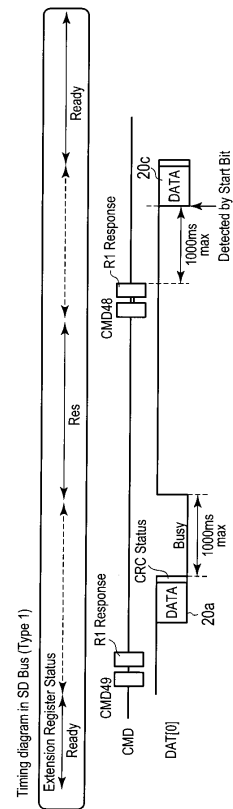
【図 4 3】

図 43



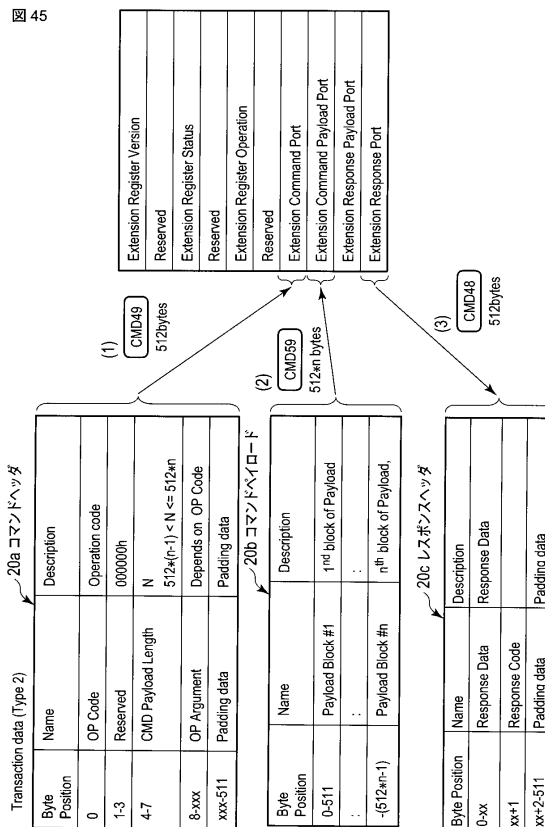
【図 4 4】

図 44



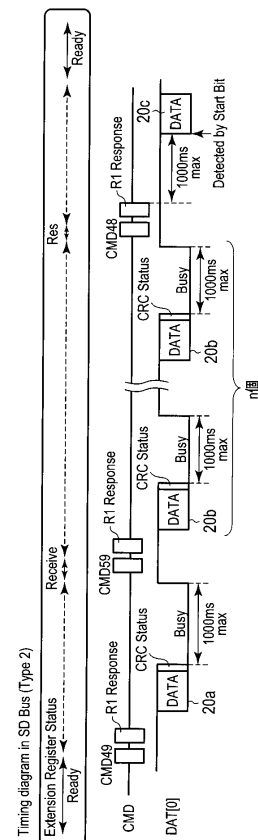
【図 4 5】

図 45

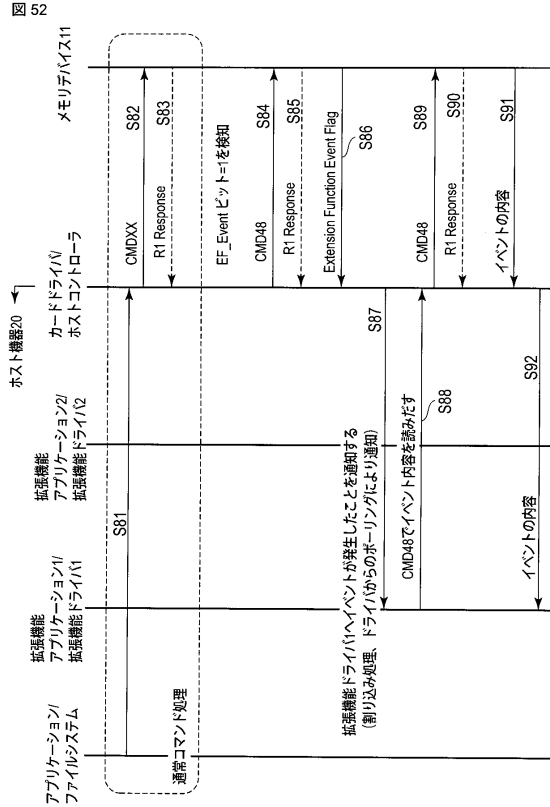


【図 4 6】

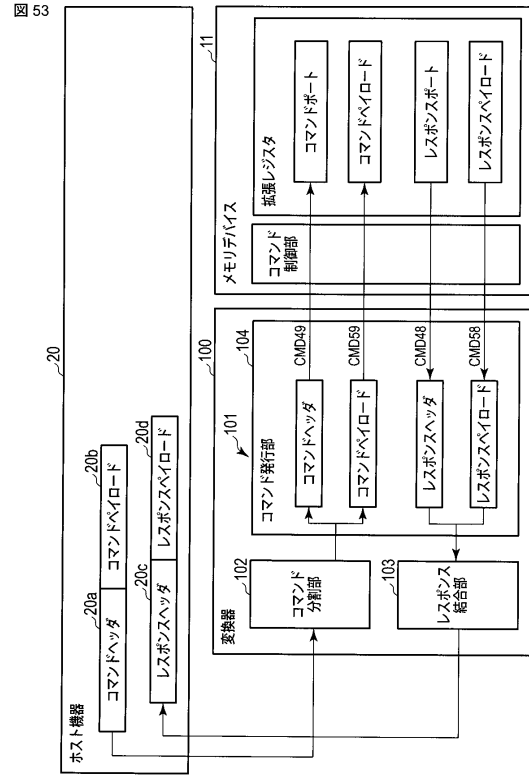
図 46



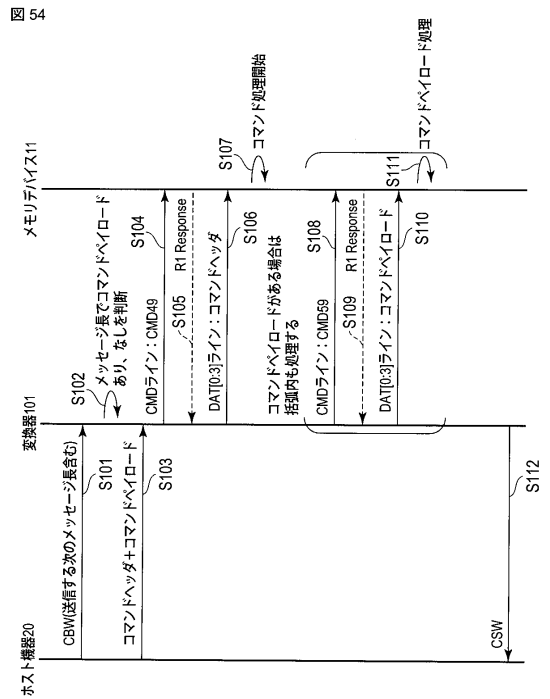
【図 5 2】



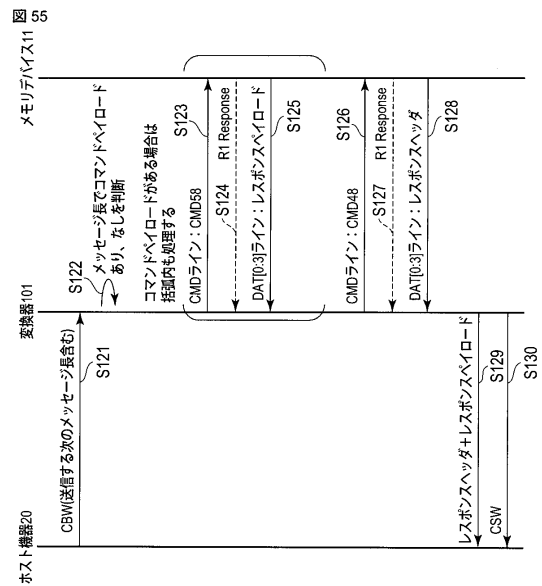
【図 5 3】



【図 5 4】

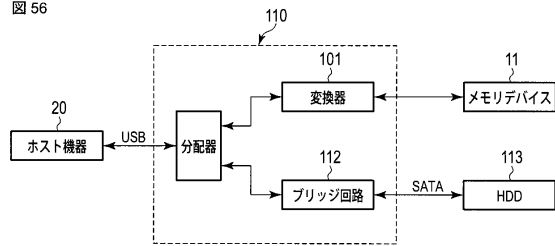


【図 5 5】



【図 56】

図 56



フロントページの続き

- (74)代理人 100153051
弁理士 河野 直樹
- (74)代理人 100140176
弁理士 砂川 克
- (74)代理人 100158805
弁理士 井関 守三
- (74)代理人 100172580
弁理士 赤穂 隆雄
- (74)代理人 100179062
弁理士 井上 正
- (74)代理人 100124394
弁理士 佐藤 立志
- (74)代理人 100112807
弁理士 岡田 貴志
- (74)代理人 100111073
弁理士 堀内 美保子
- (72)発明者 松川 伸一
東京都港区芝浦一丁目1番1号 株式会社東芝内
- (72)発明者 藤本 曜久
東京都港区芝浦一丁目1番1号 株式会社東芝内

審査官 岡北 有平

- (56)参考文献 特開2002-171303 (JP, A)
特開2005-050320 (JP, A)
特開2001-282467 (JP, A)
特開2006-031235 (JP, A)
特開2006-236200 (JP, A)
特開2003-319322 (JP, A)
特開2011-216068 (JP, A)
特開2001-356885 (JP, A)
特開2008-210301 (JP, A)
米国特許出願公開第2010/0069039 (US, A1)
特開平09-035019 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
G06K 19/00 - 19/18