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

特許第5065618号
(P5065618)

(45)発行日　平成24年11月7日(2012.11.7)　(24)登録日　平成24年8月17日(2012.8.17)

(51)Int.Cl.　　F I
G06F 12/06 (2006.01)　　G06F 12/06 510A
G06F 13/16 (2006.01)　　G06F 12/06 515F
　　G06F 13/16 510B

請求項の数 13　(全 61 頁)

(21)出願番号　特願2006-135970 (P2006-135970)
(22)出願日　平成18年5月16日 (2006.5.16)
(65)公開番号　特開2007-310430 (P2007-310430A)
(43)公開日　平成19年11月29日 (2007.11.29)
審査請求日　平成21年3月4日 (2009.3.4)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(73)特許権者　500174247
エルピーダメモリ株式会社
東京都中央区八重洲２－２－１
(74)代理人　100080001
弁理士　筒井　大和
(72)発明者　三浦　誉士
東京都国分寺市東恋ヶ窪一丁目２８０番地
株式会社日立製作所　中央研究所内
(72)発明者　藪　彰
東京都中央区八重洲二丁目２番１号　エルピーダメモリ株式会社内

最終頁に続く

(54)【発明の名称】メモリモジュール

(57)【特許請求の範囲】
【請求項１】
第１のメモリデバイスと、
前記第１のメモリデバイスに接続される第２のメモリデバイスと、を有し、
前記第１のメモリデバイスは、
情報を記憶する第１のメモリ回路と、
第１のリクエスト信号を情報処理装置から前記第１のメモリ回路に転送し、第２のリクエスト信号を前記情報処理装置から前記第２のメモリデバイスに転送する第１のリクエストキュー制御回路と、
第１のレスポンス信号を前記情報処理装置に出力し、第２のレスポンス信号を前記第２のメモリデバイスから前記情報処理装置に転送する第１のレスポンスキュー制御回路と、を具備し、
前記第２のメモリデバイスは、
情報を記憶する第２のメモリ回路と、
前記第２のリクエスト信号を前記第２のメモリ回路に転送する第２のリクエストキュー制御回路と、
前記第２のレスポンス信号を前記第１のメモリデバイスに出力する第２のレスポンスキュー制御回路と、を具備し、
前記第１のリクエスト信号は、前記第１のリクエスト信号の要求先が前記第１のメモリデバイスであることを示す第１のＩＤ値を含み、

前記第２のリクエスト信号は、前記第２のリクエスト信号の要求先が前記第２のメモリデバイスであることを示す第２のＩＤ値を含み、
前記第１のレスポンス信号は、前記第１のレスポンス信号の転送元が前記第１のメモリデバイスであることを示す第３のＩＤ値を含み、
前記第２のレスポンス信号は、前記第２のレスポンス信号の転送元が前記第２のメモリデバイスであることを示す第４のＩＤ値を含むことを特徴とするメモリモジュール。
【請求項２】
請求項１において、
前記第２のメモリデバイスと接続される第３のメモリデバイスをさらに有し、
前記第１のリクエストキュー制御回路は、第３のリクエスト信号を前記情報処理装置から前記第２のメモリデバイスに転送し、
前記第１のレスポンスキュー制御回路は、第３のレスポンス信号を前記第２のメモリデバイスから前記情報処理装置に転送し、
前記第２のリクエストキュー制御回路は、前記第３のリクエスト信号を前記第１のメモリデバイスから前記第３のメモリデバイスに転送し、
前記第２のレスポンスキュー制御回路は、前記第３のレスポンス信号を前記第３のメモリデバイスから前記第１のメモリデバイスに転送し、
前記第３のメモリデバイスは、
情報を記憶する第３のメモリ回路と、
前記第３のリクエスト信号を前記第２のメモリデバイスから前記第３のメモリ回路に転送する第３のリクエストキュー制御回路と、
前記第３のレスポンス信号を前記第２のメモリデバイスに出力する第３のレスポンスキュー制御回路と、を具備し、
前記第３のリクエスト信号は、前記第３のリクエスト信号の要求先が前記第３のメモリデバイスであることを示す第５のＩＤ値を含み、
前記第３のレスポンス信号は、前記第３のレスポンス信号の転送元が前記第３のメモリデバイスであることを示す第６のＩＤ値を含むことを特徴とするメモリモジュール。
【請求項３】
請求項１において、
前記第１のメモリデバイスは、前記第１のリクエスト信号または前記第２のリクエスト信号に関する入出力回路と、前記第１のレスポンス信号または前記第２のレスポンス信号に関する入出力回路と、を個別に有し、
前記第２のメモリデバイスは、前記第２のリクエスト信号に関する入出力回路と、前記第２のレスポンス信号に関する入出力回路と、を個別に有することを特徴とするメモリモジュール。
【請求項４】
請求項１において、
前記第１のメモリデバイスは、前記第１のリクエスト信号または前記第２のリクエスト信号のためのクロックと、前記第１のレスポンス信号または前記第２のレスポンス信号のためのクロックと、を個別に有し、
前記第２のメモリデバイスは、前記第２のリクエスト信号のためのクロックと、前記第２のレスポンス信号のためのクロックと、を個別に有することを特徴とするメモリモジュール。
【請求項５】
請求項１において、
前記第１のレスポンス信号および前記第２のレスポンス信号は、応答の優先順位に従って出力されることを特徴とするメモリモジュール。
【請求項６】
請求項５において、
前記応答の優先順位は、動的に変化されることを特徴とするメモリモジュール。

【請求項７】
請求項６において、
前記応答の優先順位は、応答回数に応じて変化されることを特徴とするメモリモジュール。
【請求項８】
請求項７において、
前記応答回数は、プログラムできることを特徴とするメモリモジュール。
【請求項９】
請求項８において、
前記応答回数は、前記第１のメモリデバイスまたは前記第２のメモリデバイスに対応した応答回数をプログラムできることを特徴とするメモリモジュール。
【請求項１０】
請求項１において、
前記第１のリクエスト信号または前記第２のリクエスト信号に関する信号には、アドレス情報、命令情報およびメモリデバイス識別情報が含まれ、前記第１のレスポンス信号または前記第２のレスポンス信号に関する信号には、信号データ情報および前記メモリデバイス識別情報が含まれ、夫々多重化されて送受信されることを特徴とするメモリモジュール。
【請求項１１】
請求項２において、
前記第１のリクエスト信号および前記第２のリクエスト信号には、メモリデバイスのクロック周波数の変更を行う命令、クロックの停止を行う命令およびクロックの再起動を行う命令のいずれか１つが含まれることを特徴とするメモリモジュール。
【請求項１２】
請求項１において、
前記第１のメモリデバイスおよび前記第２のメモリデバイスは、エラー情報を出力することを特徴とするメモリモジュール。
【請求項１３】
請求項１２において
前記エラー情報は、識別情報に関するエラー、読み出しに関するエラー、または、書込みに関するエラーであることを特徴とするメモリモジュール。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、不揮発性メモリと情報処理装置を含む情報処理システムおよびメモリモジュールの制御方法に関する。
【背景技術】
【０００２】
従来、フラッシュメモリ(32M bit容量)とスタティックランダムアクセスメモリ(SRAM(4M bit容量))とがスタックチップでＦＢＧＡ（Fine pitch Ball Grid Array）型パッケージに一体封止された複合型半導体メモリがある。フラッシュメモリとSRAMとは、ＦＢＧＡ型パッケージの入出力電極に対してアドレス入力端子とデータ入出力端子が共通化されている。但し各々の制御端子はそれぞれ独立とされている（例えば、非特許文献１参照）。
【０００３】
また、フラッシュメモリ(1GM bit容量)とダイナミックランダムアクセスメモリ(DRAM(５１２Mbit容量))とがスタックチップでＦＢＧＡ（Fine pitch Ball Grid Array）型パッケージに一体封止された複合型半導体メモリがある。フラッシュメモリとダイナミックランダムアクセスメモリとは、ＦＢＧＡ型パッケージの入出力電極に対してアドレス入力端子とデータ入出力端子、および各々の制御端子はそれぞれ独立とされている（例えば、非特許文献２参照）。

【０００４】
また、フラッシュメモリチップとDRAMチップとがリードフレーム型パッケージに一体封止された複合型半導体メモリもある。この複合型半導体メモリはフラッシュメモリとDRAMとはパッケージの入出力電極に対してアドレス入力端子、データ入出力端子、及び制御端子が共通化されて入出力される（例えば、特許文献１の図１及び図１５、特許文献２参照）。
【０００５】
また、主記憶装置として扱われるフラッシュメモリとキャッシュメモリとコントローラとCPUから構成されるシステムもある（例えば、特許文献３の図１参照）。
【０００６】
また、フラッシュメモリとＤＲＡＭと転送制御回路からなる半導体メモリもある（例えば、特許文献４の図２、特許文献５参照）。
【０００７】
また、同一種類のメモリを複数個接続したメモリモジュールがある（特許文献６、特許文献７参照）。
【非特許文献１】“ 複合メモリ（スタックドCSP）フラッシュメモリ＋RAM　データシート ”、形名LRS1380、［online］、平成１３年１２月１０日、シャープ株式会社、［平成１４年８月２１日検索］、インターネット＜ URL http://www.sharp.co.jp/products/ device/ flash/ cmlist.html＞
【非特許文献２】“ MCPデータシート ”、形名KBE00F005A-D411、［online］、平成１７年６月、三星電子株式会社、［平成１８年４月１０日検索］、<URL:http://www.samsung.com/Products/Semiconductor/common/product_list.aspx?family_cd=MCP0>
【特許文献１】特開平０５－２９９６１６号公報
【特許文献２】欧州特許出願公開第０５６６３０６号明細書
【特許文献３】特開平０７－１４６８２０号公報
【特許文献４】特開２００１－５７２３号公報
【特許文献５】特開２００２－３６６４２９号公報
【特許文献６】特開２００２－７３０８号公報
【特許文献７】特開２００４－１９２６１６号公報
【発明の開示】
【発明が解決しようとする課題】
【０００８】
本願発明者等は、本願に先立って携帯電話及びそれに使用されるプロセッサと、フラッシュメモリと、ランダムアクセスメモリから構成された情報処理システムについて検討を行った。
【０００９】
図３６に示すように携帯電話には情報処理装置PRCとメモリモジュールMCM１およびMCM2が使用されている。情報処理装置PRCは中央演算装置CPUとSRAMコントローラSRC、DRAMコントローラDRC及びNAND型フラッシュメモリコントローラNDCから構成される。メモリモジュールMCM１はNOR型フラッシュメモリNOR　FLASHとSRAMから構成される。メモリモジュールMCM２はNAND型フラッシュメモリNAND　FLASHとDRAMから構成される。情報処理装置PRCはメモリモジュールMCM１およびMCM２へアクセスを行い、データの読み出しおよび書き込みを行う。
【００１０】
電源投入後、情報処理装置PRCは、NOR型フラッシュメモリNOR　FLASHに格納されているブートデータを読み出し、自らを立ち上げる。その後、情報処理装置PRCはNOR型フラッシュメモリNOR　FLASHより必要に応じてアプリケーションプログラムを読みだし、中央演算装置CPUで実行する。SRAMおよびDRAMはワークメモリとして機能し、中央演算装置CPUでの演算結果などが保存される。
【００１１】

NAND型フラッシュメモリNAND　FLASHには主に音楽データや動画像データが格納されており、情報処理装置PRCは必要に応じて、NAND型フラッシュメモリNAND　FLASHより、音楽データや動画像データをDRAMへ読み出し、音楽や動画像の再生を行う。近年、携帯電話機に代表されるモバイル機器の多機能化はますます進展しており、多様なインターフェースを取り扱う必要が生じている。

【００１２】

図３６に示すように、現在、CPUは、異なるメモリデバイス毎にコントローラをもち、並列的にメモリと接続されている。さらに、携帯電話が取り扱うアプリケーション、データ、ワークエリアは携帯電話に付加される機能（音楽やゲーム等配信等）が増えるにつれて大きくなり、より大きな記憶容量のメモリが必要となっている。

【００１３】

このため、CPUとメモリを接続する信号配線数が増大し、基板コストの増加、ノイズの増加、信号スキューの増加を招き、携帯電話機の低コスト化、高速化、小型化には対応できないことが判明した。

【００１４】

そこで本発明の目的の一つは、情報処理装置とメモリ間および、メモリとメモリ間の信号配線数を低下させ、高速且つ低コストで、メモリ容量の拡張性を確保できる使い勝手の良い情報システム装置を提供することである。

【課題を解決するための手段】

【００１５】

本発明の代表的な手段を示せば以下の通りである。情報処理装置と、ダイナミックランダムアクセスメモリと、NOR型フラッシュメモリと、NAND型フラッシュメモリと、直列に接続し、一つの封止体に実装し、封止体に半導体チップとの配線を行うための電極と、封止体と封止体外部との接続を行うための電極を設ける。

【００１６】

この際に、情報処理装置から各メモリダイナミックランダムアクセスメモリ、NOR型フラッシュメモリ、NAND型フラッシュメモリへの読み出し要求に要求先の認識情報を含み、さらに、データの読み出しには、転送元の認識情報を含むと良い。

【００１７】

情報処理装置への各メモリ間のデータ読み出し順序は、読み出した回数に応じて動的に決められることが良い。さらに、読み出し回数は、プログラムできることが良い。

【００１８】

電源投入後は、情報処理装置が、直列に接続している各々のメモリへ識別情報を決定する制御を行うと良い。

【００１９】

メモリへ入力した読み出し要求の時間順序には関係なく、遅い読み出しデータを待たずに、早い読み出しデータを送信できる制御にすると良い。

【００２０】

各メモリの読み出し要求を受け付ける回路と、読み出したデータを送信する回路の動作は独立に行える制御にすると良い。

【００２１】

書込み動作と読み出し動作を独立に行える制御にすると良い。

【００２２】

各メモリのクロック周波数は必要に応じて変更できる制御にすると良い。

【００２３】

前記情報処理装置はNAND型フラッシュメモリからデータを読み出し時は、エラー検出と訂正を行い、書きこみ時は、書きこみが正しく行われなかった不良アドレスに対して代替処理を行うと良い。

【発明の効果】

【００２４】

高速且つ低コストで、メモリ容量の拡張性を確保できる使い勝手の良い情報処理システム装置を実現できる。

【発明を実施するための最良の形態】

【００２５】

以下、本発明の実施の形態例につき添付図面を参照しながら詳細に説明する。実施の形態例において各ブロックを構成する回路素子は、特に制限されないが、公知のCMOS（相補型ＭＯＳトランジスタ）等の集積回路技術によって、単結晶シリコンのような１個の半導体基板上に形成される。

【実施例１】

【００２６】

図１は本発明を適用した第１の実施の形態例である情報処理装置CPU_CHIPとメモリモジュールMEMとから構成される情報処理システムを示したものである。以下におのおのについ説明する。

【００２７】

情報処理装置CPU_CHIPは、情報処理回路CPU0、CPU1、CPU2、CPU3とメモリ制御回路CONから構成されている。メモリ制御回路CONは、リクエストキューRqQ、レスポンスキューRsQ、ブートデバイスIDレジスタBotID、最端デバイスIDレジスタEndIDを含む。CPU0、CPU1、CPU2、CPU3では、メモリ制御回路CONを通じて、メモリモジュールMEM0より、OSやアプリケーションプログラムおよびアプリケーションプログラムにて処理を行うデータを読みだし実行する。

【００２８】

リクエストキューRqQは、メモリモジュールMEM0へ出力するためのCPU0、CPU1、CPU2およびCPU3で実行しているアプリケーションプログラムの結果などを格納する。レスポンスキューRsQは、CPU0、CPU1、CPU2およびCPU3へ出力するためのメモリモジュールMEM0から読み出したアプリケーションプログラムなどを格納する。

【００２９】

メモリモジュールMEM0は、メモリチップM0、M1、M2から構成される。また、情報処理装置CPU_CHIPとメモリチップM0、M1、M2は直列に接続されている。メモリチップM0は揮発性メモリであり、メモリチップM1およびM2は不揮発性メモリである。代表的な揮発性メモリには、メモリアレイにダイナミックランダムアクセスメモリセルを用いたDRAM及び疑似スタティックランダムアクセスメモリPSRAM、スタティックランダムアクセスメモリセルを用いたSRAM等があり、本発明には全ての揮発性メモリセルを利用することができる。本実施例ではメモリアレイにダイナミックランダムアクセスメモリセルを用いた例を説明する。

【００３０】

不揮発性メモリにはROM(リードオンリーメモリ)、EEPROM(エレクトリカリイレーサブルアンドプログラマブルROM)、フラッシュメモリ、相変化メモリ、マグネティック・ランダムアクセスメモリMRAM、抵抗スイッチング型ランダムアクセスメモリReRAM等を用いることができる。本実施の形態例ではフラッシュメモリを例に説明する。

【００３１】

また、代表的なフラッシュメモリには、NOR型フラッシュメモリと、AND型フラッシュメモリと、NAND型フラッシュメモリと、ORNAND型フラッシュメモリがあり、本発明には全てのフラッシュメモリを利用することができる。本実施例では、NOR型フラッシュメモリとNAND型フラッシュメモリを例に説明する。

【００３２】

特に限定しないが、メモリチップM0として用いられる典型的な揮発性メモリは、ダイナミックメモリセルを利用したダイナミックランダムアクセスメモリであり、読み出し時間が15ns程度で、約1Gbitの記憶容量を持っている。特に限定しないがメモリチップＭＯは情報処理装置CPU_CHIPにてアプリケーションプログラムを実行するための一時的なワークメモリとして利用される。

【００３３】
特に限定しないが、メモリチップＭ１として用いられる典型的なフラッシュメモリは、NOR型フラッシュメモリセルを利用し、読み出し時間が80ns程度であり、約1Gbitの大きな記憶容量を持っている。特に限定しないが、メモリチップＭ１には情報処理装置CPU_CHIPにて実行するOS、ブートコード、ブートデバイスID値、最端デバイスID値およびアプリケーションプログラムなどが格納される。
【００３４】
特に限定しないが、メモリチップM2として用いられる典型的なフラッシュメモリはNAND型フラッシュメモリセルを利用し、読み出し時間が２５μｓ程度であり、約４Gbit記憶容量を持っている。特に限定しないが、メモリチップＭ１には主に情報処理装置CPU_CHIPにて再生、録音および録画処理を行うために必要な音声データ、静止画像データや動画像データなどが格納される。
【００３５】
メモリチップM0は、初期設定回路INIT、リクエストインターフェース回路ReqIFと、レスポンスインターフェース回路ResIFと、メモリ回路MemVLから構成されている。リクエストインターフェース回路ReqIFは、リクエストクロック制御回路RqCkCおよび、リクエストキュー制御回路RqCTから構成される。レスポンスインターフェース回路ResIFは、レスポンスクロック制御回路RsCkCおよび、レスポンスキュー制御回路RqCTから構成される。メモリ回路MemVLは、特に限定しないが、揮発性メモリであり、ダイナミックランダムアクセスメモリセルを利用したダイナミックランダムアクセスメモリである。リクエストクロック制御回路RqCkCは、クロックドライバ回路Drv1およびクロック分周回路Div1から構成される。メモリチップＭ１は、初期設定回路INIT、リクエストインターフェース回路ReqIFと、レスポンスインターフェース回路ResIFと、メモリ回路MemNV1から構成されている。リクエストインターフェース回路ReqIFは、リクエストクロック制御回路RqCkCおよび、リクエストキュー制御回路RqCTから構成される。レスポンスインターフェース回路ResIFは、レスポンスクロック制御回路RsCkCおよび、レスポンスキュー制御回路RqCTから構成される。
【００３６】
メモリ回路MemNV1は、特に限定しないが、不揮発性メモリであり、NOR型フラッシュメモリセルを利用したNOR型フラッシュメモリである。メモリ回路MemNV1には、ブートデバイスID値および最端デバイスID値が格納される。
【００３７】
リクエストクロック制御回路RqCkCは、クロックドライバ回路Drv1およびクロック分周回路Div1から構成される。
【００３８】
メモリチップＭ２は、初期設定回路INIT、リクエストインターフェース回路ReqIFと、レスポンスインターフェース回路ResIFと、メモリ回路MemNV２から構成されている。メモリチップM2は、直列的に接続しているメモリチップの中で、最も終端のメモリチップであることを示すため、特に限定しないがRqEn3、RsMux3、RqCk3を接地（gnd）している。
【００３９】
リクエストインターフェース回路ReqIFは、リクエストクロック制御回路RqCkCおよび、リクエストキュー制御回路RqCTから構成される。レスポンスインターフェース回路ResIFは、レスポンスクロック制御回路RsCkCおよび、レスポンスキュー制御回路RqCTから構成される。メモリ回路MemNV２は、特に限定しないが、不揮発性メモリであり、NAND型フラッシュメモリセルを利用したNAND型フラッシュメモリである。リクエストクロック制御回路RqCkCは、クロックドライバ回路Drv1およびクロック分周回路Div1から構成される。
【００４０】
メモリチップM0、M1及びM2の初期設定回路INITは電源投入直後に、それぞれのメモリチップに対し初期設定を行う。メモリチップM0、M1及びM2のリクエストキュー制御回路RqCTには、それぞれのメモリチップのID番号を格納するIDレジスタが設けられている。電源投

入直後に先ず、初期設定回路INITによって初期設定され、次に、情報処理装置CPU_CHIPによってメモリチップM0、M1、M2のID番号が決定され、それぞれのメモリチップ内のIDレジスタへID番号が格納される。

【００４１】

メモリチップM0、M1及びM2は、特に限定しないが、それぞれブートデバイス認識信号Bsigを持っており、このブートデバイス認識信号Bsigが接地（gnd）されている場合は、そのメモリチップが電源投入直後の動作を行うためのブートプログラムを格納しているブートデバイスであることを示す。ブートデバイス認識信号Bsigが電源(vdd)に接続されている場合は、そのメモリチップがブートデバイスではないことを示す。特に限定しないが、メモリチップM1がブートデバイスであり、メモリチップM0およびM2はブートデバイスに設定されていない。また、ブートデバイス認識信号Bsigによって、どのチップをブートデバイスにするかをプログラムすることができる。

【００４２】

RqCk0、RqCK1およびRqCk2は、リクエストクロックであり、RsCk0、RsCK1およびRsCk2はレスポンスクロックである。RqEN0、RqEN1およびRqEN2は、リクエストイネーブル信号であり、RsEN0、RsEN1およびRsEN2はレスポンスイネーブル信号である。RqMux0、RqMux1およびRqMux2は、リクエスト信号であり、RsMux0、RsMux1およびRsMux2はレスポンス信号である。

【００４３】

メモリチップM0は、特に限定しないが、情報処理装置CPU_CHIPからのリクエストを受け付けることが可能であればRqEN0をHighにし、受け付けることが不可能であればRqEN0をLowにする。メモリチップＭ１は、特に限定しないが、メモリチップM0からのリクエストを受け付けることが可能であればRqEN1をHighにし、受け付けることが不可能であればRqEN1をLowにする。メモリチップM2は、特に限定しないが、メモリチップM1からのリクエストを受け付けることが可能であればRqEN2をHighにし、受け付けることが不可能であればRqEN2をLowにする。

【００４４】

RqMux0、RqMux1およびRqMux2は、リクエスト信号であり、これらリクエスト信号を通じて送信されるリクエストは、特に限定しないがID値、コマンド、アドレス及び書き込みデータなどが、多重化され、それぞれのリクエストクロックRqCk0、RqCk1およびRqCk2に同期して送信される。RsMux0、RsMux1およびRsMux2のレスポンス信号であり、これらレスポンス信号を通じて送信されるレスポンスは、特に限定しないがID値及び読み出したデータなどが、多重化され、それぞれのレスポンスクロックRsCk0、RsCk1、RsCk2に同期して送信される。

【００４５】

以下に本メモリシステムの動作を説明する。先ず、電源投入直後の動作について説明する。

【００４６】

＜電源投入直後の動作説明＞

先ず、電源投入直後の本メモリシステムの動作について説明する。

【００４７】

情報処理装置CPU_CHIPへ電源が投入されると、ブートデバイスIDレジスタBotIDを１へ、最端デバイスIDレジスタEndIDを０へ設定する。

【００４８】

メモリチップM0へ電源が投入されると、自身の初期設定回路INITが、自身のリクエストキュー制御回路RqCT、レスポンスキュー制御回路RsCT、リクエスト制御回路RqCkc、レスポンスクロック制御回路RsCkC、クロック分周回路Div1、Div2およびメモリ回路MemVLを初期設定する。リクエストキュー制御回路RqCTが持っているIDレジスタを０へ、ID有効ビットをLowへ設定する。レスポンスキュー制御回路RsCTが持つレスポンス調停回路のレスポンス優先順位に関して、メモリチップM0のレスポンス優先順位は１へ、メモリチップM1の

レスポンス優先順位は2へ、メモリチップM2のレスポンス優先順位は3へ初期設定される。クロック分周回路Div1およびDiv2の分周比は１に設定される。

【００４９】

メモリチップM1へ電源が投入されると、自身の初期設定回路INITが、自身のリクエストキュー制御回路RqCT、レスポンスキュー制御回路RsCT、リクエスト制御回路RqCkc、レスポンスクロック制御回路RsCkC、クロック分周回路Div1、Div2およびメモリ回路MemNV1を初期設定する。リクエストキュー制御回路RqCTが持っているIDレジスタを0へ、ID有効ビットをLowへ設定する。メモリチップM1のレスポンスキュー制御回路RsCTが持つレスポンス調停回路のレスポンス優先順位に関して、メモリチップM1のレスポンス優先順位は1へ、メモリチップM2のレスポンス優先順位は２へ初期設定される。クロック分周回路Div1およびDiv2の分周比は１に設定される。

【００５０】

メモリチップM2へ電源が投入されると、自身の初期設定回路INITが、自身のリクエストキュー制御回路RqCT、レスポンスキュー制御回路RsCT、リクエスト制御回路RqCkc、レスポンスクロック制御回路RsCkC、クロック分周回路Div1、Div2およびメモリ回路MemNV2を初期設定する。メモリチップM2のリクエストキュー制御回路RqCTが持っているIDレジスタを0へ、ID有効ビットをLowへ設定する。メモリチップM2のレスポンスキュー制御回路RsCTが持つレスポンス調停回路のレスポンス優先順位に関してメモリチップM2のレスポンス優先順位は1へ初期設定される。クロック分周回路Div1およびDiv2の分周比は１に設定される。次に、メモリチップM2は、ブートデバイス認識信号Bsigが電源に接続されているので、自分自身はブートデバイスではないことを認識する。

【００５１】

また、情報処理装置CPU_CHIPからリクエストクロックRqCk0がメモリチップM0へ入力され、メモリチップM0のクロックドライバDrv1を通じてクロック分周回路Div1およびクロック信号ck1としてクロック分周回路Div2へ出力される。クロック分周回路Div1へ入力したクロックは、リクエストクロックRqCk1を通じてメモリチップM1へ出力する。クロック分周回路Div1へ入力したクロックは、クロック信号ck2から出力され、また、リクエストクロックRqCk1を通じてメモリチップM2へ出力する。クロック分周回路Div2へ入力したクロックはクロック信号ck3から出力され、また、レスポンスクロックRsCk0を通じて情報処理装置CPU_CHIPへ出力する。メモリチップM1のクロックドライバDrv1へ入力されたクロックは、クロック分周回路Div1およびクロック信号ck1としてクロック分周回路Div2へ出力される。クロック分周回路Div1へ入力したクロックは、クロック信号ck2から出力され、また、リクエストクロックRqCk1を通じてメモリチップM2へ出力する。クロック分周回路Div2へ入力したクロックは、クロック信号ck3から出力され、また、レスポンスクロックRsCk1を通じてメモリチップM0へ出力する。レスポンスクロックRsCk1を通じてメモリチップM0のクロックドライバDrv2へ入力されたクロックはクロック信号ck4へ出力される。メモリチップM2のクロックドライバDrv1へ入力されたクロックはクロック分周回路Div1およびおよびクロック信号ck1としてクロック分周回路Div2へ出力される。クロック分周回路Div2へ入力したクロックはクロック信号ck3から出力され、またリクエストクロックRqCk1を通じてメモリチップM2へ出力する。レスポンスクロックRsCk2を通じてメモリチップM1のクロックドライバDrv2へ入力されたクロックはクロック信号ck4へ出力される。

【００５２】

次に、メモリチップM0は、ブートデバイス認識信号Bsigが電源vddに接続されているので、自分自身はブートデバイスではないことを認識する。メモリチップM1は、ブートデバイス認識信号Bsigが接地gndされているので、自分自身がブートデバイスであることを認識し、自らのメモリ回路MemNV1が保持しているブートデバイスID値１をIDレジスタへ設定し、ID有効ビットをHighにする。メモリチップM2は、ブートデバイス認識信号Bsigが電源に接続されているので、自分自身はブートデバイスではないことを認識する。さらに、メモリチップM2は、RqEn3、RsMux3、RqCk3を接地（gnd）していることによって、直列接続しているメモリチップの最も終端のメモリチップであることを認識し、リクエストイネー

ブル信号RqEn2をHighにする。

【００５３】

次に、メモリチップM1はリクエストイネーブル信号RqEn2がHighになったことを確認し、レスポンスイネーブル信号RsEn2及びリクエストイネーブル信号RqEn1をHighにする。次に、メモリチップM0はリクエストイネーブル信号RqEn1がHighになったことを確認し、レスポンスイネーブル信号RsEn1及びリクエストイネーブル信号RqEn0をHighにする。最後に、情報処理装置CPU_CHIPは、リクエストイネーブル信号RqEn0がHighになったことを確認し、各メモリチップの信号接続が確認されたことを知り、レスポンスイネーブル信号RsEn0をHighにする。これによって、情報処理装置CPU_CHIPおよびメモリチップM0、M1、M2が直列接続されていることが正しく確認できる。

【００５４】

次に、各メモリチップの信号接続の確認後に行われるブートデータの読み出し方法について説明する。

【００５５】

情報処理装置CPU_CHIPは、ブートデバイスIDレジスタBotIDの値１を読み出し、リクエスト信号RqMux0を通じて、メモリチップM1のID値１、読みだし命令、転送データサイズおよびアドレスを多重化したリクエストReqBRD1をクロック信号RqCK0に同期させ、メモリチップM0へ転送する。メモリチップM0のID有効ビットがLowのため、メモリチップM0は、情報処理装置CPU_CHIPからのリクエストReqBRD1はメモリチップM0へのリクエストではないと判断し、リクエスト信号RqMux1を通じて、リクエストReqBRD1をクロック信号RqCK1に同期させメモリチップM1へ転送する。

【００５６】

メモリチップＭ１は、メモリチップM0からのリクエストReqBRD1を、自身のリクエストキュー制御回路RqCTへ格納する。その後、リクエストキュー制御回路RqCTはリクエストに含まれるID値１と自身のIDレジスタの値１を比較する。双方は一致しており、ID有効ビットがHighのため、メモリチップＭ１は、メモリチップM0からのリクエストを自身へのリクエストであると判断する。

【００５７】

その後、リクエストReqBRD1に含まれる読み出し命令、転送データサイズおよびアドレスによって、メモリ回路MemNV1からブートデータが、最終端デバイスIDレジスタから番号３が読み出され、レスポンスキュー制御回路RsCTへ転送される。また同時に、リクエストキュー制御回路RqCTが格納しているIDレジスタ値１もレスポンスキュー制御回路RsCTへ転送される。

【００５８】

メモリチップM1のレスポンスキュー制御回路RsCTはレスポンス信号RqMux1を通じて、メモリチップM1のID値１、ブートプログラムおよび最終端デバイスIDを多重化したレスポンスResBRD1をクロック信号RqCK1に同期させ、メモリチップM0へ転送する。

【００５９】

最後に、メモリチップM0のレスポンスキュー制御回路RsCTはレスポンス信号RqMux0を通じて、レスポンスResBRD1をクロック信号RqCK0に同期させ、情報処理装置CPU_CHIPへ転送する。

【００６０】

情報処理装置CPU_CHIPは、レスポンスResBRD1をレスポンスキューRsQへ格納する。レスポンスResBRD1に含まれるID値１により、ブートデータおよび最終端デバイスID値３が、メモリチップM1から送信されたことを知ることができる。最終端デバイスID値３はメモリ制御回路CON内の最終端デバイスIDレジスタへ保存される。

【００６１】

情報処理装置CPU_CHIPは、ブートプログラムによって自らを立ち上げ、次に各メモリチップM0、M1、M2へID番号の割り当てを行う。

【００６２】

次に、各メモリチップへのID番号付けについて説明する。情報処理装置CPU_CHIPはブートコードに従い、先ず、各メモリチップへのID番号付けを行う。情報処理装置CPU_CHIPは、リクエスト信号RqMux0を通じて、ID番号２とID設定命令をメモリチップM0へ転送する。メモリチップM0では、ID有効ビットがLowのため、まだID番号付けが行われていない。そこで、メモリチップM0は、ID番号２とID設定命令によってIDレジスタへID番号２を設定し、ID有効ビットをHighにする。ID有効ビットがHighとなることで、ID番号付けが完了したことを示す。メモリチップM0のID番号付けが完了すると、メモリチップM0はレスポンス信号RsMux0を通じて、メモリチップM0のID値２およびID番号付け完了情報を出力する。情報処理装置CPU_CHIPは、メモリチップM0のID値２およびID番号付け完了情報を受け取り、メモリチップM0のID番号付けが完了したことを知る。

【００６３】

次に、情報処理装置CPU_CHIPは、リクエスト信号RqMux0を通じてID番号３とID設定命令を多重化したリクエストReqID3をメモリチップM0へ転送する。メモリチップＭ０は自身のID番号２とリクエストReqID3に含まれるID番号３とを比較し、不一致のため、リクエストReqID3をメモリチップＭ１へ転送する。

【００６４】

メモリチップM1は自身のID番号1とリクエストReqID3に含まれるID番号３とを比較し、不一致のため、リクエストReqID3をメモリチップＭ２へ転送する。メモリチップＭ２では、ID有効ビットがLowのため、まだID番号付けが行われていない。そこで、メモリチップＭ２は、リクエストReqID3に含まれるID番号３とID設定命令によってメモリチップM2のIDレジスタへID番号３を設定し、ID有効ビットをHighにする。最終端のメモリチップM2のID番号付けが完了すると、メモリチップM2はレスポンス信号RqMux２を通じて、メモリチップＭ２のID値３およびID番号付け完了情報を多重化したレスポンスResID3をメモリチップM1へ出力する。メモリチップM1はレスポンス信号RqMux1を通じてレスポンスResID3をメモリチップM0へ出力する。メモリチップM0はレスポンス信号RqMux0を通じてレスポンスResID3を情報処理装置CPU_CHIPへ転送する。情報処理装置CPU_CHIPは、レスポンスResID3を受け取り、このレスポンスResID3に含まれるメモリチップＭ２のID値3およびID番号付け完了情報を受け取り、メモリチップM2のID番号付けが完了したことを知る。さらに、情報処理装置CPU_CHIPは、転送されたメモリチップＭ２のID値３と、メモリ制御回路CON内の最終端デバイスIDレジスタに設定されている最終端デバイスID値３とを比較し、双方が一致したことで、最終端のメモリチップまでID番号付けが行われたことを確認する。この後、メモリモジュールMEM0は情報処理装置CPU_CHIPからのリクエストを待つアイドル状態となる。

【００６５】

このように、電源投入直後に、直列接続の確認動作を行うことで、確実にメモリ同士が接続されていることが確認できる。さらに、ブートデバイスおよび、最端のメモリチップを明示し、自動的に各メモリへのID付けが行われることで、容易に、必要な分だけメモリチップを接続し、メモリ容量を拡張することができる。

【００６６】

＜通常動作の説明＞

電源投入時のパワーオンシーケンスが終了した後のメモリモジュールMEM0と情報処理装置CPU_CHIP間のデータ転送について説明する。

【００６７】

特に限定しないが、メモリチップM0、M1、M2のそれぞれのIDレジスタ値が２、１及び３に設定された場合の、メモリモジュールMEM0と情報処理装置CPU_CHIP間のデータ転送について説明する。特に限定しないが、メモリチップM0、M1、M2のリクエストキュー制御回路RqCTにはリクエストキューは2つ存在し、リクエストがエントリされていない状態であり、レスポンスキュー制御回路RsCTにはレスポンスキューが4つ存在し、レスポンスがエントリされていない空の状態である場合のデータ転送について説明する。特に限定しないが、1つのリクエストキューは１バイトのID値、１バイトの命令、2バイトのアドレス、３２バイトの読み出しデータを格納でき、1つのレスポンスキューは１バイトのID値、３２バ

イトの読み出しデータを格納できる。

【００６８】

また、特に限定しないが、メモリチップM0、M1、M2のそれぞれのメモリ回路MemVL、MemNV1、MemNV2は４つのメモリバンクから構成されており、１つのメモリバンクには１つのセンスアンプ回路が装置されている。

【００６９】

メモリチップM0は、自身のリクエストキューに情報処理装置CPU_CHIPからのリクエストがエントリされていないため、リクエストイネーブル信号RqEn0をHighにし、リクエストを受け付けることができることを情報処理装置CPU_CHIPへ知らせる。

【００７０】

情報処理装置CPU_CHIPは、リクエスト信号RqMux0を通じて、ID値２、バンクアクティブ命令BA、バンクアドレスBK０、ロウアドレスRow０を多重化したリクエストReqBAm01をクロック信号RqCK0に同期させ、メモリチップM0へ転送する。

【００７１】

続いて、リクエスト信号RqMux0を通じて、ID値２、4バイト読み出し命令RD、バンクアドレスBK０、カラムアドレスCol３を多重化したリクエストReqRDm04をクロック信号RqCK0に同期させ、メモリチップM0へ転送する。

【００７２】

メモリチップM0は、情報処理装置CPU_CHIPからのリクエストReqBAm０１とリクエストReqRDm04を順に、自身のリクエストキュー制御回路RqCTへ格納する。

【００７３】

これで、リクエストキュー制御回路RqCT内の全てのリクエストキューはエントリされ、情報処理装置CPU_CHIPからの新たなリクエストを受け付けることができないため、リクエストイネーブル信号RqEn0をLowにする。リクエストイネーブル信号RqEn0がLowになったことで、情報処理装置CPU_CHIPは、メモリチップM0が、リクエストを受け付けられなくなったことを知ることができる。

【００７４】

その後、リクエストキュー制御回路RqCTはリクエストReqBAm０１に含まれるID値２と自身のIDレジスタの値２を比較する。リクエストReqBA１に含まれるID値２とメモリチップM0のIDレジスタ値２は一致しているため、リクエストキュー制御回路RqCTはリクエストReqBA１をメモリ回路MemVLへ送信する。メモリ回路MemVLは、リクエストReqBAm０１にバンクアクティブ命令BA、バンクアドレスBK0、ロウアドレスRow０によって、バンク０内のロウ0に接続されている8192ビット分のメモリセルが活性化されセンスアンプへ転送される。

【００７５】

リクエストReqBAm０１が処理されたことによって、リクエストキュー制御回路RqCT内のリクエストキューがひとつ分空いたため、メモリチップM0はリクエストイネーブル信号RqEn0をHighにし、新たなリクエストを受け付け可能であることを情報処理装置CPU_CHIPへ知らせる。

【００７６】

次に、リクエストキュー制御回路RqCTはリクエストReqRDm04に含まれるID値２と自身のIDレジスタの値２を比較する。リクエストReqRDm04に含まれるID値２とメモリチップM0のIDレジスタ値２はまた一致しているため、リクエストキュー制御回路RqCTはリクエストReqRDm04をメモリ回路MemVLへ送信する。メモリ回路MemVLは、リクエストReqRDm04に含まれる４バイト読み出し命令RD4、バンクアドレスBK０、カラムアドレスCol３によって、メモリ回路MemVLのバンク０のセンスアンプに保持されているデータのなかで、カラムアドレス３を開始アドレスとした４バイト分のデータが読み出し、IDレジスタ値２を含めて、レスポンスキュー制御回路RsCTへレスポンスResRDm04として転送される。リクエストReqRDm04がメモリ回路MemNV1へ送信されてから、所望のデータが読み出されレスポンスキュー制御回路RsCTへレスポンスResRDm04として入力されるまでの時間は、特に限定しないが、15ns程度である。

【００７７】
レスポンスキュー制御回路RsCTは、レスポンス信号RsMux0を通じて、レスポンスRsRDm04を情報処理装置CPU_CHIPへ出力する。情報処理装置CPU_CHIPのメモリ制御回路CONはレスポンスRsRDm04を、レスポンスキューRsQへ受け取る。情報処理装置CPU_CHIPは、レスポンスキューRsQ送信されたレスポンスRsRDm04に含まれるID値２によって、リクエストRqRDm04に対応するデータが正しくメモリチップM0から送信されたことを確認できる。
【００７８】
特に限定しないが、レスポンスキューRsQへ入力したデータは情報処理回路CPU0、CPU1、CPU2およびCPU3のいずれかでデータ処理が行われる。上記では、メモリチップM0でのデータの読み出しについて説明したが、データの書込みについても同様の動作を実行できることは言うまでもない。
【００７９】
以上説明したように、情報処理装置CPU_CHIPからメモリモジュールMEM0へのリクエスト及びメモリモジュールMEM0から情報処理装置CPU_CHIPへのレスポンスにID情報を含むことで、正しくデータ転送が行えたことを確認でき、情報処理装置CPU_CHIPおよびメモリチップM0、M1、M2の直列接続によって、接続信号数を減少させながらも、情報処理装置CPU_CHIPは所望の処理を実行することができる。
【００８０】
次に、情報処理装置CPU_CHIPとメモリチップM1とのデータ転送について説明する。情報処理装置CPU_CHIPは、リクエスト信号RqMux0を通じて、ID値１、4バイトデータ読み出し命令NRD4、アドレスAdd31を多重化したリクエストReqNRD4m1をメモリチップM0へ転送する。メモリチップM0は、情報処理装置CPU_CHIPからのリクエストReqNRD4m1を自身のリクエストキュー制御回路RqCTへ格納し、リクエストReqNRD4m1に含まれるID値１と自身のIDレジスタの値２を比較する。比較結果は不一致のため、メモリチップＭ０はリクエストReqNRD4m1を自身へのリクエストではないと判断し、リクエスト信号RqMux1を通じて、メモリチップM1へ転送する。
【００８１】
メモリチップＭ１は、メモリチップM0からのリクエストReqNRD4m1を自身のリクエストキュー制御回路RqCTへ格納し、リクエストReqNRD4m1に含まれるID値１と自身のIDレジスタの値１を比較する。リクエストキュー制御回路RqCTはリクエストReqNRD4m1に含まれるID値1と自身のIDレジスタの値1を比較し、一致しているため、リクエストReqNRD4m1をメモリ回路MemNV1へ送信する。リクエストReqNRD4m1に含まれる4バイト読み出し命令NRD4、アドレスAdd31によって、アドレス３1を開始アドレスとした4バイト分のデータがメモリ回路MemNV1から読み出され、IDレジスタ値1を含めて、レスポンスキュー制御回路RsCTへレスポンスResNRD4m1として転送される。　リクエストReqNRD4m1がメモリ回路MemNV1へ送信されてから、所望のデータが読み出されるまでの時間は、特に限定しないが、80ns程度である。
【００８２】
レスポンスキュー制御回路RsCTは、レスポンス信号RsMux1を通じて、レスポンスResNRD4m1をメモリチップM0へ出力する。メモリチップM0のレスポンスキュー制御回路RsCTは受け取ったレスポンスResNRD4m1をレスポンス信号RsMux0から情報処理装置CPU_CHIPへ出力する。上記では、メモリチップM1でのデータの読み出しについて説明したが、データの書込みについても同様の動作を実行できることは言うまでもない。
【００８３】
以上説明したように、リクエストへIDを付加することで、情報処理装置CPU_CHIPからメモリチップM0を介して、メモリチップM1へリクエストが確実に転送される。また、レスポンスへIDを付加することで、メモリチップM1から読み出されメモリチップM0を介して情報処理装置CPU_CHIPが受け取ったデータは、メモリチップM1へのリクエストに対応したメモリチップM1から読み出されたデータであることを確認でき、情報処理装置CPU_CHIPおよびメモリチップM0、M1、M2の直列接続によって、接続信号数を減少させながらも、情報処理

装置CPU_CHIPは所望の処理を実行することができる。

【００８４】

次に、情報処理装置CPU_CHIPとメモリチップＭ２とのデータ転送について説明する。特に限定しないがメモリチップM2はNAND型のフラッシュメモリセルを利用したNAND型フラッシュメモリである。NAND型フラッシュメモリは書き換えを繰り返すことによって、信頼性が低下し、書き込み時に書いたデータが、読み出し時には異なるデータとなったり、書き換え時にデータが書き込まれなかったりすることが稀にあるため、512Byte分のデータと、この512Byte分のデータにエラーが発生した際に、そのエラーを訂正するための16Byte分のECCコードが１ページ分のデータとして管理されている。

【００８５】

情報処理装置CPU_CHIPは、リクエスト信号RqMux0を通じて、ID値3、1ページ（512Byte+16Byte）データ読み出し命令NDRDp1、ページアドレスPadd1を多重化したリクエストReqNDRDp1m2をメモリチップM0へ転送する。メモリチップM0は、情報処理装置CPU_CHIPからのリクエストReqNDRDp1m2を自身のリクエストキュー制御回路RqCTへ格納し、リクエストReqNRDp1m2に含まれるID値3と自身のIDレジスタの値２を比較する。比較結果は不一致のため、メモリチップＭ０はリクエスト信号RqMux1からリクエストReqNDRDp1m2をメモリチップM1へ転送する。

【００８６】

メモリチップＭ１は、メモリチップM0からのリクエストReqNDRDp1m2を自身のリクエストキュー制御回路RqCTへ格納し、リクエストReqNDRDp1m2に含まれるID値３と自身のIDレジスタの値１を比較する。比較結果は不一致のため、メモリチップＭ１はリクエスト信号RqMux2からリクエストReqNDRDp1m2をメモリチップM2へ転送する。メモリチップM2は、メモリチップM1からのリクエストReqNDRDp1m2を自身のリクエストキュー制御回路RqCTへ格納し、リクエストReqNDRDp1m2に含まれるID値３と自身のIDレジスタの値3を比較する。比較結果は一致しているため、リクエストReqNDRDp1m2をメモリ回路MemNV2へ送信する。

【００８７】

リクエストReqNDRDp1m2に含まれる1ページ読み出し命令NDRDp1、ページアドレスPadd1によって、ページアドレス1を開始アドレスとした1ページ(512Byte)分のデータ及びECCコード(16Byte)がメモリ回路MemNV2から読み出され、メモリ回路MemNV2内のデータレジスタへ転送される。次に、レスポンスキュー制御回路RsCTは、データレジスタ内のデータを32Byte単位で、IDレジスタ値3を含めて、レスポンスResNDRDp1m2-0～レスポンスResNDRDp1m2-7として順に読み出し、メモリチップM1へ転送する。最後に、ページアドレス1内の16Byte分のECCコードを読み出し、IDレジスタ値3を含めてレスポンスResNDRDp1m2ECCとして、レスポンス信号RsMux2を通じてM1へ転送する。リクエストReqNDRDp1m2がメモリ回路MemNV2へ送信されてから、所望のデータがメモリ回路MemNV2内のデータレジスタへ読み出されるまでの時間は特に限定しないが、25usec程度である。

【００８８】

レスポンスResNDRDp1m2-0、ResNDRDp1m2-1、ResNDRDp1m2-2、ResNDRDp1m2-3、ResNDRDp1m2-4、ResNDRDp1m2-5、ResNDRDp1m2-6、レスポンスResNDRDp1m2-7及び、レスポンスResNDRDp1m2ECCは、順にメモリチップM1へ転送された後、レスポンス信号RsMux1通じてメモリチップM0へ転送され、さらに、レスポンス信号RsMux0を通じて、報処理装置CPU_CHIPへ転送される。

【００８９】

情報処理装置CPU_CHIPのメモリ制御回路CONは順に、レスポンスResNDRDp1m2-0、ResNDRDp1m2-1、ResNDRDp1m2-2、ResNDRDp1m2-3、ResNDRDp1m2-4、ResNDRDp1m2-5、ResNDRDp1m2-6、レスポンスResNDRDp1m2-7及び、レスポンスResNDRDp1m2ECCを、レスポンスキューRsQへ受け取る。情報処理装置CPU_CHIPは、レスポンスキューRsQ送信されたこれらレスポンスに含まれるID値２によって、これらレスポンスがメモリチップM2から送信されたことを確認できる。

【００９０】

情報処理装置CPU_CHIPは、メモリチップM2から送信されたデータに対し、情報処理回路CPU0、CPU1、CPU2、CPU3のいずれかにて、ECCコードを利用しエラー検出を行う。エラーがなければ、そのデータに対し情報処理回路CPU0、CPU1、CPU2、CPU3のいずれかがデータ処理を行う。エラーがあれば情報処理回路CPU0、CPU1、CPU2、CPU3のいずれかにてエラー訂正を行った後、エラー訂正が行われたデータに対し情報処理回路CPU0、CPU1、CPU2、CPU3のいずれかデータ処理を行う。上記では、メモリチップＭ２でのデータの読み出しについて説明したが、データの書込みについても同様の動作を実行できることは言うまでもない。

【００９１】

以上説明したように、リクエストへIDを付加することで、情報処理装置CPU_CHIPからメモリチップM0およびM1を介して、メモリチップＭ２へリクエストが確実に転送される。また、レスポンスへIDを付加することで、メモリチップＭ２から読み出され、メモリチップM0およびM1を介して情報処理装置CPU_CHIPが受け取ったデータは、メモリチップＭ２へのリクエストに対応したメモリチップＭ２から読み出されたデータであることを確認でき、情報処理装置CPU_CHIPおよびメモリチップM0、M1、M2の直列接続によって、接続信号数を減少させながらも、情報処理装置CPU_CHIPは所望の処理を実行することができる。

【００９２】

次に、情報処理装置CPU_CHIPがデータ読み出しリクエストに続いてデータ書込みリクエストをメモリモジュールMEM0へ送信した場合のデータ転送について説明する。

【００９３】

情報処理装置CPU_CHIPがリクエスト信号RqMux0を通じて、ID値２、8バイトデータ読み出し命令RD8、バンクアドレスBK1、カラムアドレスCol15を多重化したリクエストReqRD8b1m0をメモリチップM0へ転送する。続いて、リクエスト信号RqMux0を通じて、ID値2、8バイトデータ書き込み命令WT8、バンクアドレスBK1、カラムアドレスCol３１、及び8バイト分の書き込みデータを多重化したリクエストReqWT8b1m0をメモリチップM0へ転送する。

【００９４】

メモリチップM0は、情報処理装置CPU_CHIPからのリクエストReqRD8b1m0とリクエストReqWT8b1m0を順に、自身のリクエストキュー制御回路RqCTへ格納する。リクエストキュー制御回路RqCTはリクエストReqRD8b1m0に含まれるID値２と自身のIDレジスタの値２を比較し、一致しているため、リクエストReqRD8b1m0をメモリ回路MemVLへ送信する。

【００９５】

メモリ回路MemVLはリクエストReqRD8b1m0に含まれる8バイト読み出し命令RD8、バンクアドレスBK1、カラムアドレスCol31によって、メモリ回路MemVLのバンク1のセンスアンプに保持されているデータのなかで、カラムアドレス15を開始アドレスとした8バイト分のデータを読み出し、IDレジスタ値２を含めて、レスポンスキュー制御回路RsCTへレスポンスRsRD8b1m0として転送する。

【００９６】

レスポンスキュー制御回路RsCTは、レスポンス信号RsMux0を通じて、IDレジスタ値２および8バイトデータを含むレスポンスRsRD8b1m0を情報処理装置CPU_CHIPへ出力する。

【００９７】

リクエストReqRD8b1m0が処理されたことによって、リクエストキュー制御回路RqCTはリクエストReqWT8b1m0に含まれるID値２と自身のIDレジスタの値２を比較し、一致しているため、リクエストReqWT8b1m0をメモリ回路MemVLへ送信する。

【００９８】

メモリ回路MemVLはリクエストReqWT8b1m0に含まれる8バイト書き込み命令WT8、バンクアドレスBK1、カラムアドレスCol31によって、メモリ回路MemVLのバンク1のセンスアンプへカラムアドレス３１を開始アドレスとした8バイト分のデータが書き込まれ、さらにメモリバンク１へ書き込まれる。

【００９９】

リクエストキュー制御回路RqCTとレスポンスキュー制御回路RsCTはそれぞれ独立に動作

するため、、リクエストReqRD8b1m0に対応するレスポンスRsRD8b1m0が情報処理装置CPU_CHIPへ出力されている最中でもリクエストReqWT8b1m0の書込み動作を実行することができる。

【０１００】

以上説明したように、リクエストインターフェース回路ReqIFとレスポンスインターフェース回路は独立に動作可能なため、データの読み出し動作と書込み動作を同時に実行でき、データ転送性能を向上させることができる。上記では、メモリチップM0でのデータの読み出し及び書込みについて説明したが、他のメモリチップM1及びM2においても同様の動作が実行できることは言うまでもない。さらに、それぞれののメモリチップにおいてリクエストインターフェース回路ReqIFとレスポンスインターフェース回路は独立に動作可能なため、異なるメモリチップへのデータ読み出し及び書込みリクエストが生じた場合でも、それぞれのリクエストを独立に並列に処理でき、データ転送性能を向上できることは言うまでもない。

【０１０１】

次に、情報処理装置CPU_CHIPからメモリチップＭ１へ読み出しリクエストが生じ、その後、連続してメモリチップＭ０へ読み出しリクエストが生じた場合のデータ転送について説明する。情報処理装置CPU_CHIPは、最初にリクエスト信号RqMux0を通じて、ID値１、4バイトデータ読み出し命令NRD4、アドレスAdd63を多重化したリクエストReqNRD4m1をメモリチップM0へ転送する。

【０１０２】

次に、リクエスト信号RqMux0を通じて、ID値２、4バイト読み出し命令RD4、バンクアドレスBK3、カラムアドレスCol15を多重化したリクエストReqRD4b3m0をメモリチップM0へ転送する。メモリチップM0は、情報処理装置CPU_CHIPからのリクエストReqNRD4m1とリクエストReqRD4b3m0を順に、自身のリクエストキュー制御回路RqCTへ格納する。

【０１０３】

メモリチップM0のリクエストキュー制御回路RqCTは、リクエストReqNRD4m1に含まれるID値１と自身のIDレジスタの値２を比較し、一致していないため、リクエストReqNRD4m1をリクエスト信号RqMux1からメモリチップM1へ転送する。

【０１０４】

次に、メモリチップM0のリクエストキュー制御回路RqCTは、リクエストReqRD4b3m0に含まれるID値２と自身のIDレジスタの値２を比較し、一致するため、リクエストReqRD4b3m0はメモリ回路MemVLへ転送される。リクエストReqRD4b3m0によって、約15ns後にメモリ回路MemVLから4バイトのデータが読み出され、レスポンスキュー制御回路RsCTへレスポンスResRD4b3m0として入力される。レスポンスキュー制御回路RsCTは、レスポンス信号RsMux0を通じて、レスポンスResRD4b3m0を情報処理装置CPU_CHIPへ送信する。

【０１０５】

メモリチップM0が、リクエストReqRD4b3m0に対する読み出し動作を行っているのと平行に、メモリチップM1のリクエストキュー制御回路RqCTは、リクエストReqNRD4m1に含まれるID値１と自身のIDレジスタの値1を比較し、一致するため、リクエストReqNRD4m1はメモリ回路MemNV1へ転送される。リクエストReqNRD4m1によって約80ns後にメモリ回路MemNV1から4バイトのデータが読み出され、レスポンスキュー制御回路RsCTへレスポンスResNRD4m1として入力される。メモリチップM1のレスポンスキュー制御回路RsCTは、レスポンスResNRD4m1をレスポンス信号RsMux1よりメモリチップM0へ送信し、さらに、レスポンス信号RsMux0より情報処理装置CPU_CHIPへ送信する。

【０１０６】

情報処理装置CPU_CHIPが、メモリチップM1に対するリクエストReqNRD4m1をメモリモジュールMEM0へ発行してからメモリチップM1のリクエストキュー制御回路RqCTへリクエストがReqNRD4m1完全に格納されるまでの時間は10ns程度、リクエストキュー制御回路RqCTがメモリ回路MemNV1へリクエストReqNRD4m1を送信する時間は1ns程度、メモリ回路MemNV1から4バイトのデータが読み出され、レスポンスキュー制御回路RsCTへレスポンスResNRD4m1

として入力されまでの時間が80ns程度、レスポンスResNRD4m1が情報処理装置CPU_CHIPへ到達するまでの時間が10ns程度である。したがって、情報処理装置CPU_CHIPが、メモリチップM1に対するリクエストReqNRD4m1を発行してからレスポンスをResNRD4m1受け取るまでの時間は、101ns程度となる。

【０１０７】

情報処理装置CPU_CHIPが、メモリチップM0に対するリクエストReqRD4b3m0をメモリモジュールMEM0へ発行してからメモリチップM0のリクエストキュー制御回路RqCTへリクエストReqRD4b3m0が完全に格納されるまでの時間は５ns程度、リクエストキュー制御回路RqCTがメモリ回路MemVLへリクエストReqRD4b3m0を送信する時間は1ns程度、メモリ回路MemVLから4バイトのデータが読み出され、レスポンスキュー制御回路RsCTへレスポンスResRD4b3m0として入力されまでの時間が15ns程度、レスポンスResRD4b3mが情報処理装置CPU_CHIPへ到達するまでの時間が5ns程度である。したがって、情報処理装置CPU_CHIPが、メモリチップM0に対するリクエストReqRD4b3m0を発行してからレスポンスResRD4b3m0を受け取るまでの時間は、26ns程度となる。

【０１０８】

このように、リクエストの入力順序に関わらず、早く読み出せるデータは、読み出しが遅いデータを待つことなく、すぐに読み出すことができるため、高速化が可能となる。さらに、リクエストへIDを付加することで、確実に要求先へリクエストが転送され、また、レスポンスへIDを付加することで、リクエストの入力順序と、読み出しデータの順番が異なった場合でも、情報処理装置CPU__CHIPは転送元のメモリチップを知ることができるため、情報処理装置CPU__CHIPおよびメモリチップの直列接続によって、接続信号数を少なくしながらも、情報処理装置CPU_CHIPは所望の処理を実行することができる。

【０１０９】

本実施例ではデータ読み出しを中心に説明したが、データの書き込み動作においても同様の動作を行うことができるのは言うまでもない。また、本実施例では、メモリチップM0とM1とのデータ転送動作を説明したが、その他のメモリチップの場合についても同様のデータ転送動作を行うことは言うまでもない。

【０１１０】

＜クロック制御＞

次に、メモリモジュールMEMに関するクロック制御について説明する。メモリモジュールMEMが特に限定しないが携帯機器に利用された場合、常にメモリモジュールMEM内のメモリチップM0、M1及びM2の全てが同時に動作するわけではない。そこで、携帯機器の低電力化を図るために、本メモリモジュールMEMはデータ転送に必要な場合に、必要な周波数でクロックを発生したり、データ転送が生じない場合はクロックを停止したりできる。

【０１１１】

メモリチップM0から出力するレスポンスクロック信号RsCk0の周波数制御について説明する。先ず、メモリチップM0から出力するレスポンスクロック信号RsCk0のクロック周波数を、特に限定しないが2分の１にする場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM0のID値２とレスポンスクロック分周コマンド２を入力する。

【０１１２】

メモリチップM0はクエストキュー制御回路RqCTを介して、レスポンスクロック分周コマンド２をメモリチップM0のクロック分周回路Div2へ送信すると、レスポンスクロック信号RsCk0の周波数は2分の１となる。クロックの動作周波数を低くする際は、ノイズによる誤動作を防ぐために徐々に周波数を落とし、最後に所望の周波数で動作させることが良い。

【０１１３】

次に、メモリチップM0から出力するレスポンスクロック信号RsCk0を停止する場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM0のID値２とレスポンスクロック停止コマンドを入力する。メモリチップM0はリクエストキュー制御回路RqCTを介して、レスポンスクロック停止コマンドをメモリチップM0内のクロック

分周回路Div2へ送信すると、レスポンスクロック信号RsCk0は停止する。クロックを停止する際は、ノイズによる誤動作を防ぐために徐々に周波数を落とし、最後に停止させることが良い。

【０１１４】

次に、停止しているレスポンスクロック信号RsCk0を再度動作させる場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM0のID値２とレスポンスクロック再開コマンドを入力する。メモリチップM0はクエストキュー制御回路RqCTを介して、レスポンスクロック再開コマンドをメモリチップM0内のクロック分周回路Div2へ送信すると、停止しているレスポンスクロック信号RsCk0は再度、動作を開始する。クロックを再動作させる際は、ノイズによる誤動作を防ぐために、徐々に周波数を上げ、最後に所望の周波数で動作させることが良い。

【０１１５】

メモリチップＭ１から出力するレスポンスクロック信号RsCk1の周波数制御について説明する。先ず、メモリチップM1から出力するレスポンスクロック信号RsCk1のクロック周波数を、特に限定しないが4分の１にする場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM1のID値1とレスポンスクロック分周コマンド4を入力すると、メモリチップM0を通じて、メモリチップM1へID値1とレスポンスクロック分周コマンド4が送信される。メモリチップM1がクエストキュー制御回路RqCTを介して、レスポンスクロック分周コマンド4をメモリチップM1内のクロック分周回路Div2へ送信すると、レスポンスクロック信号RsCk1の周波数は4分の１となる。クロックの動作周波数を低くする際は、ノイズによる誤動作を防ぐために徐々に周波数を落とし、最後に所望の周波数で動作させることが良い。

【０１１６】

次に、メモリチップM1から出力するレスポンスクロック信号RsCk1を停止する場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM1のID値1とレスポンスクロック停止コマンドを入力すると、メモリチップM0を通じて、メモリチップM1へID値1とレスポンスクロック分周コマンド4が送信される。メモリチップＭ１がクエストキュー制御回路RqCTを介して、レスポンスクロック停止コマンドをメモリチップＭ１内のクロック分周回路Div2へ送信すると、レスポンスクロック信号RsCk1は停止する。クロックを停止する際は、ノイズによる誤動作を防ぐために徐々に周波数を落とし、最後に停止させることが良い。

【０１１７】

次に、停止しているレスポンスクロック信号RsCk1を再度動作させる場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM1のID値1とレスポンスクロック再開コマンドを入力すると。メモリチップM0を通じて、メモリチップM1へID値1とレスポンスクロック再開コマンドが送信される。メモリチップM1が、クエストキュー制御回路RqCTを介して、レスポンスクロック再開コマンドをメモリチップM1内のクロック分周回路Div2へ送信すると、停止しているレスポンスクロック信号RsCk1は再度動作を開始する。クロックを再動作させる際は、ノイズによる誤動作を防ぐために、徐々に周波数を上げ、最後に所望の周波数で動作させることが良い。

【０１１８】

メモリチップM2から出力するレスポンスクロック信号RsCk2の周波数制御について説明する。先ず、メモリチップM2から出力するレスポンスクロック信号RsCk2のクロック周波数を、特に限定しないが8分の１にする場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM2のID値3とレスポンスクロック分周コマンド8を入力すると、メモリチップM0及びM1を通じて、メモリチップM2へID値3とレスポンスクロック分周コマンド8が送信される。メモリチップM2が自身のクエストキュー制御回路RqCTを介して、レスポンスクロック分周コマンド8をメモリチップM2内のクロック分周回路Div2へ送信すると、レスポンスクロック信号RsCk2の周波数は8分の１となる。クロックの動作周波数を低くする際は、ノイズによる誤動作を防ぐために徐々に周波数を落とし、最後

に所望の周波数で動作させることが良い。

【０１１９】

次に、メモリチップM2から出力するレスポンスクロック信号RsCk2を停止する場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM2のID値3とレスポンスクロック停止コマンドを入力すると、メモリチップM0及びM1を通じて、メモリチップM2へID値3とレスポンスクロック停止コマンドが送信される。メモリチップM2が自身のクエストキュー制御回路RqCTを介して、レスポンスクロック停止コマンドをメモリチップM2内のクロック分周回路Div2へ送信すると、レスポンスクロック信号RsCk2は停止する。クロックを停止する際は、ノイズによる誤動作を防ぐために徐々に周波数を落とし、最後に停止させることが良い。

【０１２０】

次に、停止しているレスポンスクロック信号RsCk2を再度動作させる場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM2のID値3とレスポンスクロック再開コマンドを入力すると。メモリチップM0及びM1を通じて、メモリチップM2へID値3とレスポンスクロック再開コマンドが送信される。メモリチップM2が、クエストキュー制御回路RqCTを介して、レスポンスクロック再開コマンドをメモリチップM2のクロック分周回路Div2へ送信すると、停止しているレスポンスクロック信号RsCk2は再度、動作を開始する。クロックを再動作させる際は、ノイズによる誤動作を防ぐために、徐々に周波数を上げ、最後に所望の周波数で動作させることが良い。

【０１２１】

メモリチップM0から出力するリクエストクロック信号RsCk1の周波数制御について説明する。先ず、メモリチップM0から出力するリクエストクロック信号RqCk1のクロック周波数を、特に限定しないが2分の１にする場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM0のID値２とリクエストクロック分周コマンド２を入力する。メモリチップM0が、リクエストキュー制御回路RqCTを介して、リクエストクロック分周コマンド２をメモリチップM0のクロック分周回路Div1へ送信すると、このクロック分周回路Div1はリクエストクロック信号RqCk0のクロック周波数の2分の1の周波数を持つクロックを発生させ、リクエストクロック信号RqCk1から出力する。リクエストクロック信号RqCk1は、メモリチップM1へ入力し、メモリチップM1のクロックドライバDrv2およびクロック分周回路Div２を介してレスポンスクロック信号RsCk1として出力する。クロックの動作周波数を低くする際は、ノイズによる誤動作を防ぐために徐々に周波数を落とし、最後に所望の周波数で動作させることが良い。

【０１２２】

次に、メモリチップM0から出力するリクエストクロック信号RqCk1を停止する場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM0のID値２とリクエストクロック停止コマンドを入力する。メモリチップM0が、リクエストキュー制御回路RqCTを介して、リクエストクロック停止コマンドをメモリチップM0のクロック分周回路Div1へ送信すると、このクロック分周回路Div1はリクエストクロック信号RqCk1を停止する。リクエストクロック信号RqCk1は、メモリチップM1へ入力し、メモリチップM1のクロックドライバDrv2およびクロック分周回路Div２を介してレスポンスクロック信号RsCk1として出力するためレスポンスクロック信号RsCk1も停止する。クロックを停止する際は、ノイズによる誤動作を防ぐために徐々に周波数を落とし、最後に停止させることが良い。

【０１２３】

次に、停止しているリクエストクロック信号RsCk1を再度動作させる場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM0のID値２とリクエストクロック再開コマンドを入力する。メモリチップM0が、リクエストキュー制御回路RqCTを介して、リクエストクロック再開コマンドをメモリチップM0のクロック分周回路Div1へ送信すると、このクロック分周回路Div1は停止しているリクエストクロック信号RqCk1を再度、動作させる。リクエストクロック信号RqCk1は、メモリチップM1へ入力し、メ

モリチップM1のクロックドライバDrv2およびクロック分周回路Div2を介してレスポンスクロック信号RsCk1として出力するため、レスポンスクロック信号RsCk1も再度、動作する。クロックを再動作させる際は、ノイズによる誤動作を防ぐために、徐々に周波数を上げ、最後に所望の周波数で動作させることが良い。

【０１２４】

メモリチップM1から出力するリクエストクロック信号RsCk2の周波数制御について説明する。先ず、メモリチップM1から出力するリクエストクロック信号RqCk2のクロック周波数を、特に限定しないが4分の１にする場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM1のID値1とリクエストクロック分周コマンド4を入力すると、メモリチップＭ０を通じてID値1とリクエストクロック分周コマンド4がメモリチップM1へ送信される。メモリチップM1が、リクエストキュー制御回路RqCTを介して、リクエストクロック分周コマンド4を自身のクロック分周回路Div1へ送信すると、このクロック分周回路Div1はリクエストクロック信号RqCk0のクロック周波数の4分の1の周波数を持つクロックを発生させ、リクエストクロック信号RqCk2から出力する。リクエストクロック信号RqCk2は、メモリチップM2へ入力し、メモリチップM2のクロックドライバDrv2およびクロック分周回路Div２を介してレスポンスクロック信号RsCk2として出力する。クロックの動作周波数を低くする際は、ノイズによる誤動作を防ぐために徐々に周波数を落とし、最後に所望の周波数で動作させることが良い。

【０１２５】

次に、メモリチップM1から出力するリクエストクロック信号RqCk2を停止する場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM1のID値1とリクエストクロック停止コマンドを入力するとメモリチップM0を通じてID値1とリクエストクロック停止コマンドがメモリチップM1へ送信される。メモリチップM1は、自身のリクエストキュー制御回路RqCTを介して、リクエストクロック停止コマンドを自身のクロック分周回路Div1へ送信すると、このクロック分周回路Div1はリクエストクロック信号RqCk2を停止する。リクエストクロック信号RqCk2は、メモリチップM2へ入力し、メモリチップM2のクロックドライバDrv2およびクロック分周回路Div２を介してレスポンスクロック信号RsCk2として出力するためレスポンスクロック信号RsCk2も停止する。

【０１２６】

クロックを停止する際は、ノイズによる誤動作を防ぐために徐々に周波数を落とし、最後に停止させることが良い。

【０１２７】

次に、停止しているリクエストクロック信号RsCk2を再度動作させる場合について説明する。情報処理装置CPU_CHIPが、リクエスト信号RqMux0よりメモリチップM1のID値1とリクエストクロック再開コマンドを入力すると、メモリチップM0を通じてID値1とリクエストクロック再開コマンドがメモリチップM1へ送信される。メモリチップM1が、自身のリクエストキュー制御回路RqCTを介して、リクエストクロック再開コマンドを自身のクロック分周回路Div1へ送信すると、このクロック分周回路Div1は停止しているリクエストクロック信号RqCk2を再度、動作させる。リクエストクロック信号RqCk2は、メモリチップM2へ入力し、メモリチップM2のクロックドライバDrv2およびクロック分周回路Div2を介してレスポンスクロック信号RsCk1として出力するため、レスポンスクロック信号RsCk2も再度、動作する。クロックを再動作させる際は、ノイズによる誤動作を防ぐために、徐々に周波数を上げ、最後に所望の周波数で動作させることが良い。

【０１２８】

＜実施例１の効果＞

以下、上述の実施の形態について、構成とその効果についてまとめる。

（１）電源投入直後に、直列接続の確認動作を行うことで、確実にメモリ同士が接続されていることが確認できる。さらに、ブートデバイスおよび、最端のメモリチップを明示し、自動的に各メモリへのID付けが行われることで、容易に、必要な分だけメモリチップを接続し、メモリ容量を拡張することができる。

（２）リクエストへIDを付加することで、情報処理装置CPU_CHIPから各メモリチップM0、M1およびM2へリクエストが確実に転送される。また、情報処理装置CPU_CHIPへのレスポンスへIDを付加することで、各メモリから正しく正しくデータ転送が行えたことを確認でき、情報処理装置CPU_CHIPおよびメモリチップM0、M1、M2の直列接続によって、接続信号数を減少させながらも、情報処理装置CPU_CHIPは所望の処理を実行することができる。
（３）リクエストインターフェース回路ReqIFとレスポンスインターフェース回路は独立に動作可能なため、データの読み出し動作と書き込み動作を同時に実行でき、データ転送性能を向上させることができる。
（４）リクエストの入力順序に関わらず、早く読み出せるデータは、読み出しが遅いデータを待つことなく、すぐに読み出すことができるため、高速化が可能となる。さらに、リクエストへIDを付加することで、確実に要求先へリクエストが転送され、また、レスポンスへIDを付加することで、リクエストの入力順序と、読み出しデータの順番が異なった場合でも、情報処理装置CPU__CHIPは転送元のメモリチップを知ることができる。
（５）各メモリチップM0、M1およびM2のクロックを必要に応じて、低速動作させたり、停止させたり、復帰させたりできるため、低電力化を図ることができる。
（６）メモリチップM2からの読み出し時は、エラー検出と訂正を行い、書きこみ時は、書きこみが正しく行われなかった不良アドレスに対して代替処理を行うため、信頼性を保つことができる。

【０１２９】

また、本実施例では、メモリモジュールMEM0には1つの揮発性メモリ、１つのNOR型フラッシュメモリ、１つのNAND型フラッシュメモリがが含まれているを例について説明しているが、メモリモジュールMEM0に複数個の揮発性メモリ及び複数個のNOR型フラッシュメモリ及びNAND型フラッシュメモリが含まれる場合であっても本発明を実現できるのは言うまでもない。

【０１３０】

＜メモリマップの説明＞

図２は、情報処理装置CPU_CHIPが管理するメモリモジュールMEM0に対するメモリマップの一例を示したものである。本実施の形態例では、特に限定されないが、メモリチップM0の記憶領域は１Gbit、メモリチップM1の記録領域は1Git、メモリチップM2の記憶領域は4Gbit+128Mbit(128Mbitは代替領域)であるメモリモジュールを例に代表的なメモリマップを説明する。

【０１３１】

特に限定しないが、メモリチップM0は揮発性メモリでダイナミックランダムアクセスメモリセルを利用したダイナミックランダムアクセスメモリであり、読み出し時間が15ns程度である。特に限定しないが、メモリチップＭ１は不揮発性メモリでNOR型フラッシュメモリセルを利用したNOR型フラッシュメモリであり、読み出し時間が80ns程度である。特に限定しないが、メモリチップM2は不揮発性メモリでNAND型フラッシュメモリセルを利用したNAND型フラッシュメモリであり、読み出し時間が25usec程度である。特に限定しないが、メモリチップＭ１は、ブートデバイスID格納領域BotID-AREA、最終端デバイスID格納領域EndID-AREA、初期プログラム領域InitPR-AREA、プログラム格納領域OSAP-AREAに分かれている。

【０１３２】

ブートデバイスID格納領域BotID-AREAには、ブートデバイスのID情報が格納される。最終端デバイスID格納領域EndID-AREAには、直列接続されているメモリモジュールMEM0に関する最終端メモリデバイスID情報が格納される。初期プログラム領域InitPR-AREAには、特に限定しないが、ブートプログラムが格納される。プログラム格納領域OSAP-AREAには、特に限定しないが、オペレイティングシステムやアプリケーションプログラムなどが格納される。特に限定しないが、メモリチップM0はコピー領域COPY-AREA、ワーク領域WORK-AREAに分かれている。ワーク領域WORK-AREAはプログラム実行時のワークメモリとして、コピー領域COPY-AREAはメモリチップM1及びM2からのプログラムやデータをコピーするた

めのメモリとして利用される。特に限定しないが、メモリチップＭ２は、データ領域DATA-AREA、代替領域REP-AREAに分かれている。データ領域DATA-AREAには、特に限定しないが、音楽データ、音声データ、動画データ、静止画データなどのデータが格納される。

【０１３３】

また、FLASHは書き換えを繰り返すことによって、信頼性が低下し、書き込み時に書いたデータが、読み出し時には異なるデータとなったり、書き換え時にデータが書き込まれなかったりすることが稀にある。代替領域REP-AREAは、このように不良となったデータを新たな領域へ置き換えるために設けられている。代替領域REP-AREAの大きさは、特に限定しないがメモリチップM2が保証する信頼性が確保できるように決めると良い。

【０１３４】

＜電源投入直後の動作＞

電源投入直後のメモリチップM1から情報処理装置CPU_CHIPへのデータ転送について説明する。電源投入後、情報処理装置CPU_CHIPは自身の持つブートデバイスIDレジスタBotIDを１へ設定する。メモリチップM1はブートデバイスID格納領域BotID-AREAからブートデバイスのID情報１を読み出し、自身のIDレジスタへ１を設定する。これにより、ブートデバイスがメモリチップM1に確定する。

【０１３５】

次に、情報処理装置CPU_CHIPはブートデバイスであるメモリチップM1に格納されているブートプログラム及び最終端メモリデバイスID情報を読み出すため、メモリチップM1のID番号１と読み出し命令をメモリモジュールMEM0へ送信する。メモリモジュールMEM0は、ID番号１と読み出し命令に従って、メモリチップM1の初期プログラム領域InitPR-AREAからブートプログラムを読み出し、最終端デバイスID格納領域EndID-AREAから最終端メモリデバイスID情報を読み出し、情報処理装置CPU_CHIPへ送信する。このように、電源投入直後に、ブートデバイスのIDを初期設定することで、メモリチップの直列接続によって実現されるメモリモジュールMEM0内のブートデバイスを特定することができ、情報処理装置CPU_CHIPとメモリモジュールMEM０間の接続信号数を大幅に少なくした上で、情報処理装置CPU_CHIPは、すばやく確実にブートデバイスよりブートプログラムおよび最終端メモリデバイスIDを読み出し、情報処理装置CPU_CHIP及びメモリモジュールMEM0を立ち上げることができる。

【０１３６】

＜データコピー動作の説明＞

メモリチップM0のデータ読み出し時間は、メモリチップM2の読み出し時間と比較し、大幅に短い。そこで、前もって必要な画像データをメモリチップM2からメモリチップM0へ転送すれば、情報処理装置CPU_CHIPにて高速に画像処理を行うことができる。特に限定しないが、メモリチップM0、M1、M2のそれぞれのIDレジスタ値が２、１及び３に設定された場合の、メモリチップＭ２からのメモリチップM0のへのデータ転送について説明する。

【０１３７】

情報処理装置CPU_CHIPはメモリチップＭ２のデータ領域DATA-AREAからデータを読み出すため、メモリチップM2のID番号３と1ページ(512Byteのデータ+16ByteのECCコード)データ読み出し命令をメモリモジュールMEM0へ送信する。メモリモジュールMEM0は、ID番号３と1ページデータ読み出し命令に従って、メモリチップM2のデータ領域DATA-AREAから1ページ分のデータを読み出し、ID番号３を付加し、情報処理装置CPU_CHIPへ送信する。

【０１３８】

情報処理装置CPU_CHIPでは、メモリチップM2から送信された1ページ分のデータに対しエラー検出を行う。エラーがなければ、１ページ分のデータをメモリチップM0のコピー領域COPY-AREAへデータを転送するため、情報処理装置CPU_CHIPはメモリチップM0のID番号2と1ページデータ読み出し命令をメモリモジュールMEM0へ送信する。エラーがあれば修正を行った後、１ページ分のデータをメモリチップM0のコピー領域COPY-AREAへデータを転送するため、情報処理装置CPU_CHIPはメモリチップM0のID番号2と1ページデータ読み出し命令をメモリモジュールMEM0へ送信する。メモリモジュールMEM0は、ID番号２と1ページ

データ読み出し命令に従って、メモリチップＭＯのコピー領域COPY-AREAデータ領域へ1ページ分のデータを書き込む。

【０１３９】

次に、情報処理装置CPU_CHIPからメモリチップM0へ高速に画像データが書き込まれ、必要に応じてメモリチップM2へこの画像データを保存する際の、メモリチップM0からのメモリチップM2へのデータ転送について説明する。情報処理装置CPU_CHIPはメモリチップM0のコピー領域COPY-AREAからデータを読み出すため、メモリチップM0のID番号2と1ページ(512Byte)データ読み出し命令をメモリモジュールMEM0へ送信する。メモリモジュールMEM0は、ID番号0と1ページデータ読み出し命令に従って、メモリチップM0のコピー領域COPY-AREAから1ページ分のデータを読み出し、ID番号2を付加し、情報処理装置CPU_CHIPへ送信する。情報処理装置CPU_CHIPは、メモリチップM0から送信された1ページ分のデータをメモリチップM2のデータ領域DATA-AREAへデータを転送するため、メモリチップM2のID番号2と1ページデータ書き込み命令をメモリモジュールMEM0へ送信する。

【０１４０】

メモリモジュールMEM0が、メモリチップM0及びM1を通じてメモリチップM2へID番号２と1ページデータ書込み命を送信すると、メモリチップM2は自身のデータ領域DATA-AREAへ1ページ分のデータを書き込む。メモリチップM2はデータの書き込みが成功したかどうかをチェックし、成功すれば書き込み処理を終了する。書き込みが失敗した時には、メモリチップM2は、ID番号２と書込エラー情報を送信し、メモリチップM1及びメモリチップM0を介して、情報処理装置CPU_CHIPへ書込みエラーを通達する。情報処理装置CPU_CHIPは、ID番号２と書込エラー情報を受け取ると、メモリチップM2にあらかじめ用意されている代替領域REP-AREAの新たなアドレスに対して書き込みを行うために、メモリチップM2のID番号2と1ページデータ書き込み命令をメモリモジュールMEM0へ送信する。メモリモジュールMEM0がメモリチップM0及びM1を通じてID番号２と1ページデータ書込み命令をメモリチップM2へ送信すると、メモリチップM2は自身の代替領域REP-AREAへ1ページ分のデータを書き込む。また、情報処理装置CPU_CHIPは、代替え処理を行った際は、不良アドレスと、不良アドレスに対して、どのアドレスに代替え処理を行ったかというアドレス情報を保持し管理する。

【０１４１】

以上説明したように、メモリチップM2の一部のデータをコピーできる領域をメモリチップ内に確保し、あらかじめメモリチップM2からメモリチップM0へデータを転送しておくことで、メモリチップM0と同等の速度でメモリチップM2のデータを読み出すことができ、情報処理装置CPU_CHIPでの高速処理が可能となる。また、メモリチップM2へデータを書く際は、いったんデータをメモリチップM0へ書き込み、必要に応じてメモリチップM2へ書き戻すことができるため、データの書き込みも高速化することができる。さらに、メモリチップM2からの読み出し時は、エラー検出と訂正を行い、書きこみ時は、書きこみが正しく行われなかった不良アドレスに対して代替処理を行うため、高信頼性を保つことができる。

【０１４２】

＜電源投入時の初期シーケンス＞

図３は、情報処理装置CPU_CHIPとメモリモジュールMEM0とから構成される情報システム装置の電源投入時の初期シーケンスを示す。T1の期間（PwON）で情報処理装置CPU_CHIPと、メモリモジュールMEM0内のメモリチップM0、M1及びM2へ電源投入を行い、T2の期間（RESET）でリセットを行う。リセットの方法は特に限定しないが、それぞれの内蔵回路で自動的にリセットを行う方法でも、あるいは、外部にリセット端子を持ち、このリセット信号によってリセット動作を行うこととしても良い。T2のリセット期間には、情報処理装置CPU_CHIPはブートデバイスIDレジスタBotIDを１へ、最端デバイスIDレジスタEndIDを０へ設定する。メモリチップM0、M1、M2は各々が持っているIDレジスタの値を０へ、ID有効ビットをLowへ初期設定する。また、メモリチップM0、M1、M2は、おのおのが持っているレスポンスキューの優先順位、優先順位を変えるレスポンス実行回数値の初期設定を行う。さらに、メモリチップM0、M1、M2は、各々の動作クロック周波数の分周比の初期設定を行

う。

【０１４３】

リセットが解除されたT3の期間（BootIDSet）でブートデバイスがブートデバイスIDをIDレジスタへセットする。メモリチップM0、M1およびM2は、ブートデバイス認識信号Bsigが電源に接続されているので、自分自身がブートデバイスではないことを認識し、それぞれのIDレジスタの値を０のままにする。メモリチップM1のブートデバイス認識信号Bsigが接地gndされているので、自分自身がブートデバイスであることを認識し、自らのメモリ回路MemNV1が保持しているブートデバイスID値１を読み出し、IDレジスタへ設定し、ID有効ビットをHighにする。T3の期間が終了した後のT4の期間（LinkEn）では、各メモリチップM0,M1およびM2の信号の接続確認を行う。メモリチップM2は、直列接続しているメモリチップの最も終端のメモリチップであることを認識し、リクエストイネーブル信号RqEn2をHighにする。

【０１４４】

次に、メモリチップM1はリクエストイネーブル信号RqEn2がHighになったことを確認し、レスポンスイネーブル信号RsEn2及びリクエストイネーブル信号RqEn1をHighにする。次に、メモリチップM0はリクエストイネーブル信号RqEn1がHighになったことを確認し、レスポンスイネーブル信号RsEn1及びリクエストイネーブル信号RqEn0をHighにする。最後に、情報処理装置CPU_CHIPは、リクエストイネーブル信号RqEn0がHighになったことを確認し、各メモリチップの信号接続が確認されたことを知り、レスポンスイネーブル信号RsEn0をHighにする。T4の期間が終了した後のT5の期間（BootRD）では、情報処理装置CPU_CHIPがメモリチップM1よりブートデータを読み出す。

【０１４５】

情報処理装置CPU_CHIPは、リクエスト信号RqMux0を通じて、メモリチップM1のID値１、読みだし命令、アドレスを多重化したリクエストNRDm1をクロック信号RqCK0に同期させ、メモリチップM0へ転送する。メモリチップM0のID有効ビットがLowのため、メモリチップM0はリクエスト信号RqMux1yよりリクエストReqNRDm1をクロック信号RqCK1に同期させ、をメモリチップM1へ転送する。メモリチップＭ１は、メモリチップM0からのリクエストリクエストReqNRDm1を、自身のリクエストキュー制御回路RqCTへ格納する。メモリチップM1のID有効ビットがHighのため、リクエストReqNRDm1に含まれるID値１と自身のIDレジスタの値１を比較する。比較結果は一致しているため、リクエストReqNRDm1をメモリ回路MemNV1へ転送する。リクエストReqNRDm1によってメモリ回路MemNV1からブートデータと最終端デバイスID番号３が読み出され、IDレジスタ値１とともに、レスポンスResNRDm1としてレスポンスキュー制御回路RsCTへ転送される。メモリチップM1のレスポンスキュー制御回路RsCTはレスポンス信号RqMux1より、レスポンスResNRDm1をメモリチップM0へ転送する。最後に、メモリチップM0のレスポンスキュー制御回路RsCTはレスポンス信号RqMux0よりレスポンスResNRDm1を情報処理装置CPU_CHIPへ転送する。情報処理装置CPU_CHIPは、レスポンスResNRDm1を受け取り、最終端デバイスID値３をメモリ制御回路CON内の最終端デバイスIDレジスタENDIDへ保存する。次に、受け取ったブートプログラムによって自らを立ち上げる。Ｔ５の期間が終了した後のＴ６の期間（InitID）では、ブートコードに従い、情報処理装置CPU_CHIPが各メモリチップへID番号を設定する。

【０１４６】

情報処理装置CPU_CHIPは、先ず、リクエスト信号RqMux0を通じて、ID値２とID設定命令をメモリチップM0へ転送する。メモリチップM0では、ID有効ビットがLowにより、まだID番号付けが行われていないため、ID番号２とID設定命令によってIDレジスタへID番号２を設定し、ID有効ビットをHighにする。ID有効ビットがHighとなることで、ID番号付けが完了したことを示す。メモリチップM0は、ID番号付けが完了したため、ID値２とID番号付け完了情報をレスポンス信号RsMux0を通じて情報処理装置CPU_CHIPへ知らせる。

【０１４７】

情報処理装置CPU_CHIPはメモリチップM0のID番号付けが完了したことを知ると、次にリクエスト信号RqMux0よりID番号3とID設定命令をメモリチップM0へ転送する。メモリチッ

プＭ０は自身のID番号２とID番号3とを比較し、不一致のため、ID番号3とID設定命令をメモリチップＭ１へ転送する。メモリチップM1dではすでにID番号付けがなされているいるため、自身のID番号1とID番号3とを比較し、不一致のため、ID番号3とID設定命令をリクエスト信号RqMux２より、メモリチップM2へ転送する。

【０１４８】

メモリチップＭ２では、まだID番号付けが行われていないため、メモリチップＭ２は、ID番号３とID設定命令によってIDレジスタへID番号３を設定し、ID有効ビットをHighにする。ID有効ビットがHighとなることで、ID番号付けが完了したことを示す。メモリチップM2は、ID番号付けが完了したため、ID値３とID番号付け完了情報を、メモリチップM1及びメモリチップM0を介して、情報処理装置CPU_CHIPへ送信する。情報処理装置CPU_CHIPは、送信されたID値３とメモリ制御回路CON内の最終端デバイスIDレジスタEndIDへ設定されている最終端デバイスID値３とを比較する。双方の値が一致することで、最終端のメモリチップまでID番号付けが行われたことを確認する。

【０１４９】

Ｔ６の期間が終了した後のＴ７の期間（Idle）以降は、メモリモジュールMEM０はアイドル状態となり、情報処理装置CPU_CHPからのリクエストを待つ状態となる。

【０１５０】

＜メモリチップM0の説明＞

図４は、メモリチップM0の構成図の一例である。図５はメモリチップM0へのリクエストが発生した際の、動作の一例を示すフローチャートである。図６はメモリチップM0のメモリ回路MemVLからのレスポンスが発生した際の、動作の一例を示すフローチャートである。図７はメモリチップＭ１からメモリチップM0へレスポンスが発生した際の、動作の一例を示すフローチャートである。以下で各回路ブロックの動作を説明する。

【０１５１】

メモリチップM0は、リクエストインターフェース回路ReqIFと、レスポンスインターフェース回路ResIFと、初期化回路INIT、メモリ回路MemVLから構成されている。リクエストインターフェース回路ReqIFはリクエストクロック制御回路RqCkCおよび、リクエストキュー制御回路RqCTから構成される。リクエストクロック制御回路RqCkCはクロックドライバDrv1およびクロック分周回路Div1から構成される。リクエストキュー制御回路RqCTはリクエストキュー回路RqQI、リクエストキュー回路RqQXI、リクエストキュー回路RqQXO、IDレジスタ回路dstID、ID比較回路CPQから構成される。特に限定しないが、リクエストキュー回路RqQIは2つのリクエストキューから構成され、リクエストキュー回路RqQXIは1つのリクエストキューから構成され、リクエストキュー回路RqQXOは2つのリクエストキューから構成される。レスポンスインターフェース回路ResIFはレスポンスクロック制御回路RsCkCおよび、レスポンスキュー制御回路RsCTから構成される。レスポンスクロック制御回路RsCkCはクロックドライバDrv2およびクロック分周回路Div2から構成される。レスポンスキュー制御回路RsCTは、レスポンスキュー回路RsQo、レスポンスキュー回路RsQp、ステータスレジスタ回路STReg、レスポンススケジュール回路SCHから構成される。特に限定しないが、レスポンスキュー回路RsQoは4つのレスポンスキューから構成され、スポンスキュー回路RsQpは4つのレスポンスキューから構成される。

【０１５２】

メモリ回路MemVLは、特に限定しないが、揮発性メモリであり、ダイナミックランダムアクセスメモリセルを利用したダイナミックランダムアクセスメモリである。初期化回路INITは、メモリチップM0への電源供給開始時にメモリチップM0の初期化を行う。リクエストクロック制御回路RqCkCは、クロック信号RqCk0から入力したクロックを、内部クロックck1を通じて、リクエストキュー制御回路RqCT及びレスポンスクロック制御回路RsCkCへ伝える。また、リクエストクロック制御回路RqCkCは、リクエストクロック信号RqCk0から入力されたクロックをクロックドライバDrv1及びクロック分周回路Div1を介して、クロック信号RqCk1を通じて出力する。また、リクエストクロック制御回路RqCkCはリクエスト信号RqMux0を通じて入力した命令に従い、クロック信号ck2およびリクエストクロックRqCk1の

クロック周波数を低下させたり、クロックを停止させたり、クロックを再動作させることができる。

【０１５３】

レスポンスクロック制御回路RsCkCは、内部クロック信号ck1から入力したクロックを、内部クロック信号ck3を通じて、レスポンスキュー制御回路RsCTへ出力する。また、レスポンスクロック制御回路RsCkCは内部クロック信号ck1からから入力したクロックを、クロック分周回路Div2を介してクロック信号RsCk0から出力する。また、レスポンスクロック制御回路RsCkCは、クロック信号RsCK1から入力したクロックを、クロックドライバDiv2を介して、クロック信号ck4よりレスポンスキュー制御回路RsCTへ出力する。さらに、レスポンスクロック制御回路RsCkCはリクエスト信号RqMux0を通じて入力した命令に従い、レスポンスクロックRsCk0のクロック周波数を低下させたり、また、クロックを停止させたり、さらに、クロックを再動作させることができる。

【０１５４】

リクエストキュー回路RqQIは、リクエスト信号RqMux0を通じて、ID値、命令、アドレス及び書き込みデータが多重化されメモリチップM0へ入力したリクエストを格納する。IDレジスタ回路dstIDは、メモリチップM0のID値およびID有効信号を格納する。ID比較回路CPQは、リクエストキュー回路RqQIに格納されているID値と、IDレジスタ回路dstIDに格納されているID値を比較する。

【０１５５】

リクエストキュー回路RqQXI及びリクエストキュー回路RqQXOは、リクエストキュー回路RqQIから転送されたリクエストを格納する。レスポンスキュー回路RsQoは、メモリチップM0のメモリ回路MemVLから読み出されたデータ及びIDレジスタ回路dstIDから読み出されたID値を格納する。レスポンスキュー回路RsQpは、レスポンス信号RsMux1を通じて、入力されるID値、読み出しデータおよびエラー情報およびステータス情報を格納する。

【０１５６】

ステータスレジスタ回路STRRegは、特に限定しないがレスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへレスポンスが格納されていることを示す未処理レスポンス情報などが格納される。レスポンススケジュール回路SCHは、ンスキュー回路RsQoへ格納されているレスポンスと、レスポンスキュー回路RsQpへ格納されているレスポンスとのレスポンス優先順位を決め、優先順位の高いレスポンスを、レスポンス信号RsMux0から出力するための調停を行う。レスポンス優先順位は、レスポンスキュー回路RsQoから出力されたレスポンスの回数と、レスポンスキュー回路RsQpから出力されたレスポンスの回数によってレスポンススケジュール回路SCHが動的に変化させる。

【０１５７】

次に、本メモリチップM0の動作を説明する。まず、電源投入時の動作について説明する。メモリチップM0へ電源が投入されると初期化回路INITはメモリチップM0の初期化を行う。先ず、IDレジスタ回路dstIDの持つIDレジスタの値を０へ、ID有効ビットをLowへ初期設定する。次にレスポンススケジュール回路SCHが持つレスポンスキュー回路RsQoに入力するレスポンスの優先順位を１へ、レスポンスキュー回路RsQpに入力するメモリチップM1からのレスポンスの優先順位を2へ、メモリチップM2からのレスポンスの優先順位を3へ設定する。初期化回路INITによる初期設定が終了すると、メモリチップM0は、情報処理装置CPU_CHIPとメモリチップM1との間で通信できることを確認する通信確認動作を行う。メモリチップM0はリクエストイネーブル信号RqEn1がHighになったことを確認し、レスポンスイネーブル信号RsEn1及びリクエストイネーブル信号RqEn0をHighにする。

【０１５８】

次に、情報処理装置CPU_CHIPは、リクエストイネーブル信号RqEn0がHighになったことを確認し、各メモリチップの信号接続が確認されたことを知り、レスポンスイネーブル信号RsEn0をHighにする。通信確認動作が終了すると、情報処理装置CPU_CHIPよりリクエスト信号RqMux0を通じて、ID番号２とID設定命令がメモリチップM0へ転送される。。メモリチップM0では、ID有効ビットがLowのため、まだID番号付けが行われていないと判断し、I

Dレジスタへ ID番号２を、ID有効ビットをHighに設定し、ID番号付けを完了する。次に、メモリチップM0はレスポンス信号RsMux0を通じて、メモリチップM0のID値２およびID番号付け完了情報を出力し、情報処理装置CPU_CHIPへ、メモリチップM0のID番号付けが完了したことを通達する。

【０１５９】

次に、電源投入直後の動作が終了した後に、情報処理装置CPU_CHIPからリクエストがメモリチップM0へ生じた場合の動作を説明する。メモリチップM0のリクエストキュー回路RqQIは、特に限定しないが２つのリクエストキューRqQI-0及びRqQI-1から構成されている。また、メモリチップM0は、リクエストキューRqQI-0及びRqQI-1へリクエストがエントリされていないため、リクエストイネーブル信号RqEn0をHighにし、リクエストが受け付け可能であることを情報処理装置CPU_CHIPへ知らせる。メモリチップM0のレスポンスキュー回路RqQoは、特に限定しないが2つのレスポンスキューRqQo-0及びRqQo-1から構成されている。メモリチップM0のレスポンスキュー回路RqQpは、特に限定しないが2つのレスポンスキューRqQp-0及びRqQp-1から構成されている。情報処理装置CPU_CHIPは、レスポンスイネーブル信号RsEn0をHighにし、レスポンスが受け付け可能であることをメモリチップM0へ知らせる。情報処理装置CPU_CHIPは、リクエスト信号RqMux0を通じて、ID値２、バンクアクティブ命令BA、バンクアドレスBK1、ロウアドレスRowを多重化したリクエストReqBAb0m0をクロック信号RqCk0に同期させ、メモリチップM0へ転送する（図５：Step1）。

【０１６０】

次に、リクエスト信号RqMux0を通じて、ID値２、32バイトデータ読み出し命令RD4、バンクアドレスBK０、カラムアドレスCol255を多重化したリクエストReqRD32b0m0をクロック信号RqCK0に同期させ、メモリチップM0へ転送する（図5：Step1）。リクエストイネーブル信号RqEn0がLowであれば(図5：Step2)、情報処理装置CPU_CHIPからのリクエストはメモリチップM0のリクエストキュー回路RqQIへ格納されない。リクエストイネーブル信号RqEn0がHighであれば（図５：Step2）、メモリチップM0へ、情報処理装置CPU_CHIPからのリクエストReqBAb0m0とリクエストReqRD32b0m0は順に、メモリチップM0の、リクエストキュー回路RqQIのリクエストキューRqQI-0およびRqQI-1へ格納される(図５：Step3)。これで、リクエストキュー回路RqQIの全リクエストキューはエントリされ、情報処理装置CPU_CHIPからの新たなリクエストを受け付け不可能なため、リクエストイネーブル信号RqEn0をLowにする。リクエストイネーブル信号RqEn0がLowになったことで、情報処理装置CPU_CHIPは、メモリチップM0がリクエストを受け付けられなくなったことを知ることができる。

【０１６１】

その後、ID比較回路CPQは、リクエストキューRqQI-0へエントリされたリクエストReqBAb0m0に含まれるID値２と、IDレジスタ回路dstIDに保持されているID値２を比較する（図５：Step4）。比較結果が一致したため、リクエストReqBAb0m0は、リクエストキュー回路RqQXIへ転送される（図５：Step5）。比較結果が不一致の場合は、リクエストReqBAb0m0は、リクエストキュー回路RqQXOへ転送され、メモリチップM1へ転送される（図５：Step12）。

【０１６２】

次に、リクエストキュー回路RqQXIは格納しているレスポンスが読み出し命令を含むかどうかチェックする（図５：Step6）。読み出し命令を含んでいる場合は、リクエストキュー回路RqQXIは、レスポンスキュー回路RsQoのレスポンスキューRqQp-0及びRqQp-1に空きがあるかをチェックする（図５：Step7）。リクエストReqBAb0m0は読み出し命令を含んでいないため、リクエストキュー回路RqQXIは格納しているリクエストReqBAb0m0をメモリ回路MemVLへ転送する（図５：Step10）。メモリ回路MemVLはリクエストReqBAb0m0に従って動作する（図５：Step11）。具体的には、メモリ回路MemVLはリクエストReqBAb0m0に含まれるバンクアクティブ命令BA、バンクアドレスBK0、ロウアドレスRow63によって、バンク０内のロウ６３に接続されている1kByte分のメモリセルを活性化し、バンク０内のセンスアンプへ転送する（図５：Step11）。

【０１６３】

リクエストReqBAb0m0が処理されたことによって、リクエストキューRqQI-0がひとつ分空いたため、メモリチップM0は、リクエストイネーブル信号RqEn0をHighにし、新たなリクエストを受け付け可能であることを情報処理装置CPU_CHIPへ知らせる。情報処理装置CPU_CHIPは、メモリチップM0のリクエストイネーブル信号RqEn0がHighになったことを確認し、新たなリクエストとしてリクエスト信号RqMux0を通じて、ID値２、32バイト書き込み命令WT、バンクアドレスBK０、カラムアドレスCol127、32バイト分の書き込みデータを多重化したリクエストReqWT23b0m0をクロック信号RqCK0に同期させ、メモリチップM0へ転送する(図５：Step1)。

【０１６４】

リクエストイネーブル信号RqEn0をチェックし（図５：Step2）、リクエストイネーブル信号RqEn0がHighのため、メモリチップM0は、情報処理装置CPU_CHIPからのリクエストReqWT23b0m0を自身のリクエストキュー制御回路RqCT内のリクエストキューRqQI-0へ格納する（図５：Step3）。

【０１６５】

メモリチップM0は、新たなリクエストReqWT23b0m0を、自身のリクエストキュー回路RqQI内のリクエストキューRqQI-0へ格納すること（図5: Step3）とは独立して並行に、すでにリクエストキューRqQI-1に格納されているリクエストReqRD32b0m0に対する処理を行うことができる(図５：Step4以降)。

【０１６６】

次に、すでにリクエストキューRqQI-1に格納されているリクエストReqRD32b0m0についての動作を説明するID比較回路CPQは、リクエストキューRqQI-1へエントリされたリクエストReqRD32b0m0に含まれるID値２と、IDレジスタ回路dstIDに保持されているID値２を比較する（図５：Step4）。比較結果が一致したため、リクエストReq RD32b0m0は、リクエストキュー回路RqQXIへ転送される（図５：Step5）。比較結果が不一致の場合は、リクエストReqRD32b0m0は、リクエストキュー回路RqQXOへ転送され、メモリチップM1へ転送される（図５：Step12）。次に、リクエストキュー回路RqQXIは格納しているレスポンスが読み出し命令を含むかどうかチェックする（図５：Step6）。リクエストReqRD32b0m0読み出し命令を含んでいるため、リクエストキュー回路RqQXIは、レスポンスキュー回路RsQoのレスポンスキューRqQp-0及びRqQp-1に空きがあるかをチェックする（図５：Step7）。レスポンスキュー回路RsQoのレスポンスキューRqQp-0及びRqQp-1に空きがなければ、空きができるまで、リクエストキュー回路RqQXIは、リクエストReq RD32b0m0の転送を中断する。レスポンスキュー回路RsQoのレスポンスキューRqQp-0及びRqQp-1に空きがあれば、リクエストキュー回路RqQXIは格納しているリクエストReq RD32b0m0をメモリ回路MemVLへ転送する（図５：Step8）。メモリ回路MemVLはリクエストReq RD32b0m0に従って動作する（図５：Step9）。具体的には、メモリ回路MemVLはリクエストReqRD32b0m0に含まれる、ID値２、32バイトデータ読み出し命令RD、バンクアドレスBK０、カラムアドレスCol255によって、バンク０のセンスアンプに保持されているデータのなかで、カラムアドレス255を開始アドレスとした32バイト分のデータが読み出だし（図５：Step９）、IDレジスタ値２を含めて、レスポンスキュー制御回路RsCT内のレスポンスキューRsQoのレスポンスキューRsQo-0へレスポンスResRD32b0m0としてエントリされる（図６：Step13）。

【０１６７】

レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへレスポンスがエントリされると、レスポンススケジュール回路SCHは、レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへエントリされているレスポンス数を、ステータスレジスタSTRegへ保存する（図６：Step14）。さらに、レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへエントリされているレスポンスに対するレスポンス優先順位を決定する（図６：Step15）。次に、レスポンスイネーブル信号RsEn0をチェックし（図６：Step16）、スポンスイネーブル信号RsEn0がHighの際に、レスポンス優先順位の最も高いレスポンスをレスポンス信号RsMux0を通じて、情報処理装置CPU_CHIPへ送信する（図６：Step17）。レスポンスイネーブル信号RsEn0がLowであれば、情報処理装置CPU_CHIPへ送信は行わない。

【０１６８】
レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpの１つのレスポンスが情報処理装置CPU_CHIPへ完全に送信されると、レスポンススケジュール回路SCHは、スポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへエントリされているレスポンス数をチェックし、最新のレスポンス数をステータスレジスタSTRegへ保存する（図６：Step18）。ここでは、レスポンスイネーブル信号RsEn0がHighであり、レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへエントリされているレスポンスが、レスポンスResRD32b0m0のみのため、レスポンススケジュール回路SCHは、ステータスレジスタSTRegへレスポンス数１を保存し、さらにレスポンスeRsRD32b0m0のレスポンス優先順位を最高位に設定し、レスポンスeRsRD32b0m0を情報処理装置CPU_CHIPへ送信する。レスポンスResRD32b0m0が情報処理装置CPU_CHIPへ送信されると、レスポンススケジュール回路SCHは、レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへエントリされているレスポンスが存在しないため、ステータスレジスタSTRegへレスポンス数0を保存する。

【０１６９】
リクエストReqRD32b0m0に対応するレスポンスResRD32b0m0がレスポンスキュー回路RsQoへエントリされると、レスポンスResRD32b0m0が情報処理装置CPU_CHIPへ出力されている最中でも、リクエストReqWT23b0m0に対する処理を行うことができる(図５：Step4以降)。

【０１７０】
次に、すでにリクエストキューRqQI-0に格納されているリクエストReq WT23b0m0についての動作を説明する。ID比較回路CPQは、リクエストキューRqQI-0へエントリされたリクエストReq WT23b0m0に含まれるID値２と、IDレジスタ回路dstIDに保持されているID値２を比較する（図５：Step4）。比較結果が一致したため、リクエストReq WT23b0m0は、リクエストキュー回路RqQXIへ転送される（図５：Step5）。比較結果が不一致の場合は、リクエストReqWT23b0m0は、リクエストキュー回路RqQXOへ転送され、メモリチップM1へ転送される（図５：Step12）。

【０１７１】
次に、リクエストキュー回路RqQXIは格納しているレスポンスが読み出し命令を含むかどうかチェックする（図５：Step6）。読み出し命令を含んでいる場合は、リクエストキュー回路RqQXIは、レスポンスキュー回路RsQoのレスポンスキューRqQp-0及びRqQp-1に空きがあるかをチェックする（図５：Step7）。リクエストReqWT23b0m0は読み出し命令を含んでいないため、リクエストキュー回路RqQXIは格納しているリクエストReqWT23b0m0をメモリ回路MemVLへ転送する（図５：Step10）。メモリ回路MemVLはリクエストReqWT23b0m0に従って動作する（図５：Step11）。具体的には、メモリ回路MemVLはリクエストReqWT23b0m0に含まれるID値２、32バイト書き込み命令WT、バンクアドレスBK０、カラムアドレスCol127および32バイト分の書き込みデータによって、メモリバンク0のセンスアンプへ、カラムアドレス127を開始アドレスとした32バイト分のデータを書き込む。

【０１７２】
図７はメモリチップM１からメモリチップM0へレスポンスが発生した際の、動作の一例を示すフローチャートである。レスポンス信号RsMux1より、レスポンスクロック信号RqCK1に同期し、メモリチップM0へレスポンスが送信される（図７：Step1）と、レスポンスイネーブル信号ResEn1がLowであれば（図7：Step2）、メモリチップM0のレスポンスキュー回路RsQpへ格納されない。レスポンスイネーブル信号ResEn1がHighであれば（図7：Step2）、メモリチップM0のレスポンスキュー回路RsQpへ格納される（図7：Step3）。レスポンスキュー回路RsQpへレスポンスがエントリされると、レスポンススケジュール回路SCHは、レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへエントリされているレスポンス数を、ステータスレジスタSTRegへ保存する（図６：Step4）。さらに、レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへエントリされているレスポンスに対するレスポンス優先順位を決定する（図６：Step5）。次に、レスポンスイネーブル信号RsEn0をチェックし（図６：Step6）、スポンスイネーブル信号RsEn0がHighの際に、レスポンス優先順位の最も高いレスポンスをレスポンス信号RsMux0より、情報処理装置CPU_CHIPへ

送信する（図６：Step7）。レスポンスイネーブル信号RsEn0がLowであれば、情報処理装置CPU_CHIPへ送信は行わない。

【０１７３】

レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpの１つのレスポンスが情報処理装置CPU_CHIPへ完全に送信されると、レスポンススケジュール回路SCHは、スポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへエントリされているレスポンス数をチェックし、最新のレスポンス数をステータスレジスタSTRegへ保存する（図６：Step8）。

【０１７４】

レスポンススケジュール回路SCHの動作について説明する。図8はレスポンススケジュール回路SCHの動作を示すフローチャートである。レスポンススケジュール回路SCHでは、先ず、レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへレスポンスがエントリされているかをチェックする(Step1)。レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpのいずれにもレスポンスがエントリされていなければ、再度、レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへのエントリをチェックする。レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpのいずれかにレスポンスがエントリされていれば、レスポンスの優先順位をチェックし、最高位のレスポンス優先順位を持つレスポンスの送信準備を行う（Step2）。

【０１７５】

次に、レスポンスイネーブル信号RsEn0をチェックし（Step3）、Lowの際はレスポンスを出力せず、レスポンスイネーブル信号RsEn0がHighを待つレスポンスイネーブル信号RsEn0がHighを待つ。レスポンスイネーブル信号RsEn0がHighの際は、最高位のレスポンス優先順位を持つレスポンスを出力する（Step4）。その、レスポンスが出力された後は、レスポンスに関する出力優先順位を変化させる（Step5）。

【０１７６】

メモリチップM0のレスポンススケジュール回路SCHで行うレスポンス優先順位の変更動作の一例について説明する。図9では、メモリチップM0が装備するレスポンススケジュール回路SCHが行う動的レスポンス優先順位の制御を示す。

【０１７７】

先ず、メモリチップM0での、レスポンス優先順位の制御を説明する。電源投入直後の初期設定（Initial）にて、レスポンスキュー回路RsQoへエントリされるメモリチップM0のレスポンスの優先順位（PRsQo(M0)）は１、レスポンスキュー回路RsQpエントリされるメモリチップM1のレスポンスの優先順位（PRsQp(M1)）は２に、レスポンスキュー回路RsQpエントリされるメモリチップM2のレスポンスの優先順位（PRsQp(M2)）は３に設定される。特に限定しないが、レスポンスの順位の小さい方がレスポンスの順位が高いとする。レスポンスキュー回路RsQoへエントリしたメモリチップM0のレスポンス（RsQo(M0)）がNtime回分出力すると、レスポンスキュー回路RsQoへエントリされるメモリチップM0のレスポンスの優先順位（PRsQo(M0)）は最も低い３となり、メモリチップM1のレスポンスの優先順位（PRsQp(M1)）は最も高い１となり、レスポンスキュー回路RsQPエントリされるメモリチップＭ２のレスポンスの優先順位（PRsQp(M2)）は２となる。

【０１７８】

レスポンスキュー回路RsQpへエントリされるメモリチップＭ１のレスポンスPRsQp(M1))が、Mtime回分出力するとレスポンスキュー回路RsQpへエントリされるメモリチップM1のレスポンスの優先順位（PRsQp(M1)）は最も低い３となり、レスポンスキュー回路RsQPエントリされるメモリチップＭ２のレスポンスの優先順位（PRsQp(M1)）は最も高い1となり、レスポンスキュー回路RsQPoへエントリされるメモリチップＭ０のレスポンスの優先順位（PrsQo(Ｍ０)）は２となる。

【０１７９】

次に、レスポンスキュー回路RsQpへエントリされるメモリチップＭ２のレスポンスPRsQp(M2))が、Ltime回分出力するとレスポンスキュー回路RsQPへエントリされるメモリチップM2のレスポンスの優先順位（PRsQp(M2)）は最も低い３となり、レスポンスキュー回路R

sQPoへエントリされるメモリチップＭＯのレスポンスの優先順位（PrsQo(ＭＯ)）は最も高い1となる。レスポンスキュー回路RsQPエントリされるメモリチップＭ２のレスポンスの優先順位（PRsQp(M1)）は2となる。レスポンスキュー回路RsQoへエントリされるメモリチップM0からのレスポンスのレスポンス優先順位を変更するためのレスポンス出力回数Ntime、レスポンスキュー回路RsQpへエントリされるメモリチップM1からのレスポンスのレスポンス優先順位を変更するためのレスポンス出力回数Mtimeおよびレスポンスキュー回路RsQpへエントリされるメモリチップM2からのレスポンスのレスポンス優先順位を変更するためのレスポンス出力回数Ltimeは、電源投入直後の初期設定（Initial）にて、特に限定しないが、それぞれ、10回、2回、1回に設定される。

【０１８０】

さらに、レスポンス出力回数Ntime、Mtime、Ltimeは、情報処理装置CPU_CHIPから設定可能であり、本発明が利用される携帯機器などのシステム構成にあわせて、高性能化が図れるように、それぞれを設定することができる。

【０１８１】

＜クロック制御＞

図10(a)は、メモリチップM0から出力するレスポンスクロック信号RsCk0を停止する動作の一例である。情報処理装置CPU_CHIPは、レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへエントリされているレスポンス数ResNを確認するために、リクエスト信号RqMux0よりメモリチップM0のID値２とレスポンス数確認命令を多重化したリクエストReqRNoを入力する（Step2）。メモリチップM0のリクエストキュー回路RqQIはリクエストReqRNoを格納する。次に、ID比較回路CPQは、リクエストキュー回路RqQIへ格納されているリクエストReqRNoに含まれるID値２とIDレジスタ回路dstIDに保持されているID値２を比較し、一致したため、リクエストReqBAb0m0は、リクエストキュー回路RqQXIへ転送される。

【０１８２】

リクエストキュー回路RqQXIは、リクエストReqBAb0m0を、ステータスレジスタ回路STRegへする。ステータスレジスタ回路STRegは、ID値２含めて、レスポンス数ResNをレスポンスキュー回路RsQoへ送信し、レスポンスキュー回路RsQoは、レスポンス信号RsMux0を通じて、ID値２およびレスポンス数ResNを情報処理装置CPU_CHIPへ送信する（Step3）。つぎに、ID値２およびレスポンス数ResNを受け取った情報処理装置CPU_CHIPは、レスポンス数ResNが０であるかどうかチェックを行う（Step4）。レスポンス数ResNが０では無い場合、未だ、レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへエントリされているレスポンスが存在するので、再度、レスポンス数確認命令をメモリチップM0へ送信する（Step2）。

【０１８３】

レスポンス数ResNが0の場合は、レスポンスキュー回路RsQoおよびレスポンスキュー回路RsQpへエントリされているレスポンスが存在しないため、リクエスト信号RqMux0より、レスポンスクロック信号RsCk0の停止命令を、メモリチップM0へ送信する（Step5）。リクエスト信号RqMux0よりリクエストとしてID値２、レスポンスクロック停止コマンドが多重化されたリクエストReqStop2がメモリチップM0へ入力する。メモリチップM0はリクエストReqStop2を自身のリクエストキュー制御回路RqCT内のリクエストキューへ格納する。その後、リクエストキュー制御回路RqCT内のID比較回路はリクエストReqStop2に含まれるID値２と自身のIDレジスタの値２を比較する。比較結果は一致しており、リクエストキュー制御回路RqCTはレスポンスクロック制御回路RsCkC内のクロック分周回路Div2へリクエストReqStop2を送信する（Step5）。

【０１８４】

クロック分周回路Div2は、リクエストReqStop2に従い、レスポンスクロック信号RsCK0のクロック周波数を徐々に低下させ、レスポンスクロック信号RsCK0の停止準備が整った時点で、レスポンススケジュール回路SCHを通じて、レスポンス信号RsMux0より、ID値２およびレスポンスクロック停止通達情報を情報処理装置CPU_CHIPへ送信する（Step6）。その後、クロック分周回路Div2は、クロック信号ck3およびレスポンスクロック信号RsCK0

を停止する（Step7）。

【０１８５】

図10(b)は、メモリチップM0から出力するレスポンスクロック信号RsCk0のクロック周波数を低下させるための動作の一例である。図10(b)のStep１からStep4までの動作は、図11（a）と同等であるため、Step5より説明する。リクエスト信号RqMux0よりリクエストとしてID値2、レスポンスクロック分周コマンド、および分周比8が多重化されたリクエストReqDIV8がメモリチップM0へ送信される（Step5）。メモリチップM0は、自身のリクエストキュー制御回路RqCT内のID比較回路にて、リクエストReqDIV8に含まれるID値2と自身のIDレジスタの値2を比較する。比較結果は、一致のため、リクエストReqDIV8は、リクエストクロック制御回路RqCkC内のクロック分周回路Div2へ送信される（Step5）。

【０１８６】

クロック分周回路Div2は、リクエストReqDIV8に従い、レスポンスクロック信号RsCK0のクロック周波数を徐々に低下させ、最終的に、リクエストクロック信号RqC2を8分の1分周したクロックを、クロックCK3およびレスポンスクロック信号RsCk2から出力する（Step6）。レスポンスクロック信号RsCK0のクロック周波数が、所望の周波数へ変更された後、クロック分周回路Div2はレスポンススケジュール回路SCHを通じて、レスポンス信号RsMux0より、ID値２およびレスポンスクロック分周完了情報を情報処理装置CPU_CHIPへ送信する（Step７）。

【０１８７】

図10(c)は、周や停止されたレスポンスクロック信号RsCk0を再度リクエストクロック信号RqCk0と同等の周波数で動作させ場合の動作の一例である。メモリチップM0から出力するレスポンスクロック信号RsCk0のクロック周波数を低下させるための動作の一例である。リクエスト信号RqMux0よりリクエストとしてID値２、レスポンスクロック再開コマンドが多重化されたリクエストReqStart2がメモリチップM0入力する。

【０１８８】

メモリチップM0はリクエストReqStart2を自身のリクエストキュー制御回路RqCT内のリクエストキューへ格納する(Step2)。その後、リクエストキュー制御回路RqCT内のID比較回路はリクエストReqStart2に含まれるID値２と自身のIDレジスタの値２を比較する。比較結果は一致するため、リクエストReqDIV4を自身へのリクエストであると判断する。リクエストキュー制御回路RqCTはレスポンスクロック制御回路RsCkC内のクロック分周回路Div2へリクエストReqStart2を送信する(Step2)。クロック分周回路Div3は、リクエストReqStart2に従い、徐々にクロック周波数を上げ、最終的に、リクエストクロック信号RqCk0と同等の周波数を持つのクロックを、クロックck3およびレスポンスクロック信号RsCK0より出力する(Step3)。

【０１８９】

レスポンスクロック信号RsCK0のクロック周波数が、所望の周波数へ変更された後、クロック分周回路Div2はレスポンススケジュール回路SCHを通じて、レスポンス信号RsMux0より、ID値２およびレスポンスクロック再開完了情報を情報処理装置CPU_CHIPへ送信する（Step4）。上記はレスポンスクロック信号RsCk0についてのクロック制御方法について説明したが、リクエストクロック信号RqCk1についてのクロック制御も同様に行うことができることは言うまでもない。

【０１９０】

図11は、メモリチップM0が装備するメモリ回路MemVLの回路ブロック図の一例である。メモリ回路MemVLは、コマンドデコーダCmdDec、制御回路Cont Logic、ロウアドレスバッファRAdd Lat、カラムアドレスバッファCAdd Lat、リフレッシュカウンタRefC、温度計Thmo、ライトデータバッファWdata Lat、リードデータバッファRDataLat、ロウデコーダRowDec、カラムデコーダColDec、センスアンプSenseAmp、データ制御回路DataCont、メモリバンクBank0～Bank7から構成されている。メモリ回路MemVLの読み出し動作を説明する。

【０１９１】

リクエストキューRqQXIへバンクアドレス７、ロウアドレス５格納されており、バンク

アクティブ命令BAがコマンド信号Commandから、バンクアドレス７およびロウアドレス５が、アドレス信号Addressよりメモリ回路MemVLへ送信される。コマンドデコーダCmdDecはバンクアクティブ命令BAを解読し、制御回路Cont Logicがロウアドレスバッファ RaddLatへバンクアドレス７およびロウアドレス５を格納するように指示する。バンクアドレス７およびロウアドレス５は、制御回路Cont Logicの指示によりロウアドレスバッファRaddへ格納される。ロウアドレスバッファRaddへ格納されたバンクアドレス７によってメモリバンクBank７が選択され、ロウアドレス５はメモリバンクBank７のロウデコーダRowDecへ入力される。その後メモリバンクBank７内のロウアドレス５に接続されているメモリセルが活性化され、1kByte分のデータがメモリバンクBank７内のセンスアンプSenseAmpへ転送される。

【０１９２】

つぎに、リクエストキューRqQXIへ８バイトデータリード命令RD８、バンクアドレス７、カラムアドレス６３が格納されており、８バイトデータリード命令RD８がコマンド信号Commandから、バンクアドレス７およびカラムアドレス６３が、アドレス信号Addressよりメモリ回路MemVLへ送信される。コマンドデコーダCmdDecは８バイトデータリード命令RD８を解読し、制御回路Cont Logicがカラムアドレスバッファ CAddLatへバンクアドレス７およびカラムアドレス６３を格納するように指示する。バンクアドレス７およびカラムアドレス６３は、制御回路Cont Logicの指示によりカラムアドレスバッファCAddLatへ格納される。

【０１９３】

カラムアドレスバッファCaddLatへ格納されたバンクアドレス７によってメモリバンクBank７が選択され、カラムアドレス６３はメモリバンクBank７のカラムデコーダColDecへ入力される。その後メモリバンクBank７内のカラムアドレス63を開始アドレスとして、８バイト分のデータが、データ制御回路DataContを介してリードデータバッファRdataLatへ転送され格納される。その後読み出された８バイト分のデータはレスポンスキュー回路RsQoへ転送される。

【０１９４】

次に、メモリ回路MemVLの書込み動作を説明する。リクエストキューRqQXIへ８バイトデータライト命令WT８、バンクアドレス７、カラムアドレス１２７が格納されており、８バイトデータライト命令RD８がコマンド信号Commandから、バンクアドレス７およびカラムアドレス１２７が、アドレス信号Addressより、８バイトデータがライトデータ信号WDataよりメモリ回路MemVLへ送信される。コマンドデコーダCmdDecは８バイトデータライト命令WT８を解読し、制御回路Cont Logicがカラムアドレスバッファ CAddLatへバンクアドレス７およびカラムアドレス１２７を格納するように、ライトデータバッファWdata Latへ８バイト分のライトデータを格納するように指示する。バンクアドレス７およびカラムアドレス１２７は、制御回路Cont Logicの指示によりカラムアドレスバッファCAddLatへ格納される。８バイト分のライトデータは制御回路Cont Logicの指示によりライトデータバッファWdata Latへ格納される。

【０１９５】

カラムアドレスバッファCaddLatへ格納されたバンクアドレス７によってメモリバンクBank７が選択され、カラムアドレス１２７はメモリバンクBank７のカラムデコーダColDecへ入力される。その後メモリバンクBank７内のカラムアドレス１２７を開始アドレスとして、８バイト分のデータが、ライトデータバッファWdata Latから、データ制御回路DataContを介して、メモリバンクBank７内のセンスアンプSenseAmpへ転送され、モリバンクBank７内のロウアドレス５に接続され活性化されているメモリセルへ書き込まれる。

【０１９６】

つぎに、リフレッシュ動作について説明する。メモリ回路MemVLは揮発性メモリのため、データ保持のために定期的にリフレッシュ動作を行う必要がある。リクエストキューRqQXIへ格納されているリフレッシュ命令REFが、コマンド信号Commandより入力する。コマンドデコーダCmdDecは、リフレッシュ命令REFを解読し、制御回路Cont Logicがリフレッ

シュカウンタRefCへリフレッシュ動作を行うように指示する。リフレッシュカウンタRefCは制御回路Cont Logicの指示により、リフレッシュ動作を行う。

【０１９７】

つぎに、セルフリフレッシュ動作について説明する。メモリ回路MemVLへのリクエストが長期間生じないときは、セルフリフレッシュ状態へ動作モードを切り替え、メモリ回路MemVL自らがリフレッシュ動作を行うことができる。

【０１９８】

リクエストキューRqQXIへ格納されているセルフリフレッシュ・エントリ命令SREFが、コマンド信号Commandより入力する。コマンドデコーダCmdDecは、セルフリフレッシュ・エントリ命令SREFを解読し、制御回路Cont Logicは全回路をセルフリフレッシュ状態へ動作モードを切り替える。さらに、リフレッシュカウンタRefCへ、自動的に、定期的にセルフリフレッシュ動作を行うように指示する。リフレッシュカウンタRefCは制御回路Cont Logicの指示により、自動的に、定期的にセルフリフレッシュ動作を行う。

【０１９９】

この際のセルフリフレッシュ動作では、温度によってセルフリフレッシュの頻度を変化させることができる。

【０２００】

一般的に、揮発性メモリでは、温度が高い場合はデータ保持時間が短くなり、低い場合は長くなるという性質がある。そこで、温度計で温度を検知し、温度が高い場合が、セルフリフレッシュの周期を短くし、温度が低い場合はセルフリフレシュの周期を長くし、セルフリフレッシュ動作を行う。これによって、無駄なセルフリフレシュ動作を削減でき低電力化が図れる。

【０２０１】

セルフリフレッシュ状態を抜け出すには、セルフリフレッシュ・解除命令SREFXを、コマンド信号Commandより入力することで実現できる。セルフリフレッシュ状態を抜け出した後の、データ保持動作はリフレッシュ命令REFによって行われる。

【０２０２】

＜メモリチップM1の説明＞

図12は、メモリチップＭ１の構成図の一例である。メモリチップM1は、リクエストインターフェース回路ReqIFと、レスポンスインターフェース回路ResIFと、初期化回路INIT1、メモリ回路MemNV1から構成されている。リクエストインターフェース回路ReqIFはリクエストクロック制御回路RqCkCおよび、リクエストキュー制御回路RqCTから構成される。リクエストクロック制御回路RqCkCはクロックドライバDrv1およびクロック分周回路Div1から構成される。リクエストキュー制御回路RqCTはリクエストキュー回路RqQI、リクエストキュー回路RqQXI、リクエストキュー回路RqQXO、IDレジスタ回路dstID、ID比較回路CPQから構成される。レスポンスインターフェース回路ResIFはレスポンスクロック制御回路RsCkCおよび、レスポンスキュー制御回路RsCTから構成される。

【０２０３】

レスポンスクロック制御回路RsCkCはクロックドライバDrv2およびクロック分周回路Div2から構成される。レスポンスキュー制御回路RsCTは、レスポンスキュー回路RsQo、レスポンスキュー回路RsQp、ステータスレジスタ回路STReg、レスポンススケジュール回路SCHから構成される。メモリ回路MemNV1は、特に限定しないが、不揮発性メモリであり、NOR型フラッシュメモリセルを利用したNOR型フラッシュメモリである。メモリ回路MemNV1には、ブートデバイスID値BotIDおよび終端デバイスID値EndIが格納される。メモリ回路MemNV1および初期化回路INIT1以外の、メモリチップ１を構成する回路および動作は、図４のメモリチップM0と同等である。

【０２０４】

次に、本メモリチップM1の動作を説明する。まず、電源投入時の動作について説明する。メモリチップM1へ電源が投入されると初期化回路INIT1はメモリチップM1の初期化を行う。メモリチップM1は、ブートデバイス認識信号Bsigが接地gndされているので、自分自

身がブートデバイスであることを認識し、自らのメモリ回路MemNV1が保持しているブートデバイスID値１をIDレジスタdstIDへ設定し、ID有効ビットをHighにする。

【０２０５】

次にレスポンススケジュール回路SCHが持つレスポンスキュー回路RsQoに入力するレスポンスの優先順位を１へ、レスポンスキュー回路RsQpに入力するメモリチップM2からのレスポンスの優先順位を2へ設定する。クロック分周回路Div1およびDiv2の分周比は１に設定される。初期化回路INIT1による初期設定が終了すると、メモリチップM1はメモリチップM1とメモリチップM2との間で通信できることを確認する通信確認動作を行う。メモリチップM1はリクエストイネーブル信号RqEn2がHighになったことを確認し、レスポンスイネーブル信号RsEn2及びリクエストイネーブル信号RqEn1をHighにする。

【０２０６】

次に、メモリチップM0は、リクエストイネーブル信号RqEn1がHighになったことを確認し、レスポンスイネーブル信号RsEn1をHighにする。通信確認動作が終了すると、メモリ回路MemNV1よりブートデータが読み出され、メモリチップM0を介して、情報処理装置CPU_CHIPへ送信される。次に、メモリチップＭ１での、レスポンス優先順位の制御を説明する。

【０２０７】

図13ではメモリチップM1が装備するレスポンススケジュール回路SCHが行う動的レスポンス優先順位の制御を示す。

【０２０８】

図１に示すように、メモリチップM1へは、メモリチップM0のレスポンスは生じない接続構成になっている場合は、メモリチップM1のレスポンスおよびメモリチップM2のレスポンスについてのみレスポンスの優先順位が付けられる。電源投入直後の初期設定（Initial）にて、レスポンスキュー回路RsQoへエントリされるメモリ回路MemNV1からのレスポンスの優先順位（PRsQo(Ｍ１)）は１、レスポンスキュー回路RsQpエントリされるメモリチップＭ２からのレスポンスの優先順位（PRsQp(Ｍ２)）は２に設定される。特に限定しないが、レスポンスの順位の小さい方がレスポンスの順位が高いとする。

【０２０９】

次に、レスポンスキュー回路RsQoへエントリしたメモリ回路MemNV1のレスポンス（RsQo(Ｍ１)）がＭ１time回分出力すると、レスポンスキュー回路RsQoへエントリされるレスポンスの優先順位（PRsQo(Ｍ１)）は最も低い２となり、メモリチップＭ２のレスポンスの優先順位（PRsQp(Ｍ２)）は最も高い１となる。

【０２１０】

次に、レスポンスキュー回路RsQpへエントリされるメモリチップＭ２からのレスポンスPRsQp(M2)）が、L1time回分出力するとレスポンスキュー回路RsQpへエントリされるメモリチップM2からのレスポンスの優先順位（PRsQp(M2)）は最も低い２となり、レスポンスキュー回路RsQoエントリされるレスポンスの優先順位（PrsQp(M1)）は最も高い1となる。レスポンスキュー回路RsQoへエントリされるメモリ回路MemNV1からのレスポンスのレスポンス優先順位を変更するためのレスポンス出力回数M1time、レスポンスキュー回路RsQpへエントリされるメモリチップM2からのレスポンスのレスポンス優先順位を変更するためのレスポンス出力回数L1timeは、電源投入直後の初期設定（Initial）にて、特に限定しないが、それぞれ、10回、1回に設定される。さらに、レスポンス出力回数M1time、L1timeは、情報処理装置CPU_CHIPから設定可能であり、本発明が利用される携帯機器などのシステム構成にあわせて、高性能化が図れるように、それぞれを設定することができる。

【０２１１】

また、メモリチップM1が装備するレスポンススケジュール回路SCHが行う動的レスポンス優先順位の制御は、図８で示した動作と同等である。また、リクエストクロック信号RqCk2およびレスポンスクロック信号RsCk1ののクロック制御方法は、図１０で示したクロック制御方法と同様である。

【０２１２】

＜メモリチップＭ２の説明＞

図14は、メモリチップM2の構成図の一例である。メモリチップＭ２は、リクエストインターフェース回路ReqIFと、レスポンスインターフェース回路ResIFと、初期化回路INIT２、メモリ回路MemNV２から構成されている。リクエストインターフェース回路ReqIFはリクエストクロック制御回路RqCkCおよび、リクエストキュー制御回路RqCTから構成される。リクエストクロック制御回路RqCkCはクロックドライバDrv1およびクロック分周回路Div1から構成される。リクエストキュー制御回路RqCTはリクエストキュー回路RqQI、リクエストキュー回路RqQXI、リクエストキュー回路RqQXO、IDレジスタ回路dstID、ID比較回路CPQから構成される。レスポンスインターフェース回路ResIFはレスポンスクロック制御回路RsCkCおよび、レスポンスキュー制御回路RsCTから構成される。レスポンスクロック制御回路RsCkCはクロックドライバDrv2およびクロック分周回路Div2から構成される。

【０２１３】

レスポンスキュー制御回路RsCTは、レスポンスキュー回路RsQo、レスポンスキュー回路RsQp、ステータスレジスタ回路STReg、レスポンススケジュール回路SCHから構成される。メモリ回路MemNV２は、特に限定しないが、不揮発性メモリであり、NAND型フラッシュメモリセルを利用したNAND型フラッシュメモリである。メモリ回路MemNV２および初期化回路INIT２以外の、メモリチップ１を構成する回路および動作は、図４のメモリチップM0と同等である。

【０２１４】

次に、本メモリチップＭ２の動作を説明する。まず、電源投入時の動作について説明する。メモリチップM2へ電源が投入されると初期化回路INIT２はメモリチップＭ２の初期化を行う。先ず、IDレジスタ回路dstIDの持つIDレジスタの値を０へ、ID有効ビットをLowへ初期設定する。次にレスポンススケジュール回路SCHが持つレスポンスキュー回路RsQoへ入力するレスポンスの優先順位を１へ設定する。クロック分周回路Div1およびDiv2の分周比は１に設定される。初期化回路INIT２による初期設定が終了すると、メモリチップＭ２はメモリチップＭ１との間で通信できることを確認する通信確認動作を行う。メモリチップM2は、RqEn3、RsMux3、RqCk3を接地（gnd）していることによって、直列接続しているメモリチップの最も終端のメモリチップであることを認識し、リクエストイネーブル信号RqEn2をHighにする。

【０２１５】

次に、メモリチップM1はリクエストイネーブル信号RqEn2がHighになったことを確認し、レスポンスイネーブル信号RsEn2及びリクエストイネーブル信号RqEn1をHighにする。次に、メモリチップＭ２での、レスポンス優先順位の制御を説明する。図１５ではメモリチップＭ２が装備するレスポンススケジュール回路SCHが行う動的レスポンス優先順位の制御を示す。図１に示すように、メモリチップM2が直列接続の最終チップである場合は、メモリチップM2へはメモリチップM0およびメモリチップM1のレスポンスは生じない。

【０２１６】

そのため、メモリチップM2のレスポンスについてのみレスポンスの優先順位が付けられる。したがって、電源投入直後の初期設定（Initial）にて、レスポンスキュー回路RsQOへエントリされるメモリチップＭ２のレスポンスの優先順位（PRsQO(Ｍ２)）は１に設定された後は、変化しない。レスポンスキュー回路RsQoへエントリされるメモリ回路NV2のレスポンスの優先順位（PRsQO(Ｍ２)）を変更することが無いため、レスポンスキュー回路RsQoへエントリされるメモリチップM2からのレスポンスのレスポンス優先順位を変更するためのレスポンス出力回数は、電源投入直後の初期設定（Initial）にて、特に限定しないが、０回に設定され、変更する必要がない。また、レスポンスクロック信号RsCk2のクロック制御方法は、図１０で示したクロック制御方法と同様である。

【０２１７】

図１６は、情報処理装置CPU_CHIPからメモリモジュールMEMへ送信されたリクエストに含まれるID値がメモリチップM0、M1およびM2のIDレジスタ値のいずれにも一致せず、エラーが発生した場合の動作の一例を示すフローチャートである。情報処理装置CPU_CHIPから

リクエストとID値がメモリモジュールMEMへ送信される（Step1）。リクエストイネーブル信号RqEn0がLowであれば(Step2)、情報処理装置CPU_CHIPからのリクエストはメモリチップM0のリクエストキュー回路RqQIへ格納されない。リクエストイネーブル信号RqEn0がHighであれば（Step2）、メモリチップM0の、リクエストキュー回路RqQIへ格納される(Step3)。

【０２１８】

その後、ID比較回路CPQは、リクエストキュー回路RqQIへエントリされたリクエストに含まれるID値とIDレジスタ回路dstIDに保持されているID値を比較する（Step4）。比較結果が一致すれば、リクエストキュー回路RqQIへエントリされたリクエストはリクエストキュー回路RqQXIへ転送される（Step5）。比較結果が不一致の場合は、メモリチップM0が最終端のメモリチップかどうかをチェックする（Step6）。メモリチップM0が最終端のデバイスではないので、リクエストキュー回路RqQIへエントリされたリクエストはリクエストキュー回路RqQXOへ転送され、さらに、次のメモリチップM1へ転送される（Step9）。メモリチップM1では、Step1からStep９を繰り返す。メモリチップＭ２では、Step1からStep4を行う。Step4での比較結果が一致すれば、リクエストキュー回路RqQIへエントリされたリクエストはリクエストキュー回路RqQXIへ転送される（Step5）。比較結果が不一致の場合は、メモリチップM0が最終端のメモリチップかどうかをチェックする（Step6）。

【０２１９】

メモリチップＭ２は最終端のメモリチップであるため、情報処理装置CPU_CHIPからメモリモジュールMEMへ送信されたリクエストに含まれるID値がメモリチップM0、M1およびM2のIDレジスタ値のいずれにも一致せず、IDエラーとなる（Step７）。IDエラーは、最終端のメモリチップM2からメモリチップM1およびM2と経由して情報処理装置CPU_CHIPへ送信される。

【０２２０】

次に、メモリモジュールMEMへ入力するリクエストの動作波形について説明する。図１７および図１８は、情報処理装置CPU_CHIPが、メモリモジュールMEMへ送信するリクエストの動作波形およびメモリモジュールMEMから情報処理装置CPU_CHIPへのレスポンスの動作波形の一例である。

【０２２１】

図17(a)は、メモリチップM0へのバンクアクティブ命令BAを含むバンクアクティブリクエストである。特に限定しないが、バンクアクティブリクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、バンクアクティブ命令BA、アドレスAD20およびAD21が多重化されメモリチップM0へ入力される。アドレスAD20およびAD21には、バンクアドレスおよびロウアドレスが含まれる。本バンクアクティブリクエストによって、メモリチップM0内のメモリバンクの1つが活性化される。

【０２２２】

図17(b)は、メモリチップM0への４バイトデータリード命令RD4を含むリードリクエストである。特に限定しないが、リードリクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、リード命令RD4、アドレスAD22およびAD22が多重化されメモリチップM0へ入力される。アドレスAD22およびAD23には、バンクアドレスおよびカラムアドレスが含まれる。本リードリクエストによって、メモリチップM0内の活性化されているメモリバンクからデータが読み出される。

【０２２３】

図17(c)は、メモリチップM0のID値およびメモリチップM0から読み出されたデータを含むリードレスポンスである。特に限定しないが、リードレスポンスは、レスポンスイネーブル信号RsEN0がHighの際に、レスポンスクロック信号RsCk0に同期して、メモリチップM0のID値ID２、４バイト分のデータD0、D1、Ｄ２およびD3が多重化され、情報処理装置CPU_CHIPへ入力される。

【０２２４】
図17(d)は、メモリチップM0への２バイトデータの書込み命令WT2を含むライトリクエストである。特に限定しないが、ライトリクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、ライト命令WT2、アドレスAD24およびAD25が多重化されメモリチップM0へ入力される。アドレスAD22およびAD23には、バンクアドレスおよびカラムアドレスが含まれる。本ライトリクエストによって、メモリチップM0内の活性化されているメモリバンクへデータが書き込まれる。
【０２２５】
図17(e)は、メモリチップM0へのプリチャージ命令PREを含むプリチャージリクエストである。特に限定しないが、プリチャージリクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、プリチャージ命令PRE、アドレスAD28が多重化されメモリチップM0へ入力される。アドレスAD28には、バンクアドレスが含まれる。本プリチャージリクエストによって、メモリチップM0内のメモリバンクの1つが非活性化される。
【０２２６】
図18(a)は、メモリチップM0へのオートリフレッシュ命令REFを含むリフレッシュリクエストである。特に限定しないが、リフレッシュリクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、リフレッシュ命令REFが多重化されメモリチップM0へ入力される。本リフレッシュリクエストREFによって、メモリチップM0に対してリフレッシュ動作が行われる。図18(b)は、メモリチップM0へのセルフリフレッシュ命令SREFを含むセルフリフレッシュエントリリクエストである。特に限定しないが、セルフリフレッシュエントリリクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID値ID２、セルフリフレッシュエントリ命令SREFおよび全メモリバンク指定ALL、自動温度補償無効指定ATInvが多重化され、メモリチップM0へ入力される。本セルフリフレッシュエントリリクエストによって、メモリチップM0は、セルフリフレッシュ状態となり、メモリチップM0自信が内部で自動的に、全メモリバンクに対するリフレシュ動作を行う。
【０２２７】
図18(c)は、メモリチップM0へのセルフリフレッシュ命令SREFを含むセルフリフレッシュエントリリクエストである。特に限定しないが、セルフリフレッシュエントリリクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、セルフリフレッシュエントリ命令SREFおよび全メモリバンク指定BK7および自動温度補償無効指定ATInvが多重化されメモリチップM0へ入力される。本セルフリフレッシュエントリリクエストによって、メモリチップM0は、セルフリフレッシュ状態となり、メモリチップM0自信が内部で自動的に、メモリバンク７のみに対するリフレシュ動作を行う。
【０２２８】
図18(d)は、メモリチップM0へのセルフリフレッシュ命令SREFを含むセルフリフレッシュエントリリクエストである。特に限定しないが、セルフリフレッシュエントリリクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、セルフリフレッシュエントリ命令SREFおよび全メモリバンク指定BK7および自動温度補償有効指定ATVldが多重化されメモリチップM0へ入力される。本セルフリフレッシュエントリリクエストによって、メモリチップM0は、セルフリフレッシュ状態となり、メモリチップM0自信が内部で自動的に、メモリバンク７のみに対するリフレシュ動作を行う。また、自動温度補償有効指定ATVldがあるため、特に限定しないがメモリチップM0の内部に組み込んだ温度センサーで周囲温度を検知し、温度に応じてセルフリフレッシュの頻度を自動的に調節することができる。
【０２２９】

図18(e)は、メモリチップM0へのセルフリフレッシュ解除命令SREXを含むセルフリフレッシュExitリクエストである。特に限定しないが、セルフリフレッシュExitリクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID値ID２、セルフリフレッシュ解除命令SREXが多重化され、メモリチップM0へ入力される。本セルフリフレッシュExitリクエストによって、メモリチップM0は、セルフリフレッシュ状態から抜け出す。

【０２３０】

図19(a)は、メモリチップM0へのパワーダウンエントリ命令PDEを含むパワーダウンエントリリクエストである。特に限定しないが、パワーダウンエントリリクエストPDEは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、パワーダウンエントリ命令PDEが多重化されメモリチップM0へ入力される。本パワーダウンエントリリクエストによって、メモリチップM0は、パワーダウン状態となり、メモリチップM0の内部クロックを非活性にする。本実施例では、メモリチップＭ０へのパワーダウンエントリリクエストを説明したが、パワーダウンエントリ命令は、メモリチップのID値を変えることで、メモリモジュールMEM内の全てのメモリチップに対して適用できることは言うまでもない。

【０２３１】

特に限定しないが、メモリチップＭ１のID値ID1およびパワーダウンエントリ命令PDEを多重化したリクエストは、メモリチップM0を介してメモリチップM1へ送信され、メモリチップＭ１の内部クロックを非活性にする。また、特に限定しないが、メモリチップM2のID値ID2およびパワーダウンエントリ命令PDEを多重化したリクエストは、メモリチップM0およびM1を介してメモリチップM2へ送信され、メモリチップM2の内部クロックを非活性にする。

【０２３２】

図19(ｂ)は、メモリチップM0へのパワーダウン解除命令PDXを含むパワーダウン解除リクエストである。特に限定しないが、パワーダウン解除リクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、パワーダウン解除命令PDXが多重化されメモリチップM0へ入力される。本パワーダウン解除リクエストによって、メモリチップM0は、パワーダウン状態から解除する。本実施例では、メモリチップＭ０へのパワーダウン解除リクエストを説明したが、パワーダウン解除リクエストに含まれるID値を変えることでメモリモジュールMEM内の全てのメモリチップに対して適用できることは言うまでもない。

【０２３３】

図19(ｃ)は、メモリチップM0へのディープパワーダウンエントリ命令DPDEを含むディープパワーダウンエントリリクエストである。特に限定しないが、ディープパワーダウンエントリリクエストDPDEは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、ディープパワーダウンエントリ命令PDEが多重化されメモリチップM0へ入力される。本ディープパワーダウンエントリリクエストによって、メモリチップM0は、ディープパワーダウン状態となり、メモリチップM0の内部クロックを非活性にした上でさらに、リフレッシュ用の内部クロック回路をも停止する。本実施例では、メモリチップＭ０へのパワーダウンエントリリクエストを説明したが、パワーダウンエントリリクエストに含まれるメモリチップのID値を変えることで、メモリモジュールMEM内のそれぞれのメモリチップに対して適用できることは言うまでもない。

【０２３４】

図19(d)は、メモリチップM0へのディープパワーパワーダウン解除命令DPDXを含むディープパワーパワーダウン解除リクエストである。特に限定しないが、ディープパワーダウン解除リクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、ディープパワーダウン解除命令PDXが多重化されメモリチップM0へ入力される。本ディープパワーダウン解除リクエストによって

、メモリチップM0は、ディープパワーダウン状態から解除する。本実施例では、メモリチップＭ０へのディープパワーダウン解除リクエストを説明したが、ディープパワーダウン解除リクエストに含まれるID値を変えることでメモリモジュールMEM内それぞれのメモリチップに対して適用できることは言うまでもない。

【０２３５】

図19(e)は、メモリチップM0へのステータスレジスタリード命令STRDを含むステータスレジスタリードリクエストである。特に限定しないが、ステータスレジスタリードリクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のＩＤ２、ステータスレジスタリード命令STRD、レスポンスエントリ数指定情報QCHが多重化されメモリチップM0へ入力される。本ステータスレジスタリード命令STRDおよびレスポンスエントリ数指定情報QCHによって、メモリチップM0は、レスポンスキューへエントリされているれレスポンス数を情報処理装置CPUへ送信する。

【０２３６】

図20(a)は、メモリチップM1への４バイトデータリード命令RD4を含むリードリクエストである。特に限定しないが、リードリクエストは、メモリチップM0を介して、リクエストイネーブル信号RqEN1がHighの際に、リクエストクロック信号RqCk1に同期して、メモリチップM1のID値ID1、リード命令RD4、アドレスAD10、AD11,AD12およびAD13が多重化されメモリチップM1へ入力される。本リードリクエストによって、メモリチップＭ１内のメモリ回路NV1からデータが読み出される。

【０２３７】

図20(b)は、メモリチップM1のID値およびメモリチップM1から読み出されたデータを含むリードレスポンスである。特に限定しないが、リードレスポンスは、レスポンスイネーブル信号RsEN1がHighの際に、レスポンスクロック信号RsCk1に同期して、メモリチップM1のID値ID1、４バイト分のデータD0、D1、Ｄ２およびD3が多重化され、メモリチップM0へ送信され、さらに情報処理装置CPU_CHIPへ送信される。

【０２３８】

図20(c)は、メモリチップM2への512バイトデータリード命令RD512を含むリードリクエストである。特に限定しないが、リードリクエストは、メモリチップM0およびM1を介して、リクエストイネーブル信号RqEN2がHighの際に、リクエストクロック信号RqCk2に同期して、メモリチップM2のID値ID3、リード命令RD512、アドレスAD30、AD31,AD32およびAD33が多重化されメモリチップM3へ入力される。本リードリクエストによって、メモリチップM3内のメモリ回路NV2から512バイト分のデータが読み出される。

【０２３９】

図20(d)は、メモリチップM2のID値ID3およびメモリチップM2から読み出されたデータを含むリードレスポンスである。特に限定しないが、リードレスポンスは、レスポンスイネーブル信号RsEN2がHighの際に、レスポンスクロック信号RsCk2に同期して、32バイト分のデータ毎にメモリチップM2のID値ID1が多重化され、順々に、メモリチップＭ１へ送信され、さらにM0へ送信され、最後に情報処理装置CPU_CHIPへ送信される。最終的に512バイト分のデータが情報処理装置CPU_CHIPへ送信される。

【０２４０】

図21(a)は、メモリチップM1への1バイトデータの書込み命令WT1を含むライトリクエストである。特に限定しないが、ライトリクエストは、メモリチップM0を介して、リクエストイネーブル信号RqEN1がHighの際に、リクエストクロック信号RqCk1に同期して、メモリチップM1のID値ID1、ライト命令WT1、アドレスAD10,AD11,AD12およびAD13、書込みデータD0が多重化されメモリチップＭ１へ入力される本ライトリクエストによって、メモリチップM1内のメモリ回路NV1へ１バイト分のデータが書き込まれる。

【０２４１】

図21(b0)および(b1)は、メモリチップM2への512バイトデータの書込み命令WT512を含むライトリクエストである。特に限定しないが、ライトリクエストは、メモリチップM0およびM1を介して、リクエストイネーブル信号RqEN2がHighの際に、リクエストクロック信号R

qCk2に同期して、メモリチップM2のID値ID3、ライト命令WT512、アドレスAD30,AD31,AD32およびAD33、512バイト分の書き込みデータD0～D511が多重化されメモリチップM2へ入力される。本ライトリクエストによって、メモリチップM2内のメモリ回路NV2へ512バイト分のデータが書き込まれる。

【０２４２】

図22(a)は、メモリチップM0のレスポンスクロックRsCk0のドライブ能力を変更するためのレスポンスクロックドライブ能力指定命令DPDEを含むレスポンスクロックドライブ能力指定リクエストである。特に限定しないが、レスポンスクロックドライブ能力指定リクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、レスポンスクロックドライブ能力指定命令DPDEおよびドライブ能力値DrvC4が多重化されメモリチップM0へ入力される。本リクエストによって、メモリチップM0のレスポンスクロック信号RsCk0のドライブ能力が、基準ドライブ能力の４分の１に設定される。本実施例では、メモリチップＭ０のレスポンスクロックRsCk0のドライブ能力を変更する場合について説明したが、レスポンスクロックドライブ能力指定リクエストに含まれるメモリチップのID値を変えることで、メモリモジュールMEM内のそれぞれのメモリチップのレスポンスクロックに対するドライブ能力を変更できることは言うまでもない。

【０２４３】

図22(ｂ)は、メモリチップM0から出力するレスポンスクロック信号RsCk0以外の信号で、レスポンスクロック信号RsCk0と同一出力方向の信号（RsMux0およびRqEN1）のドライブ能力を変更するためのアップストリーム信号ドライブ能力指定命令Updrを含むアップストリーム信号ドライブ能力指定リクエストである。特に限定しないが、アップストリーム信号ドライブ能力指定リクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、アップストリーム信号ドライブ能力指定命令Updrおよびドライブ能力値DrvC2が多重化されメモリチップM0へ入力される。本リクエストによって、メモリチップM0から出力されるレスポンスクロック信号RsCk0以外の信号で、レスポンスクロック信号RsCk0と同一出力方向の信号（RsMux0およびRqEN1）レスポンス信号のドライブ能力が、基準ドライブ能力の２分の１に設定される。本実施例では、メモリチップＭ０の場合について説明したが、アップストリーム信号ドライブ能力指定リクエストに含まれるメモリチップのID値を変えることで、メモリモジュールMEM内のそれぞれのメモリチップのアップストリーム信号に対するドライブ能力を変更できることは言うまでもない。

【０２４４】

図22(c)は、メモリチップM0のリクエストクロックRqCk1のドライブ能力を変更するためのリクエストクロックドライブ能力指定命令Rsckdrを含むリクエストクロックドライブ能力指定リクエストである。特に限定しないが、リクエストクロックドライブ能力指定リクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、リクエストクロックドライブ能力指定命令Rsckdrおよびドライブ能力値DrvC8が多重化され、メモリチップM0へ入力される。本リクエストによって、メモリチップM0のリクエストクロック信号RsCk1のドライブ能力が、基準ドライブ能力の８分の１に設定される。本実施例では、メモリチップＭ０のリクエストクロックRsCk1のドライブ能力を変更する場合について説明したが、リクエストクロックドライブ能力指定リクエストに含まれるメモリチップのID値を変えることで、メモリモジュールMEM内のそれぞれのメモリチップのリクエストクロックに対するドライブ能力を変更できることは言うまでもない。

【０２４５】

図22(d)は、メモリチップM0から出力するリクエストクロック信号RsCk0以外の信号で、リクエストクロック信号RqCkqと同一出力方向の信号（RqMux1およびRsEN0）のドライブ能力を変更するためのダウントリーム信号ドライブ能力指定命令Dwndrを含むダウンストリーム信号ドライブ能力指定リクエストである。特に限定しないが、ダウンストリーム信号

ドライブ能力指定リクエストは、リクエストイネーブル信号RqEN0がHighの際に、リクエストクロック信号RqCk0に同期して、メモリチップM0のID２、ダウンストリーム信号ドライブ能力指定命令Updrおよびドライブ能力値DrvC2が多重化されメモリチップM0へ入力される。本リクエストによって、メモリチップM0から出力されるリクエストクロック信号RqCk1以外の信号で、リクエストクロック信号RqCk1と同一出力方向の信号（RqMux1およびRsEN0）リクエスト信号のドライブ能力が、基準ドライブ能力と同等に設定される。本実施例では、メモリチップＭ０の場合について説明したが、ダウンストリーム信号ドライブ能力指定リクエストに含まれるメモリチップのID値を変えることで、メモリモジュールMEM内のそれぞれのメモリチップのダウンストリーム信号に対するドライブ能力を変更できることは言うまでもない。

【０２４６】

図２３は情報処理装置CPU_CHIPからメモリチップＭ１へ読み出しリクエストが生じ、連続して、メモリチップＭ０へ読み出しリクエストが生じた場合のデータ転送波形を示す。情報処理装置CPU_CHIPは、リクエスト信号RqMux0を通じて、ID値１、2バイトデータ読み出し命令NRD2およびアドレスAD0、AD1を多重化したリクエストReqNRD2をメモリチップM0へ転送する。続いて、リクエスト信号RqMux0を通じて、ID値２、2バイトデータ読み出し命令RD2、アドレスAD0、AD1を多重化したリクエストReqRD2をメモリチップM0へ転送する。メモリチップM0のリクエストキューRqQIへリクエストReqNRD2およびリクエストReqRD2が入力される。リクエストReqNRD2はメモリチップM1へのリクエストのため、メモリチップM0のリクエストキューRqQXOへ転送される。また、リクエストReqNRD2はリクエスト信号RqMux1を通じてメモリチップM1へ転送される。リクエストReqNRD2はメモリチップM1のリクエストキューRqQIへ入力され、次にリクエストキューRqQXIへ転送される。リクエストReqNRD2に対応するデータがメモリチップM1のメモリ回路MemNV1から読み出され、IDレジスタ値１も含めて、レスポンスRsNRD2としてレスポンスキューRsQoへ入力される。レスポンスキューRsQoへ入力されたレスポンスRsNRD2は、レスポンス信号RqMux1を通じて転送され、メモリチップM0のレスポンスキューRsQpへ格納される。レスポンスキューRsQpへ格納されたレスポンスRsNRD2は、レスポンス信号ResMux0を通じて、ID値１と読み出しデータとして出力される。

【０２４７】

リクエストReqRD2はメモリチップＭ０へのリクエストのため、メモリチップM0のリクエストキューRqQXIへ転送される。リクエストReqRD2に対応するデータがメモリチップM0のメモリ回路MemVLから読み出され、IDレジスタ値2も含めて、レスポンスRsRD2としてレスポンスキューRsQoへ入力される。レスポンスキューRsQoへ入力されたレスポンスRsRD2は、レスポンス信号RqMux0を通じて、ID値2と読み出しデータとして出力される。リクエストReqRD2がメモリチップM0のリクエストキューRqQIへ入力され、このリクエストに対するレスポンスResRD2がレスポンス信号ResMux0から出力される時間は、約15nsである。一方、リクエストReqNRD2がメモリチップM1のリクエストキューRqQIへ入力され、このリクエストに対するレスポンスResRD2がレスポンス信号ResMux0から出力される時間は、約70nsである。そのため、リクエストReqRD2がリクエストReqNRD2の後に入力されたにもかかわらず、先に出力できる。本実施例ではデータ読み出しを中心に説明したが、データの書込み動作においても同様の動作を行うことができるのは言うまでもない。また、本実施例では、メモリチップM0とM1とのデータ転送動作を説明したが、M1とその他のメモリチップについても同様のデータ転送動作を行うことは言うまでもない。

【０２４８】

以上説明したように、リクエストの入力順序によらず、メモリチップの読み出し時間が異なる場合でも、早く読み出せるデータは、遅く読み出すデータを待つことなく、すぐに読み出すことができるため、高速化が可能となる。さらに、リクエストへIDを付加することで、確実に要求先へリクエストが転送され、また、レスポンスへIDを付加することで、リクエストの入力順序と、読み出しデータの順番が異なった場合でも、情報処理装置CPU__CHIPは転送元のメモリチップを知ることができるため、情報処理装置CPU__CHIPおよび

メモリチップの直列接続によって、接続信号数を少なくしながらも、情報処理装置CPU_CHIPは所望の処理を実行することができる。

【実施例２】

【０２４９】

図２４は、本発明の第２の実施形態である。情報処理装置CPU_CHIPとメモリモジュールMEM２４から構成される情報処理システムを示した実施例である。

【０２５０】

メモリモジュールMEM２４はダイナミックランダムアクセスメモリDRAM0およびDRAM1、NOR型フラッシュメモリNORおよびNAND型フラッシュメモリから構成される。

【０２５１】

情報処理装置CPU__CHIPは、図１で示したものと同等である。ダイナミックランダムアクセスメモリDRAM0およびDRAM1は、図４で示したメモリと同等である。NOR型フラッシュメモリNORは、図１２示したメモリと同等である。NAND型フラッシュメモリNANDは図１４で示したメモリと同等である。

【０２５２】

本発明では、容易に、ダイナミックランダムアクセスメモリを複数接続することができ、情報処理装置CPU_CHIPが必要とするワーク領域やコピー領域を用意に拡張でき、高速処理が可能となる。

【０２５３】

本実施例では、ダイナミックランダムアクセスメモリの複数接続について説明したが、NOR型フラッシュメモリNORや、NAND型フラッシュメモリNANDは、必要に応じて複数個接続でき、容易にプログラム領域やデータ領域を拡張でき、携帯機器のシステム構成に合わせて柔軟に対応できる。

【実施例３】

【０２５４】

図２５は、本発明の第３の実施形態である。情報処理装置CPU_CHIPとメモリモジュールMEM２５から構成される情報処理システムを示した実施例である。情報処理装置CPU__CHIPは、図１で示したものと同等である。NOR型フラッシュメモリNORは、図１２示したメモリと同等である。ダイナミックランダムアクセスメモリDRAMは、図４で示したメモリと同等である。NAND型フラッシュメモリNANDは図１４で示したメモリと同等である。

【０２５５】

メモリモジュールMEM２５は、それを構成するメモリの接続の順番が、情報処理装置CPU_CHIPから近い順に、NOR型フラッシュメモリセルを利用したNOR型フラッシュメモリNOR、ダイナミックメモリセルを利用したダイナミックランダムアクセスメモリDRAM、NAND型フラッシュメモリセルを利用したNAND型フラッシュメモリNANDである。

【０２５６】

携帯電話機では、電話やメールの待ち受け時は、OSや通信用プログラムなどが格納されるNOR型フラッシュメモリNORへの間欠的なアクセスが支配的となる。したがって、不揮発性メモリであるNOR型フラッシュメモリNORを、情報処理装置CPU_CHIPから最も近くに接続する本実施形態では、ダイナミックランダムアクセスメモリDRAMをセルフリフレッシュ状態にし、さらにダイナミックランダムアクセスメモリDRAM やNAND型フラッシュメモリNANDへのリクエストクロック(RqCk1およびRqCk0)や、レスポンスクロック(RsCk1やRsCk2)を停止し、NOR型フラッシュメモリNORのみを動作させることができ、電話やメールの待ち受け時の消費電力を低減することができる。

【実施例４】

【０２５７】

図２６は、情報処理装置CPU_CHIPとメモリモジュールMEM２６から構成される情報処理システムを示したものである。メモリモジュールMEM２６は、ダイナミックランダムアクセスメモリDRAM、NOR型フラッシュメモリNOR、NAND型フラッシュメモリNAND０およびNAND１から構成される。情報処理装置CPU__CHIPは、図１で示したものと同等である。ダイナ

ミックランダムアクセスメモリDRAM0およびDRAM1は、図４で示したメモリと同等である。NAND型フラッシュメモリNAND０およびNAND1は図１４で示したメモリと同等である。NAND型フラッシュメモリNAND０およびNAND1はNOR型ラッシュメモリより大容量かつ低コストを実現できるメモリである。NOR型フラッシュメモリの代わりにNAND型フラッシュメモリNAND０を利用することでNAND型フラッシュメモリNAND０へ、OSやアプリケーションプログラムを格納でき、大容量かつ低コストの情報処理システムが実現できる。さらに、NAND型フラッシュメモリNAND０へ格納しているOSやアプリケーションプログラムをあらかじめダイナミックランダムアクセスメモリDRAMへ転送することで、情報処理システムの高性能化が図れる。

【実施例５】

【０２５８】

図２７は、情報処理装置CPU_CHIPとメモリモジュールMEM２７から構成される情報処理システムを示したものである。メモリモジュールMEM２７は、ダイナミックランダムアクセスメモリDRAM、NOR型フラッシュメモリNOR、NAND型フラッシュメモリおよびハードディスクHDDから構成される。情報処理装置CPU__CHIPは、図１で示したものと同等である。ダイナミックランダムアクセスメモリDRAM0およびDRAM1は、図４で示したメモリと同等である。NOR型フラッシュメモリNORは、図１２示したメモリと同等である。NAND型フラッシュメモリNANDは図１４で示したメモリと同等である。ハードディスクHDDは、NAND型フラッシュメモリNANDより大容量かつ低コストを実現できるメモリである。

【０２５９】

データの読み出し単位や、アドレス管理方法や、エラー検出訂正方法に関していえば、もともとハードディスクHDDで実現していたデータの読み出し単位や、アドレス管理方法や、エラー検出訂正方法などをフラッシュメモリが引き継いでいるため、ハードディスクHDDを容易に追加接続し、大容量かつ低コストのメモリモジュールを実現できる。

【実施例６】

【０２６０】

図２８は、情報処理装置CPU_CHIPとメモリモジュールMEM２８から構成される情報処理システムを示したものである。メモリモジュールMEM２８は、第１の不揮発性メモリMRAM、第２の不揮発性メモリNOR、第３の不揮発性メモリNANDから構成される。情報処理装置CPU_CHIPは、図１で示したものと同等である。第１の不揮発性メモリMRAMは、図４で示したメモリ回路MemVLが、不揮発性のマグネティックメモリセルで構成されているマグネティックランダムアクセスメモリMRAMである。第２の不揮発性メモリNORは、図12で示したNOR型フラッシュメモリと同等である。第３の不揮発性メモリNANDは、図１４で示したNAND型フラッシュメモリNANDと同等である。

【０２６１】

揮発性のダイナミックランダムアクセスメモリDRAMの代わりに不揮発性のマグネティックランダムアクセスメモリMRAMを用いることで、メモリ回路内のデータ保持動作を定期的に行う必要がないため、低電力化が可能となる。また、第２の不揮発性メモリM280は、図12で示したメモリ回路NV1が不揮発性の相変化メモリセルから構成される相変化メモリであっても良い。

【実施例７】

【０２６２】

図２９は本発明における第７の実施の形態例を示したものである。図２９（ａ）は上面図であり、図２９（ｂ）は上面図に示したＡ－Ａ’線に沿った部分の断面図である。

【０２６３】

本実施の形態のマルチチップ・モジュールは、ボールグリッドアレイ(BGA)によって装置に実装する基盤(例えばガラスエポキシ基板でできたプリント回路ボード)PCB上に、CHIPM1、CHIPM2、CHIPM3が搭載されている。特に限定しないが、CHIPM1は第１の不揮発性メモリで、CHIPM2は第２の不揮発性メモリで、CHIPM3は第１の揮発性メモリである。

【０２６４】

本マルチチップ・モジュールにより、図１で示すメモリモジュールMEMおよび、図２５で示すメモリモジュールMEM２５、図２６で示すメモリモジュールMEM２６、図２８で示すメモリモジュールMEM２８を１つの封止体に集積できる。

【０２６５】

CHIPM1と基盤PCB上のボンディングパットはボンディングワイヤ(PATH2)で接続され、CHIPM2と基盤PCB上のボンディングパットはボンディングワイヤ(PATH1)で接続されている。CHIPM3と基盤PCB上のボンディングパットはボンディングワイヤ(PATH4)で接続されている。CHIPM1とCHIPM2はボンディングワイヤ(PATH３)で接続され、CHIPM2とCHIPM3はボンディングワイヤ(PATH5)で接続されている。

【０２６６】

チップの搭載された基盤PCBの上面は樹脂モールドが行われて各チップと接続配線を保護する。なお、さらにその上から金属、セラミック、あるいは樹脂のカバー(COVER)を使用しても良い。

【０２６７】

本実施の形態例ではプリント回路ボードPCB上にベアチップを直接搭載するため、実装面積の小さなメモリモジュールを構成することができる。また、各チップを積層することができるため、チップと基盤PCB間の配線長を短くすることができ、実装面積を小さくすることができる。チップ間の配線及び各チップと基盤間の配線をボンディングワイヤ方式で統一することによって少ない工程数でメモリモジュールを製造することができる。

【０２６８】

さらにチップ間をボンディングワイヤで直接配線することによって基盤上のボンディングパット数とボンディングワイヤの本数を削減して少ない工程数でメモリモジュールを製造することができる。樹脂のカバーを使用した場合には、より強靭なメモリモジュールを構成することができる。セラミックや金属のカバーを使用した場合には、強度のほか、放熱性やシールド効果に優れたメモリモジュールを構成することができる。

【実施例８】

【０２６９】

図３０は本発明における第８の実施の形態例を示したものである。図３０（ａ）は上面図であり、図３０（ｂ）は上面図に示したＡ－Ａ’線に沿った部分の断面図である。

【０２７０】

本実施の形態のマルチチップ・モジュールは、ボールグリッドアレイ(BGA)によって装置に実装する基盤(例えばガラスエポキシ基板でできたプリント回路ボード)PCB上に、CHIPM1、CHIPM2、CHIPM3が搭載されている。CHIPM1は第１の不揮発性メモリ、CHIP2Mは第２の不揮発性メモリである。CHIP3Mはランダムアクセスメモリである。本マルチチップ・モジュールにより、図１で示すメモリモジュールMEMおよび、図２５で示すメモリモジュールMEM２５、図２６で示すメモリモジュールMEM２６、図２８で示すメモリモジュールMEM２８を１つの封止体に集積できる。

【０２７１】

CHIPM1と基盤PCB上のボンディングパットはボンディングワイヤ(PATH2)で接続され、CHIPM2と基盤PCB上のボンディングパットはボンディングワイヤ(PATH1)で接続されている。CHIPM1とCHIPM2はボンディングワイヤ(PATH３)で接続される。また、CHIP3Mの実装および配線にボールグリッドアレイが用いられている。

【０２７２】

本実装方法では３チップを積層することができるので実装面積を小さく保つことができる。さらに、CHIPM3と基盤間とのボンディングは不要となりボンディング配線の本数を削減することができるため組み立て工数を削減できる上、より信頼性の高いマルチチップモジュールが実現できる。

【実施例９】

【０２７３】

図３１は本発明に係るマルチチップ・モジュールの第９の実施の形態例を示したもので

ある。図３１（ａ）は上面図であり、図３１（ｂ）は上面図に示したＡ－Ａ’線に沿った部分の断面図である。

【０２７４】

本実施の形態のメモリモジュールは、ボールグリッドアレイ(BGA)によって装置に実装する基盤(例えばガラスエポキシ基板でできたプリント回路ボード)PCB上に、CHIPM1、CHIPM2、CHIPM3、CHIPM4が搭載されている。CHIPM1およびCHIPM2は不揮発性メモリ、CHIPM3はランダムアクセスメモリである。

【０２７５】

CHIPM4は情報処理装置CPU_CHIPはである。本実装方法では、図１で示す情報処理システムおよび、図２５で示す情報処理システム、図２６で示す情報処理システムおよび図２８で示す情報処理システムを１つの封止体に集積できる。

【０２７６】

CHIPM1と基盤PCB上のボンディングパットはボンディングワイヤ(PATH2)で接続され、CHIPM2と基盤PCB上のボンディングパットはボンディングワイヤ(PATH４)で接続され、CHIPM３と基盤PCB上のボンディングパットはボンディングワイヤ(PATH１)で接続されている。

【０２７７】

CHIPM1とCHIPM３はボンディングワイヤ(PATH３)で接続され、CHIPM2とCHIPM3はボンディングワイヤ(PATH５)で接続される。CHIPM4の実装および配線にボールグリッドアレイ（BGA）が用いられている。本実装方法ではプリント回路ボードPCB上にベアチップを直接搭載するため、実装面積の小さなメモリモジュールを構成することができる。また、各チップを近接して配置することができるため、チップ間配線長を短くすることができる。

【０２７８】

チップ間をボンディングワイヤで直接配線することによって基盤上のボンディングパット数とボンディングワイヤの本数を削減して少ない工程数でメモリモジュールを製造することができる。さらに、CHIPM４と基盤間とのボンディングは不要となりボンディング配線の本数を削減することができるため組み立て工数を削減できる上、より信頼性の高いマルチチップモジュールが実現できる。

【実施例１０】

【０２７９】

図３２は本発明に係るメモリシステムの第１０の実施の形態例を示したものである。図３２（ａ）は上面図であり、図３２（ｂ）は上面図に示したＡ－Ａ’線に沿った部分の断面図である。

【０２８０】

本実施の形態のメモリモジュールは、ボールグリッドアレイ(BGA)によって装置に実装する基盤(例えばガラスエポキシ基板でできたプリント回路ボード)PCB上に、CHIPM1、CHIPM2、CHIPM3が搭載されている。CHIPM1およびCHIPM2は不揮発性メモリ、CHIPM3はランダムアクセスメモリである。

【０２８１】

チップ間の配線及び各チップと基盤間の配線をボンディングワイヤ方式で統一することによって少ない工程数でメモリモジュールを製造することができる。本実装方法では、、図１で示すメモリモジュールMEMおよび、図２５で示すメモリモジュールMEM２５、図２６で示すメモリモジュールMEM２６、図２８で示すメモリモジュールMEM２８を１つの封止体に集積できる。

【０２８２】

CHIPM1と基盤PCB上のボンディングパットはボンディングワイヤ(PATH2)で接続され、CHIPM2と基盤PCB上のボンディングパットはボンディングワイヤ(PATH１)で接続され、CHIPM３と基盤PCB上のボンディングパットはボンディングワイヤ(PATH３)で接続されている。本実施の形態例ではプリント回路ボードPCB上にベアチップを直接搭載するため、実装面積の小さなメモリモジュールを構成することができる。また、各チップを近接して配置することができるため、チップ間配線長を短くすることができる。

【０２８３】
各チップと基盤間の配線をボンディングワイヤ方式で統一することによって少ない工程数でメモリモジュールを製造することができる。
【実施例１１】
【０２８４】
図３３は本発明に係るメモリシステムの第１１の実施の形態例を示したものである。図３２（ａ）は上面図であり、図３２（ｂ）は上面図に示したＡ－Ａ’線に沿った部分の断面図である。
【０２８５】
本実施の形態のメモリモジュールは、ボールグリッドアレイ(BGA)によって装置に実装する基板(例えばガラスエポキシ基板でできたプリント回路ボード)PCB上に、CHIPM1、CHIPM2、CHIPM3、CHIPM4が搭載されている。CHIPM1およびCHIPM2は不揮発性メモリ、およびCHIPM3はランダムアクセスメモリである。CHIPM4は情報処理装置CPU_CHIPはである。本実装方法では、図１で示す情報処理システムおよび、図２５で示す情報処理システム、図２６で示す情報処理システムおよび図２８で示す情報処理システムを１つの封止体に集積できる。
【０２８６】
CHIPM1と基盤PCB上のボンディングパットはボンディングワイヤ(PATH2)で接続され、CHIPM2と基盤PCB上のボンディングパットはボンディングワイヤ(PATH１)で接続され、CHIPM３と基盤PCB上のボンディングパットはボンディングワイヤ(PATH３)で接続されている。CHIPM4の実装および配線にボールグリッドアレイ（BGA）が用いられている。
【０２８７】
本実施の形態例ではプリント回路ボードPCB上にベアチップを直接搭載するため、実装面積の小さなメモリモジュールを構成することができる。また、各チップを近接して配置することができるため、チップ間配線長を短くすることができる。CHIPM４と基盤間とのボンディングは不要となりボンディング配線の本数を削減することができるため組み立て工数を削減できる上、より信頼性の高いマルチチップモジュールが実現できる。
【実施例１２】
【０２８８】
図３４に、本発明に係るメモリモジュールを利用した携帯電話機の第１２の実施の形態例を示す。携帯電話は、アンテナANT、無線ブロックRF、音声コーデックブロックSP、スピーカーSK、マイクロホンMK、情報処理装置CPU、液晶表示部LCD、キーボードKEYおよび本発明のメモリモジュールMSMで構成される。情報処理装置CPU__MAINは複数の情報処理回路を持ち、その中の１つの情報処理回路CPU0はベースバンド処理回路BBとして、他の中の少なくとも１つの情報処理回路CPU1はアプリケーションプロセッサAPとして動作する。
【０２８９】
通話時の動作を説明する。アンテナANTを通って受信された音声は無線ブロックRFで増幅され、情報処理装置CPU0へ入力される。情報処理装置CPU0では、音声のアナログ信号をデジタル信号に変換し、エラー訂正と復号処理おこない、音声コーデックブロックSPへ出力する。音声コーデックブロックがデジタル信号をアナログ信号に変換しスピーカーSKに出力すると、スピーカーから相手の声が聞こえる。
【０２９０】
携帯電話機から、インターネットのホームページにアクセスし、音楽データをダウンロードし、再生して聞き、最後にダウンロードした音楽データを保存するという一連の作業を行うときの動作を説明する。
【０２９１】
メモリモジュールMEMには、OS、アプリケーションプログラム（メール、Webブラウザ、音楽再生プログラム、動作再生プログラム、ゲームプログラムなど）、音楽データ、静止画データ、動画データなどが格納されている。
【０２９２】

キーボードより、Webブラウザの起動を指示すると、メモリモジュールMSM内のNOR型フラッシュメモリへ格納されているWebブラウザのプログラムは情報処理回路CPU1によって読み出され、実行され、液晶表示LCDにWebブラウザが表示される。所望のホームページにアクセスし、気に入った音楽データのダウンロードをキーボードKEYより指示すると、音楽データは、アンテナANTを通って受信され、無線ブロックRFで増幅され、情報処理装置CPU0へ入力される。情報処理装置CPU0では、アナログ信号である音楽データをデジタル信号に変換し、エラー訂正と復号処理おこなう。デジタル信号化された音楽データはメモリモジュールMSM内のダイナミックランダムアクセスメモリDRAMへ一旦、保持され、最終的に、メモリモジュールMEMのNAND型フラッシュメモリへと転送され格納される。

【０２９３】

次に、キーボードKEYより、音楽再生プログラムの起動を指示するとメモリモジュールMSM内のNOR型フラッシュメモリへ格納されている音楽再生プログラムが、情報処理回路CPU1によって読み出され、実行され、液晶表示LCDに音楽再生プログラムが表示される。

【０２９４】

キーボードKEYより、メモリモジュール内NAND型フラッシュメモリへダウンロードした音楽データを聞くための指示を行うと、情報処理回路CPU１は音楽再生プログラムを実行し、NAND型フラッシュメモリへ保持している音楽データを処理し、最終的にスピーカーSKから音楽が聞こてくる。本発明のメモリモジュールMSM内のNOR型フラッシュメモリには、Webブラウザと音楽再生プログラムや、電子メールプログラムなどの複数のプログラムが格納され、情報処理装置CPU_MAINは複数の情報処理回路CPU0からCPU3を持つため、同時に複数のプログラムを実行することができる。

【０２９５】

電話や電子メールの待ちうけ時には、情報処理装置CPU_MAINは、メモリモジュールMSMへのクロックを必要最小限の周波数で動作させることができ消費電力を極端に小さくできる。

【０２９６】

このように、本発明に係るメモリモジュールを用いることにより、大量のメール、音楽再生、アプリケ-ションプログラムや音楽データ、静止画像データ、動画データなどを格納でき、さらに複数のプログラムを同時に実行できる。

【実施例１３】

【０２９７】

図３５に、本発明に係るメモリシステムを利用した携帯電話機の第１３の実施の形態例を示す。携帯電話は、アンテナANT、無線ブロックRF、音声コーデックブロックSP、スピーカーSK、マイクロホンMK、液晶表示部LCD、キーボードKEYおよび、メモリモジュールMSMと情報処理装置CPU_MAINを１つの封止体に集積した本発明の情報処理システムSLPで構成される。

【０２９８】

本発明の情報処理システムSLPを用いることによって、部品点数を削減できるため、低コスト化ができ、携帯電話の信頼性が向上する、携帯電話機を構成する部品の実装面積を小さくでき、携帯電話小型化ができる。

【０２９９】

＜実施例に示される発明の効果のまとめ＞

以上説明したように本明細書に開示される発明によって得られる主な効果は以下の通りである。

【０３００】

第１に、電源投入直後に、直列接続の確認動作を行うことで、確実にメモリ同士が接続されていることが確認できる。さらに、ブートデバイスおよび、最端のメモリチップを明示し、自動的に各メモリへのID付けが行われることで、容易に、必要な分だけメモリチップを接続し、メモリ容量を拡張することができる。

【０３０１】

第２に、リクエストへIDを付加することで、情報処理装置CPU_CHIPから各メモリチップM0、M1およびM2へリクエストが確実に転送される。また、情報処理装置CPU_CHIPへのレスポンスへIDを付加することで、各メモリから正しく正しくデータ転送が行えたことを確認でき、情報処理装置CPU_CHIPおよびメモリチップM0、M1、M2の直列接続によって、接続信号数を減少させながらも、情報処理装置CPU_CHIPは所望の処理を実行することができる。

【０３０２】

第３に、リクエストインターフェース回路ReqIFとレスポンスインターフェース回路は独立に動作可能なため、データの読み出し動作と書き込み動作を同時に実行でき、データ転送性能を向上させることができる。

【０３０３】

第４に、リクエストの入力順序に関わらず、早く読み出せるデータは、読み出しが遅いデータを待つことなく、すぐに読み出すことができるため、高速化が可能となる。さらに、リクエストへIDを付加することで、確実に要求先へリクエストが転送され、また、レスポンスへIDを付加することで、リクエストの入力順序と、読み出しデータの順番が異なった場合でも、情報処理装置CPU__CHIPは転送元のメモリチップを知ることができる。

【０３０４】

第５に、情報処理装置への各メモリからのレスポンス順序は、読み出した回数に応じて動的に変化するため、データ転送性能を向上することができる。さらに、読み出し回数は、プログラムすることができ、利用するシステムに柔軟に対応することができる。

【０３０５】

第６に、メモリチップから情報処理処理装置へエラーを送信することができるので、情報処理装置はエラーを検出して、すぐにエラーに対処することができ、信頼性の高い情報処理システムを構築できる。

【０３０６】

第７に、各メモリチップM0、M1およびM2のクロックの動作周波数を必要に応じて、変更することができ低電力化を図ることができる。

【０３０７】

第８に、メモリチップM2からの読み出し時は、エラー検出と訂正を行い、書きこみ時は、書きこみが正しく行われなかった不良アドレスに対して代替処理を行うため、信頼性を保つことができる。

【０３０８】

第９に、複数の半導体チップを一つの封止体に実装することによって実装面積の小さなシステムメモリ・モジュールを提供できる。

【図面の簡単な説明】

【０３０９】

【図１】本発明を適用した情報処理システムの構成の一例を示す構成図である。

【図２】本発明を適用した情報処理システムのアドレスマップの一例を示す説明図である。

【図３】本発明を適用した情報処理システムの電源投入時の動作の一例を示す図である。

【図４】本発明を適用した情報処理システムを構成するメモリの構成の一例を示す図である。

【図５】本発明を適用した情報処理システム内で発生したリクエストに対する動作の一例を示すフローチャートである。

【図６】本発明を適用した情報処理システムでのレスポンスに対する動作の一例を示すフローチャートである。

【図７】本発明を適用した情報処理システムでのレスポンスに対する動作の一例を示すフローチャートである。

【図８】レスポンススケジュール回路SCHの動作を示すフローチャートである。

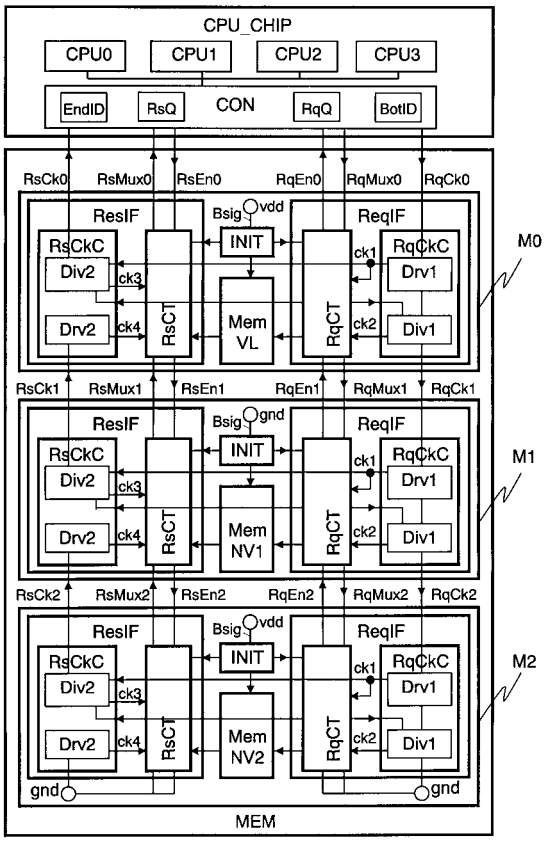
【図９】レスポンススケジュール回路SCHのレスポンス優先順位の変更動作の一例を示す図である。

【図１０】本発明を適用した情報処理システムのクロック制御動作の一例を示すフローチャートである。
【図１１】本発明を適用した情報処理システムを構成するメモリのメモリ回路の構成の一例を示す図である。
【図１２】本発明を適用した情報処理システムを構成するメモリの構成の一例を示す図である。
【図１３】レスポンススケジュール回路SCHのレスポンス優先順位の変更動作の一例を示す図である。
【図１４】本発明を適用した情報処理システムを構成するメモリの構成の一例を示す図である。
【図１５】レスポンススケジュール回路SCHのレスポンス優先順位の変更動作の一例を示す図である。
【図１６】本発明を適用した情報処理システムでのエラーレスポンスに対する動作の一例を示すフローチャートである。
【図１７】本発明を適用した情報処理システムでの動作波形の一例を示す図である。
【図１８】本発明を適用した情報処理システムでの動作波形の一例を示す図である。
【図１９】本発明を適用した情報処理システムでの動作波形の一例を示す図である。
【図２０】本発明を適用した情報処理システムでの動作波形の一例を示す図である。
【図２１】本発明を適用した情報処理システムでの動作波形の一例を示す図である。
【図２２】本発明を適用した情報処理システムでの動作波形の一例を示す図である。
【図２３】本発明を適用した情報処理システムでの動作波形の一例を示す図である。
【図２４】本発明を適用した情報処理システムの構成図である。
【図２５】本発明を適用した情報処理システムの構成図である。
【図２６】本発明を適用した情報処理システムの構成図である。
【図２７】本発明を適用した情報処理システムの構成図である。
【図２８】本発明を適用した情報処理システムの構成図である。
【図２９】本発明によるメモリ情報処理システムの実装形態の一例を示す図である。
【図３０】本発明によるメモリ情報処理システムの実装形態の一例を示す図である。
【図３１】本発明によるメモリ情報処理システムの実装形態の一例を示す図である。
【図３２】本発明によるメモリ情報処理システムの実装形態の一例を示す図である。
【図３３】本発明によるメモリ情報処理システムの実装形態の一例を示す図である。
【図３４】本発明によるメモリ情報処理システムを利用した携帯電話の構成例を示すブロック図である。
【図３５】本発明によるメモリ情報処理システムを利用した携帯電話の構成例を示すブロック図である。
【図３６】携帯電話に利用されている従来のメモリ構成例を示すブロック図である。
【符号の説明】
【０３１０】
CPU_CHIP・・・情報処理装置、CPU0、CPU1、CPU2、CPU3・・・情報処理回路、CON・・・メモリ制御回路、リクエストキューRqQ・・・リクエストキュー、RsQ・・・レスポンスキュー、BotID・・・ブートデバイスIDレジスタ、EndID・・・最端デバイスIDレジスタ、MEM・・・メモリモジュール、M0、M1、M2・・・メモリチップ、INIT・・・初期設定回路、ReqIF・・・リクエストインターフェース回路、ResIF・・・レスポンスインターフェース回路、MemVL 、MemNV1 、MemNV２・・・メモリ回路、ReqIF・・・リクエストインターフェース回路、RqCkC・・・リクエストクロック制御回路、RqCT・・・リクエストキュー制御回路、dstID・・・IDレジスタ、Bsig・・・ブートデバイス認識信号、RqCk0、RqCK1、RqCk2・・・リクエストクロック、RsCk0、RsCK1、RsCk2・・・レスポンスクロック、RqEN0、RqEN1、RqEN2・・・リクエストイネーブル信号、RsEN0、RsEN1、RsEN2・・・レスポンスイネーブル信号、RqMux0、RqMux1、RqMux2・・・リクエスト信号、RsMux0、RsMux1、RsMux2・・・レスポンス信号、ck1、ck2、ck3、ck4・・・クロック信号、BotID-AREA・・

・ブートデバイスID格納領域、EndID-AREA・・・最終端デバイスID格納領域、InitPR-AREA・・・初期プログラム領域、OSAP-AREA・・・プログラム格納領域、COPY-AREA・・・コピー領域、WORK-AREA・・・ワーク領域、DATA-AREA・・・データ領域、REP-AREA・・・代替領域、PwOn…電源投入期間、RESET…リセット期間、BootIDSet…ブートデバイスID設定期間、LinkEn・・・接続確認期間、BootRD・・・ブートデータ読み出し期間、InitID・・・ID番号設定期間、Idle・・・アイドル期間、RqQI、RqQXI、RqQXO・・・リクエストキュー回路、dstID・・・IDレジスタ回路、CPQ・・・ID比較回路、RsQo、RsQp・・・レスポンスキュー回路、STReg・・・ステータスレジスタ回路、SCH・・・レスポンススケジュール回路、CmdDec・・・コマンドデコーダ、ContLogic・・・制御回路、RaddLat・・・ロウアドレスバッファ、CaddLat・・・カラムアドレスバッファ、RefC・・・リフレッシュカウンタ、Thmo・・・温度計、WdataLat・・・ライトデータバッファ、RdataLat・・・リードデータバッファ、RowDec・・・ロウデコーダ、ColDec・・・カラムデコーダ、SenseAmp・・・センスアンプ、DataCont・・・データ制御回路、Bank0、Bank1、Bank2、Bank3、Bank4、Bank5、Bank6、Bank7、・・・メモリバンク、BotID・・・ブートデバイスID値、EndID・・・終端デバイスID値DRAM,DRAM0、DRAM1・・・ダイナミックランダムアクセスメモリ、NOR・・・NOR型フラッシュメモリ、NAND、NAND0、NAND1・・・NAND型フラッシュメモリ、HDD・・・ハードディスク、MRAM・・・マグネティックランダムアクセスメモリ、CHIPM1、CHIPM2、CHIP3M、CHIP4M…半導体チップ、PCB…プリント回路基板、COVER…モジュールの封止カバー、PATH1～PATH5…ボンディング配線、ANT…アンテナ、RF…無線ブロック、SP…音声コーデックブロック、SK…スピーカー、MK…マイクロホン、CPU…プロセッサ、DRAM…ダイナミックランダムアクセスメモリ、LCD…液晶表示部、KEY…キーボード、MSM…メモリモジュール、CPU_MAIN・・・情報処理装置、SLP…情報処理装置CPU_MAINとメモリモジュールMSMとを、１つの封止体に集積したモジュール、PRC…情報処理装置、MCM1、MCM2…メモリモジュール、CPU…中央演算装置、SRC、DRAC、NDC…メモリコントローラ、NOR FLASH…NOR型フラッシュメモリ、SRAM…スタティックランダムアクセスメモリ、NAND FLASH…NAND型フラッシュメモリ、DRAM…ダイナミックランダムアクセスメモリ。

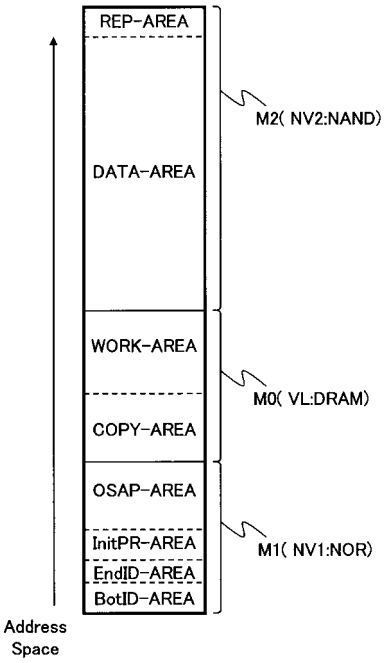
【図１】

図1



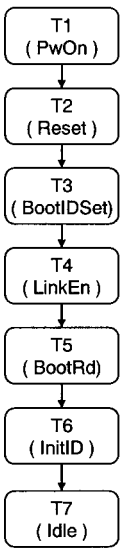
【図２】

図2



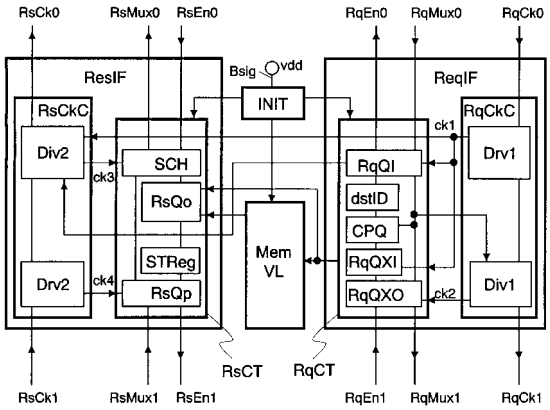
【図３】

図3



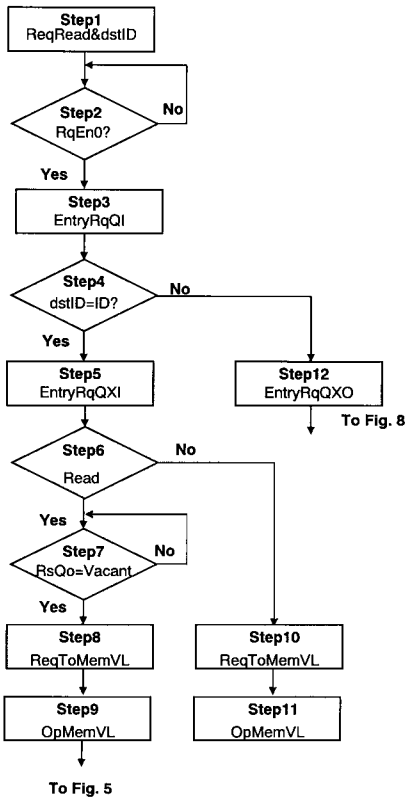
【図４】

図4



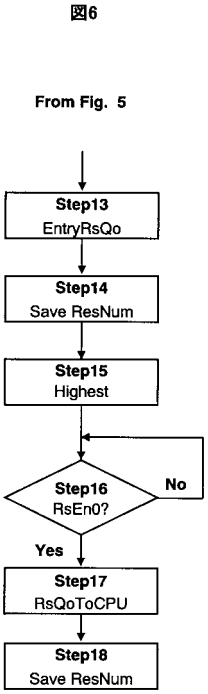
【図５】

図5



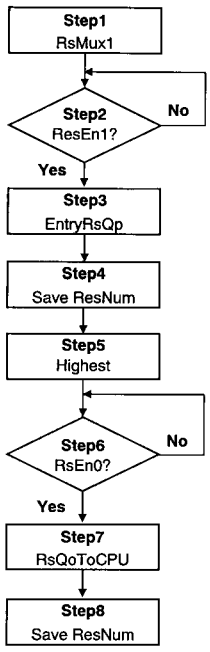
【図６】

図6



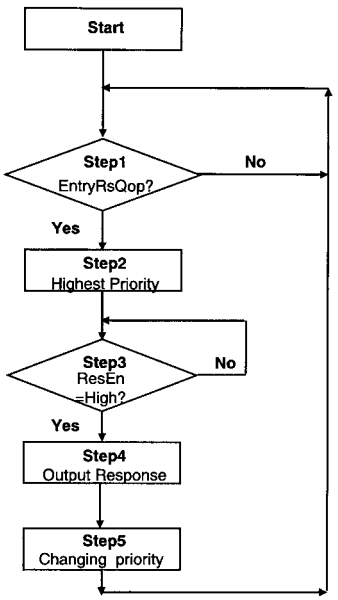
【図７】

図7



【図８】

図8



【図９】

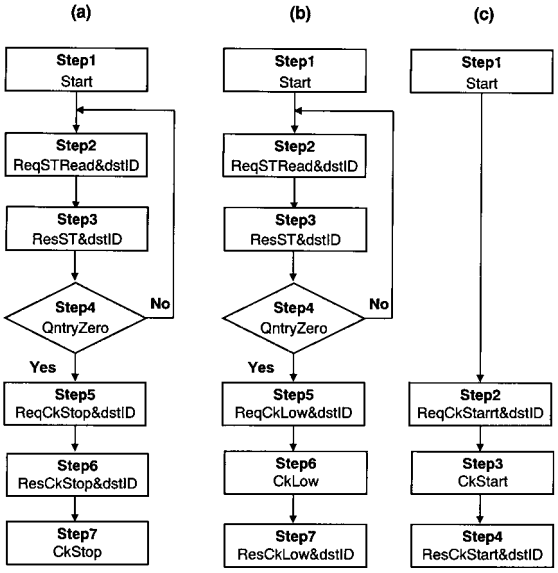
図9

(a) M0 priority control

Response Priority	Initial	Number of Response: RsQO(M0) N time →	RsQP(M1) M time →	RsQP(M2) L time
PRsQo(M0)	1	3	2	1
PRsQp(M1)	2	1	3	2
PRsQp(M2)	3	2	1	3

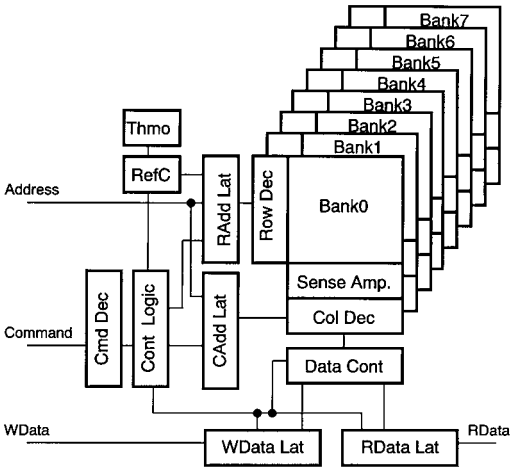
【図１０】

図10



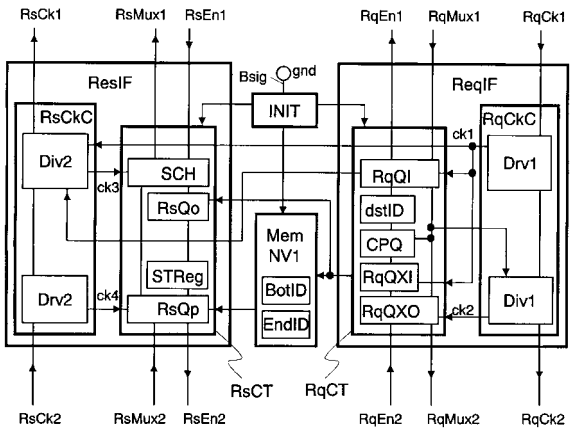
【図１１】

図11



【図１２】

図12



【図１３】

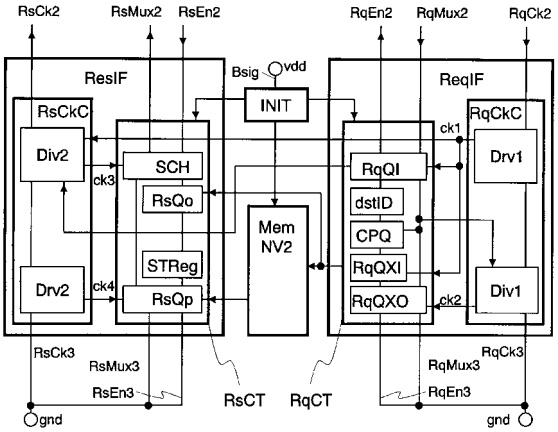
図13

(b) M1 priority control

Response Priority	Initial	Number of Response: RsQO(M0) → RsQO(M1) → RsQP(M2)
		- → M1 time → L1 time
PRsQo(M1)	1	- → 2 → 1
PRsQp(M2)	2	- → 1 → 2

【図１４】

図14



【図１５】

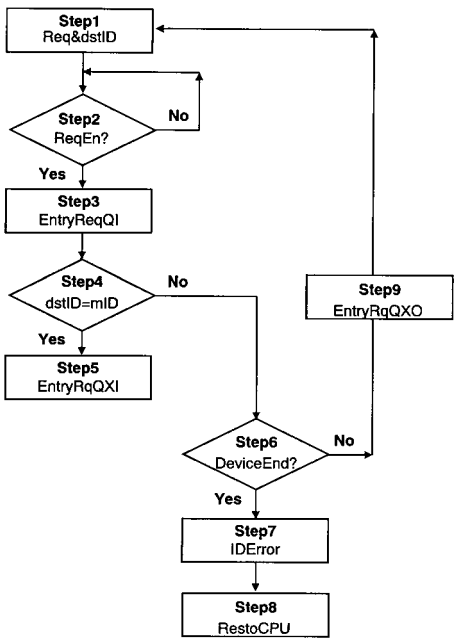
図15

(c) M2 priority control

Response Priority	Initial	Number of Response: RsQO(M0) → RsQO(M1) → RsQO(M2)
		- → - → L2time(0)
PRsQo(M2)	1	- → - → 1

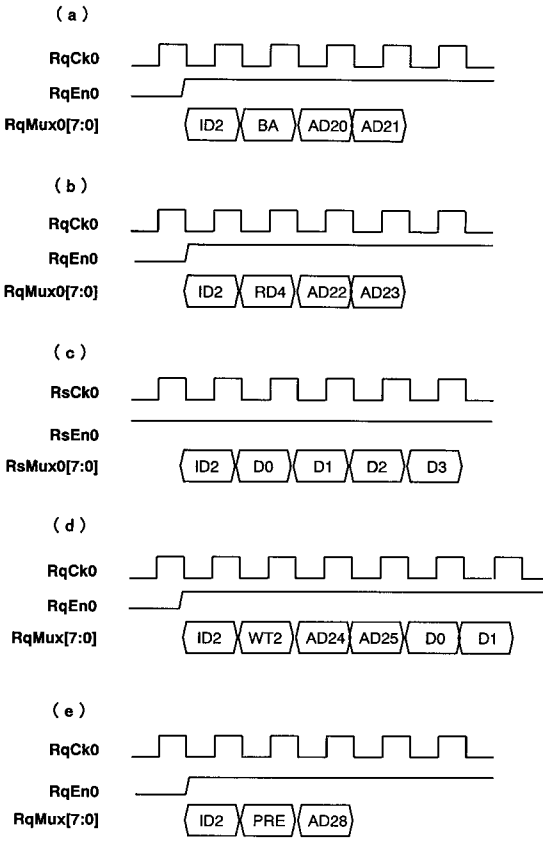
【図１６】

図16



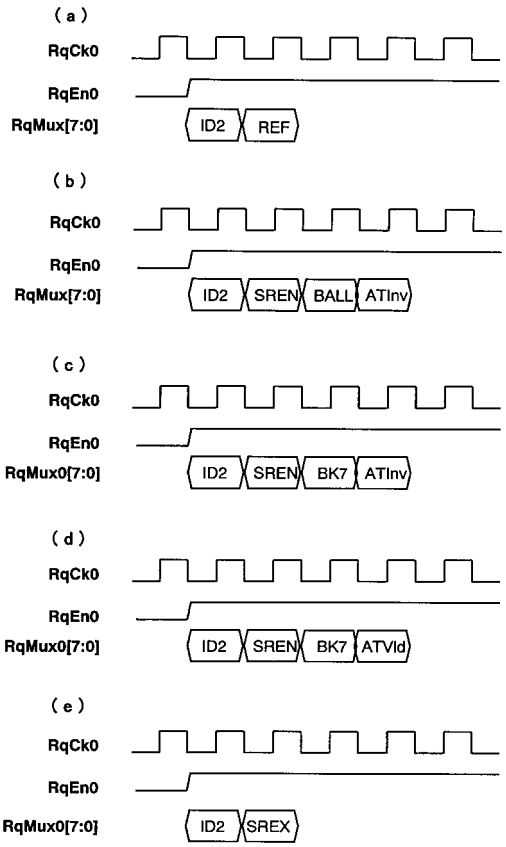
【図１７】

図17



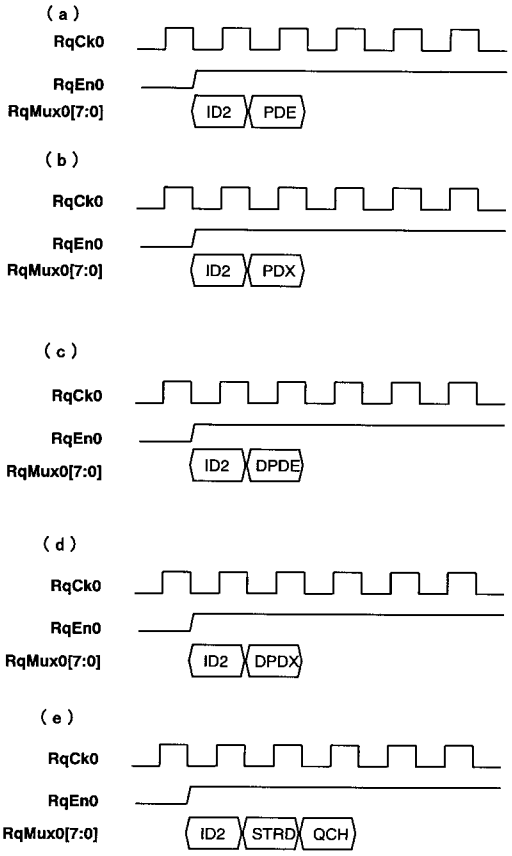
【図１８】

図18



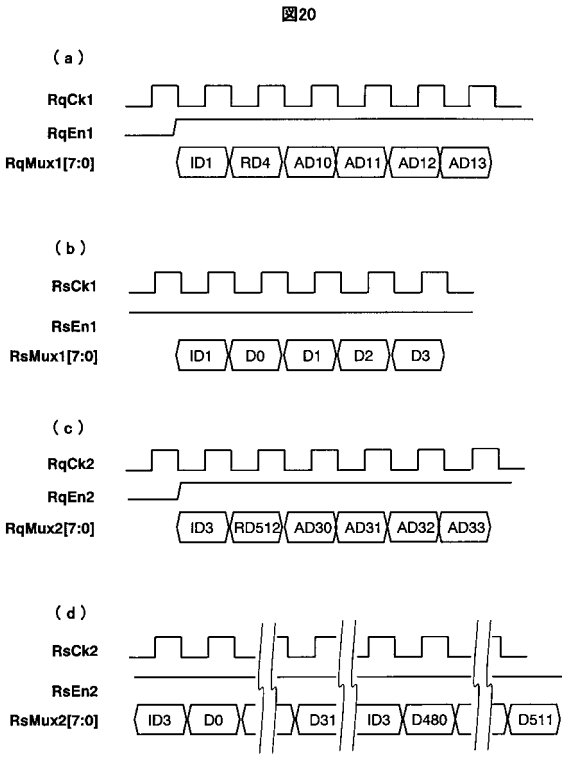
【図１９】

図19



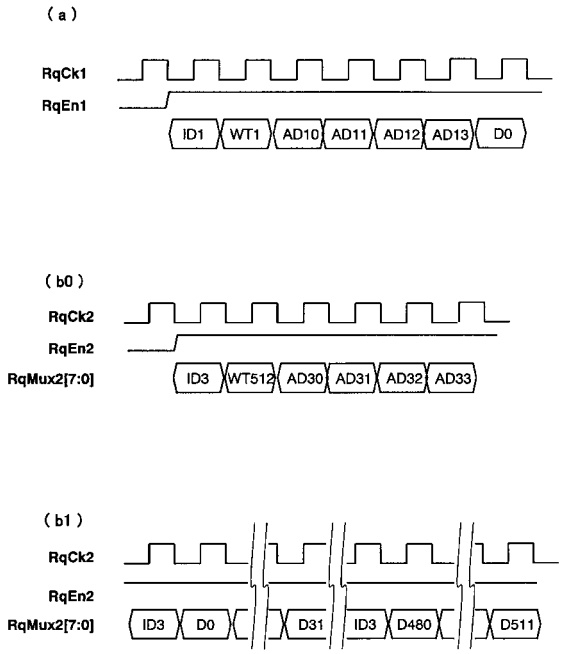
【図２０】

図20



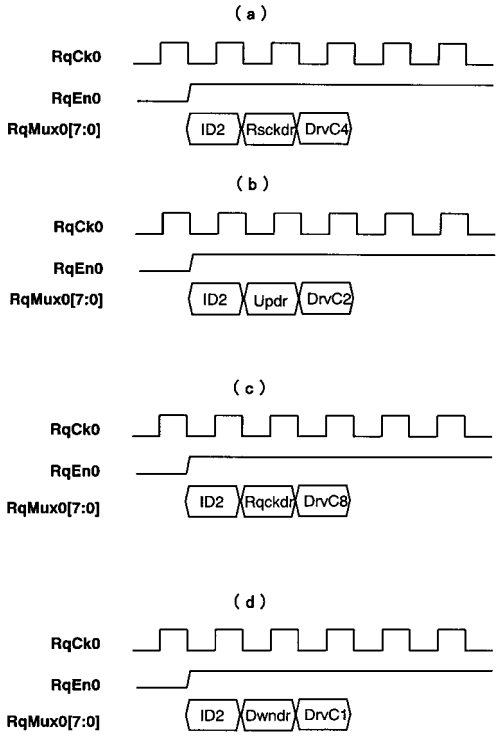
【図２１】

図21



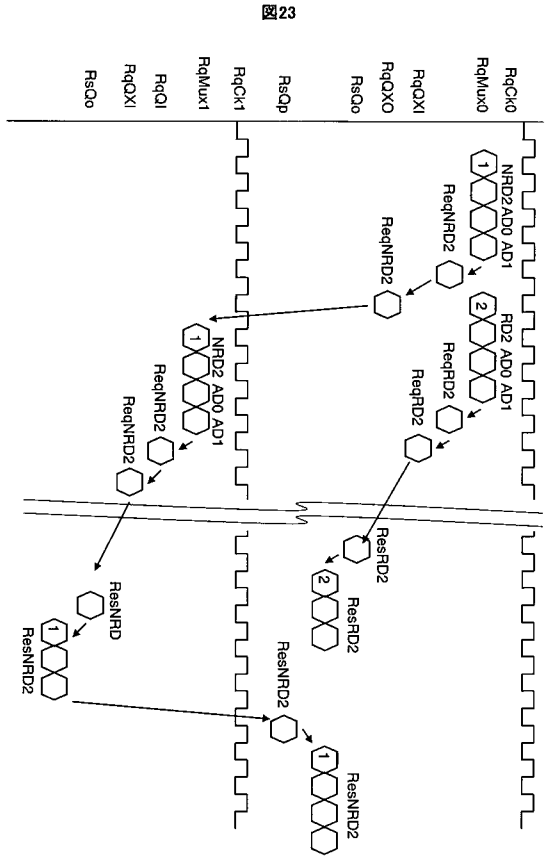
【図２２】

図22



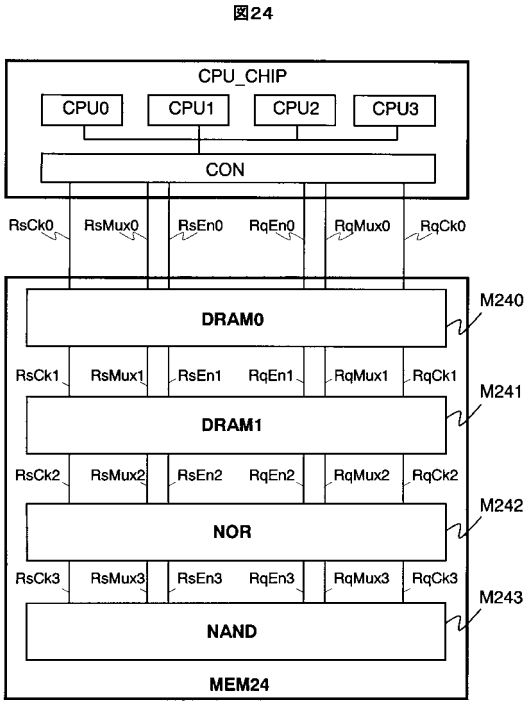
【図２３】

図23



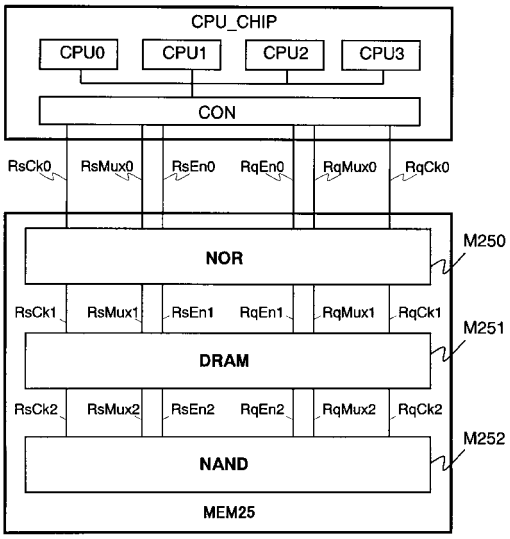
【図２４】

図24



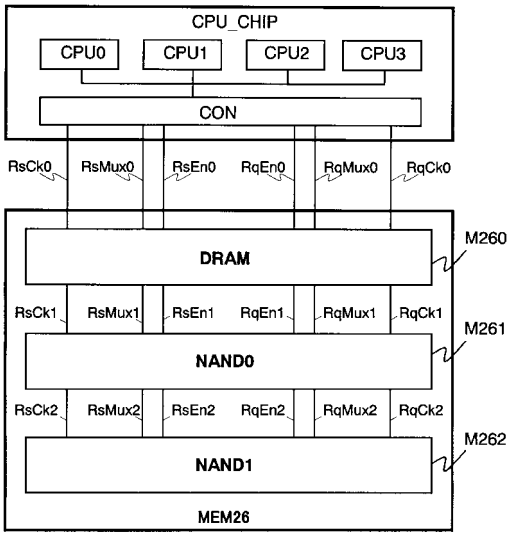
【図２５】

図25



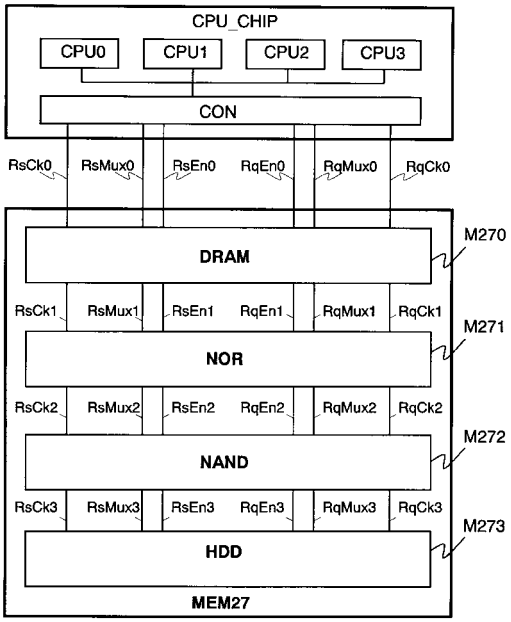
【図２６】

図26



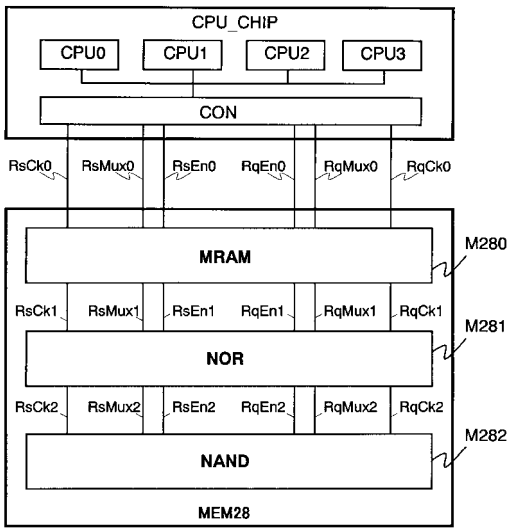
【図２７】

図27



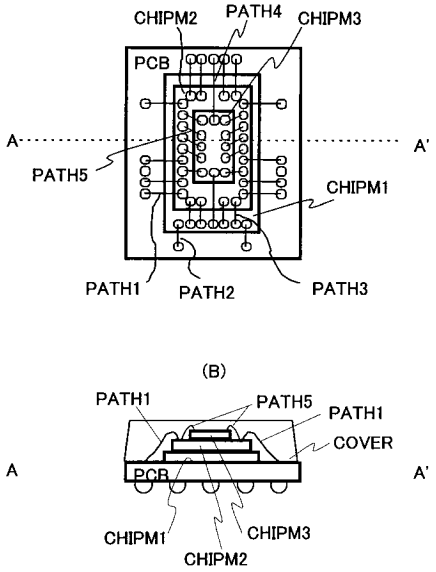
【図２８】

図28



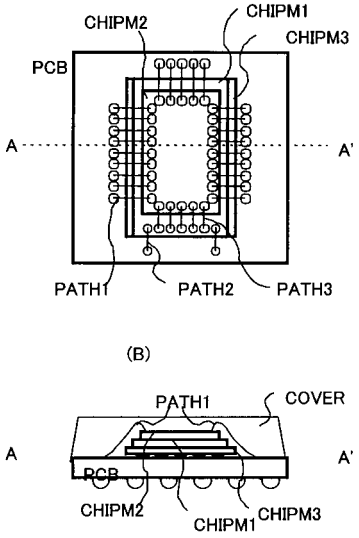
【図２９】

図29



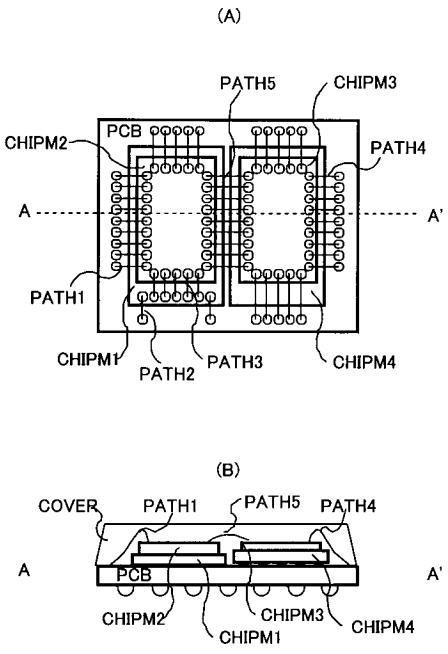
【図３０】

図30



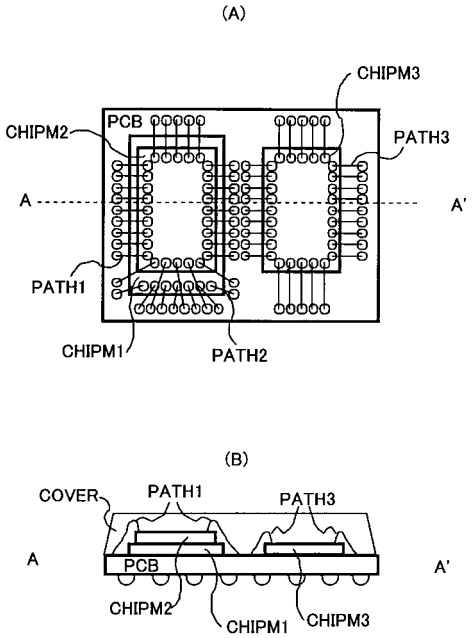
【図３１】

図31



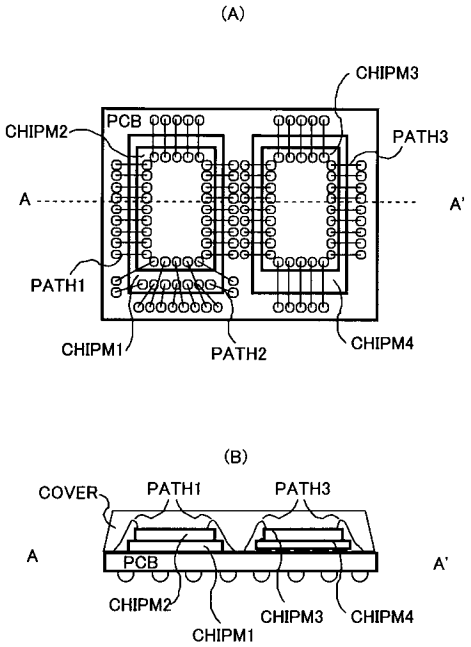
【図３２】

図32



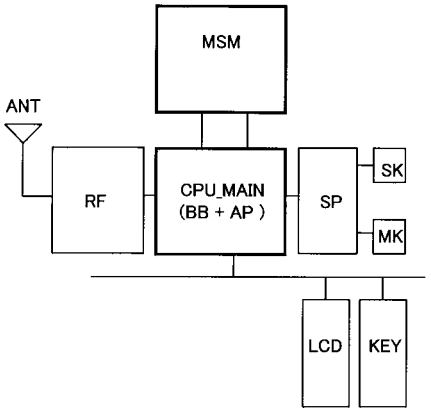
【図３３】

図33



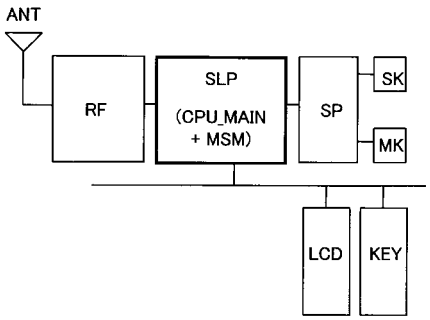
【図３４】

図34



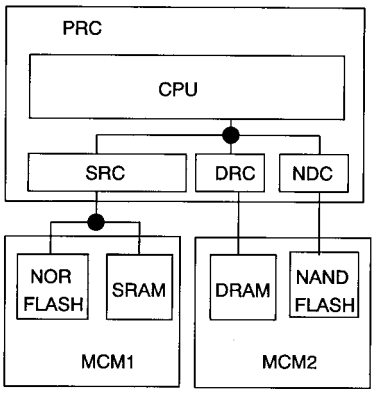
【図３５】

図35



【図３６】

図36



フロントページの続き

(72)発明者　原口　嘉典
東京都中央区八重洲二丁目２番１号　エルピーダメモリ株式会社内

審査官　吉田　誠

(56)参考文献　特開２００４－１３９５５２（ＪＰ，Ａ）
国際公開第２００５／１１９４５７（WO，Ａ１）
国際公開第２００５／１１４６６９（WO，Ａ１）
国際公開第２００５／０１７９０３（WO，Ａ１）
特開２００２－００７３０８（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１２／０６
Ｇ０６Ｆ　１３／１６