

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4472186号
(P4472186)

(45) 発行日 平成22年6月2日(2010.6.2)

(24) 登録日 平成22年3月12日(2010.3.12)

(51) Int. Cl.	F I		
G06F 12/06 (2006.01)	G06F	12/06	550A
G06F 12/00 (2006.01)	G06F	12/00	571B
G06F 3/06 (2006.01)	G06F	3/06	301E

請求項の数 2 (全 40 頁)

(21) 出願番号	特願2000-582897 (P2000-582897)	(73) 特許権者	501195636
(86) (22) 出願日	平成11年11月15日 (1999.11.15)		インフィネオン・テクノロジーズ・アーゲー
(65) 公表番号	特表2002-530742 (P2002-530742A)		—
(43) 公表日	平成14年9月17日 (2002.9.17)		INFINEON TECHNOLOGIES AG
(86) 国際出願番号	PCT/US1999/027017		ドイツ連邦共和国 デー-81541 ミュンヘン, ザンクト・マルティン・シュトラッセ, 76
(87) 国際公開番号	W02000/029956	(74) 代理人	110000028
(87) 国際公開日	平成12年5月25日 (2000.5.25)		特許業務法人明成国際特許事務所
審査請求日	平成18年10月25日 (2006.10.25)	(72) 発明者	ストラコブスキー・ヘンリー
(31) 優先権主張番号	60/108,930		アメリカ合衆国 カリフォルニア州95118 サン・ホセ, チェリー・アベニュー, 3962
(32) 優先日	平成10年11月16日 (1998.11.16)		
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	09/439,860		
(32) 優先日	平成11年11月12日 (1999.11.12)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 外部デバイスへのアクセスを優先順序付けるための方法および装置

(57) 【特許請求の範囲】

【請求項1】

外部デバイスへのアクセスを優先順序付けるための装置であって、
要求元デバイスから前記外部デバイスへの任意数の要求を格納するように適切に構成された要求キューと、

前記要求キューに結合され、そこに格納された任意の要求を取り出すように適切に構成された要求キューコントローラユニットと、

前記外部デバイスからの任意数の応答を格納するように適切に構成された応答キューと、

前記応答キューに結合され、そこに格納された任意の要求を取り出すように適切に構成された応答キューコントローラユニットと

を備え、

前記応答およびそれに関連付けられた前記要求のそれぞれは、前記要求の発信元であり対応する前記応答の送信先である特定グループの要求デバイスを示すグループ識別番号に関連付けられており、

前記応答キューコントローラおよび前記要求キューコントローラユニットは、グループ優先順序セレクトレジスタに格納された優先番号を使用し、前記格納された要求および応答をそれぞれ優先順序付けることによって、優先順序の高い要求または応答に優先順序の低い要求または応答をバイパスさせる、装置。

【請求項2】

請求項1に記載の装置であって、さらに、

10

20

優先順序の高い応答または要求が優先順序の低い応答または要求をバイパスする状態でアクティブにされることによって、動ロック条件を実質的に回避するように構成された動ロックカウンタレジスタを備える、装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、一般に、コンピュータシステムに関する。本発明は、特に、マルチプロセッサコンピュータシステム等のコンピュータシステムにおいて、共用リソースへのアクセスを提供することに関する。さらに特定すると、外部デバイスへのアクセスを優先順序付けるための方法および装置に関する。

10

【0002】

【従来の技術】

基本的なコンピュータシステムにおいては、中央処理機構、またはCPUが、関連するメモリ内に記憶されている所定のプログラムまたは命令一式に従って動作する。そのような、プロセッサの動作を規定する、記憶されている命令一式またはプログラム以外にも、処理中における中央プロセッサの情報操作を容易にするために、メモリ空間がプロセッサメモリまたは関連する追加メモリ内に備えられている。追加メモリは、プロセッサによって作成される情報の記憶場所を提供し、加えて、プログラムの処理に際してプロセッサが臨時に、すなわち“メモ帳”代わりにして使用する情報の記憶場所を提供する。さらに、関連メモリは命令一式を実行しているプロセッサの出力情報を設置する場所を提供し、システムの出力装置がその情報を利用できるようにする。

20

【0003】

現存するメモリにアクセスするために、多くの構成部分（プロセッサ、ハードドライブなど）が1つの共通のバスを共同使用しなければならないシステムにおいては、メモリへのアクセスをめぐる衝突が発生する可能性が上昇する。特にマルチプロセッサコンピュータシステムなどの場合、異なるプロセッサを利用するシステムが同時に操作されるため、メモリまたは他の共用リソースへのアクセスが複雑になる。各プロセッサまたはプロセッサシステムが同じメモリへのアクセスを同時に要求する可能性が高いので、プロセッサ間での衝突は一般的には回避不可能である。基本的に、マルチプロセッサコンピュータシステムにおける2つ以上のプロセッサまたはプロセッサシステムの動作は、結果として、共用メモリ或いは他の共用リソースに対するメモリコマンドの断続的な重複を生じさせる。

30

【0004】

共用メモリに対するメモリアクセス要求の衝突という問題の解決のために採られてきた従来の方策は、ある場合では、プロセッサのそれぞれに使用されるメモリの完全な重複や、プロセッサシステムの隔離が含まれる。しかし、メモリアクセス要求の衝突問題を解決するためのこの方策は、しばしば、多重プロセッサシステムにおいて意図された利点を無にしてしまう。そのような多重プロセッサは、一方のプロセッサが他方のプロセッサの動作を補助しながら、同じデータでの並行した算出動作を行うといった状況で操作される場合に最も効果的である。従来、そのようなプロセッサシステムは、プロセッサ同士がメモリなどの共用リソースへのアクセスを競い合うという時間的共用であるか、またはプロセッサシステムが二重のポートを有しており、各プロセッサは例えば個別にメモリバスを有し、一方がアクセスを許可された場合には他方は待機しているといった状態である。

40

【0005】

上記の衝突問題を回避するため様々な方策が採られてきた。ある方策では、各プロセッサの逐次的な使用、またはプロセッサの時間的共用によって衝突の回避が達成される。この方法では、単純に、衝突を避けるために、プロセッサが順番に共用リソースへのアクセスを行う。一般的に利用されるこのようなシステムは“リングパッシング”または“トークンシステム”を含み、それにより、ユーザーグループでのリングの伝達と類似する所定の連続動作に従って、衝突する可能性のあるプロセッサがシステムによってポーリングされる。

50

【 0 0 0 6 】

残念ながら、プロセッサによる逐次的なアクセスの方法論は、コンピュータシステムの全体的な動作に著しい制限を課すことになる。この制限は、システムが、衝突するプロセッサをポーリングするためかなりの時間を費やすという事実起因する。さらに、単独のプロセッサが作動しており、例えば共用メモリへのアクセスを要求する場合には、システムによる連続動作の実行に伴い、共用リソースへのプロセッサアクセスの間で、各メモリサイクル毎に遅延が生じる。

【 0 0 0 7 】

衝突回避のための他の一般的な方策は、コンピュータシステム内のプロセッサ間での優先順位付けによるものである。そのような方法では、各プロセッサに対してシステムの重要性の階層に従った優先順位が付される。衝突が発生するたびに、メモリコントローラは単純により高い優先順位を有するプロセッサにアクセスを提供する。例えば、2つのプロセッサを有するシステムの場合、共用メモリへは第一のプロセッサ、第二のプロセッサがアクセスする。共用メモリは、典型的には、メモリに維持記憶されているデータの周期的なリフレッシュが必要となるダイナミックDRAM(DRAM)型メモリ装置などである。一般的に、DRAM型メモリでは別の独立したリフレッシュシステムによってリフレッシュが行われる。そのようなマルチプロセッサシステムの場合、プロセッサとリフレッシュシステムとの双方が共用メモリへのアクセスを競うことになり、プロセッサおよびリフレッシュシステムに割り当てられた優先順位に従って、システムメモリコントローラがメモリアクセス要求の衝突やコマンドを処理する。そのようなシステムは、衝突の問題を解決し、さらに、単純な逐次的アクセスによる衝突回避のシステムよりも効果的である反面、依然として柔軟性に欠くものである。

【 0 0 0 8 】

衝突解消のための他の従来的な方策は、メモリコントローラ内に組み込まれた意思決定機能である。残念ながら、メモリコントローラの意思決定部分はクロックシステムによる制御およびタイミングによって操作されるため、実際に意思決定を実行し、メモリコントローラが共用メモリへのアクセスを許可し得るようになるまでに多くの時間を費やすことになる。

【 0 0 0 9 】

残念なことに、この実際の意思決定の実行という問題は、従来のメモリコントローラが有するマルチバンク型メモリシステムへのアクセス許可能能力を実質上低下させる。マルチバンク型メモリシステムでは、実際のメモリコアは特定の領域、すなわちバンクに区分され、読み出されるデータもそこに記憶される。より迅速且つより効率的なメモリアクセスが提供できる反面、マルチバンクメモリ装置に対応するために従来のメモリコントローラには複雑な仕組みが求められ、その結果、全体的なシステムとしては、アクセススピードが全体的に著しく低下することになる。

【 0 0 1 0 】

【 発明が解決しようとする課題 】

以上から、外部デバイスへのアクセスを優先順序付けるための方法および装置が必要とされていることがわかる。

【 0 0 1 1 】

【 課題を解決するための手段およびその作用・効果 】

本発明にしたがって、外部デバイスへのアクセスを優先順序付けるための方法および装置を説明する。外部デバイスへのアクセスを優先順序付けるための装置は、要求元デバイスから外部デバイスへの任意数の要求を格納するように適切に構成された要求キューと、要求キューに結合され、そこに格納された任意の要求を取り出すように適切に構成された要求キューコントローラユニットと、外部デバイスからの任意数の応答を格納するように適切に構成された応答キューと、を備える。この装置は、また、応答キューに結合され、そこに格納された任意の要求を取り出すように適切に構成された応答キューコントローラユニットを備え、応答およびそれに関連付けられた要求のそれぞれは、要求の発信元であり

10

20

30

40

50

対応する応答の送信先である特定グループの要求デバイスを示すグループ識別番号に関連付けられており、応答キューコントローラおよび要求キューコントローラユニットは、グループ優先順序セレクタレジスタに格納された優先番号を使用し、格納された要求および応答をそれぞれ優先順序付けることによって、優先順序の高い要求または応答に優先順序の低い要求または応答をバイパスさせる。

【 0 0 1 2 】

本発明の特質および利点については、本明細書中の以下に示される部分および図面により、その一層の理解が達成される。

【 0 0 1 3 】

【発明の実施の形態】

プロセッサなどの複数の装置が同じリソースを共用するシステムの場合、1つ以上の装置が共用リソースへのアクセスを要求する際に典型的に発生する衝突を回避するために、様々な方策が採られてきた。ある方策では、プロセッサを順番に動作させることでその衝突を回避し、またはプロセッサの時間的な共用によって衝突回避を達成する。この方法においては、衝突を避けるために、単純に、プロセッサが順番に共用リソースへのアクセスを行う。一般的に利用されるこのようなシステムは“リングパッシング”または“トークンシステム”を含み、それにより、ユーザーグループでのリングの伝達と類似する所定の連続動作に従って、衝突する可能性のあるプロセッサがシステムによってポーリングされる。

【 0 0 1 4 】

残念ながら、このようなプロセッサによる逐次的なアクセスの方法論では、衝突するプロセッサをシステムがポーリングするためにかなりの時間を費やすため、コンピュータシステムの全体的な動作に著しい制限を課すことになる。

【 0 0 1 5 】

衝突回避のための他の一般的な方策は、コンピュータシステム内のプロセッサ間での優先順位付けによるものである。そのような方法では、各プロセッサに対してシステムの重要性の階層に従った優先順位が付される。そのようなシステムは衝突の問題を解決し、さらに、単純な逐次的アクセスによる衝突回避のシステムよりも効果的である反面、依然として柔軟性に欠くものである。

【 0 0 1 6 】

衝突回避のためのさらに別な一般的方策は、コントローラ型装置に組み込まれた意思決定ロジックを含む。残念ながら、意思決定ロジックの複雑さから、実際の意思決定が実行されコントローラによる共用メモリへのアクセス許可が可能になるまでに、多くの時間を費やすこととなる。

【 0 0 1 7 】

複雑なロジックがシステムの動作速度を低下させるこの問題は、それぞれに異なる動作特性を有しながら相互に接続された複数のメモリ装置の間に点在するメモリを含むマルチチップモジュール型メモリシステムにおいてはさらに顕著である。従来のロジックは、様々なメモリ装置に固有の、それぞれに異なるアクセス特性を補うようには構成され得ないので、全体的なシステムの実行性能を低下させることで解決しようとする。

【 0 0 1 8 】

広義には、図 1 A に示される通り、本発明は、それぞれがユニバーサルデバイスコントローラ 1 0 4 に結合された要求デバイス 1 0 2 を有するシステム 1 0 0 として示すことができる。ここで、要求デバイス 1 0 2 は、任意の数および種類の共用リソース 1 0 8 へのアクセスを提供するように適切に形成されたシステムバス 1 0 6 によってユニバーサルデバイスコントローラ 1 0 4 に結合される。1つの実施形態では、システムバス 1 0 6 は関連するシステムインターフェース層 1 1 0 によってユニバーサルコントローラ 1 0 4 に結合され、さらには、そのユニバーサルコントローラ 1 0 4 が共用リソースインターフェース 1 0 9 によって共用リソース 1 0 8 に結合される。広義には、ユニバーサルコントローラ 1 0 4 は、任意の要求システム 1 0 2 からの共用リソースへの要求および共用リソース動

10

20

30

40

50

作特性パラメータ113に基づき、共用リソース108の状態を判定するように設定される。

【0019】

要求システム102がマルチプロセッサシステム内の1つのプロセッサであり、同じようにそこに結合されているほかのプロセッサによっても共用されているメモリ装置108としての共用リソース108に対するアクセスを要求する場合、ユニバーサルコントローラ104は、所望のリソースアクセス要求を完了するために、実行されるべき動作の順序を決定する。例えばメモリ装置108がSDRAMである場合、その動作は典型的にはプレチャージ、ページクローズ、ページオープン、およびページリードまたはページライトを含む。

10

【0020】

ある特定の動作順序が決定されると、例えばデータ衝突または他の種類の対立を回避するために、ユニバーサルコントローラ104が順序付けられた連続動作の間の適当な時間間隔を決定する。好ましい実施形態では、その時間間隔は、例えばルックアップテーブルに記憶されている共用メモリ装置の動作特性に、部分的に基づいて決定される。その後、適切に順序付けられたアクセスコマンドがユニバーサルコントローラにより出力され、次いで、共用メモリによって応答される。

【0021】

以下に示される本発明の詳細な説明では、発明についての完全な理解を促進するために複数の具体的な実施形態が示される。しかし、当業者には明白となるように、本発明はこれらの特定の詳細に依らず、または他の要素あるいは工程を利用することにより実践され得る。他の場合では、本発明の本質が不明瞭になることを避けるため、周知の工程、手順、構成要素、および回路についての詳細な説明が割愛されている。

20

【0022】

以下、本発明は、プロセッサと共用メモリとの間の連絡機構として機能するように設定されたメモリコントローラの観点から説明される。しかし、本発明は、共用であるか否かに関わらず、任意のリソースへのアクセスを制御し得るユニバーサルコントローラとしても実行し得ることに留意されたい。そのようなリソースはメモリである必要性は無く、事実、本発明は、例えば、バスアクセスの待ち時間を低減することによりシステムバスの効率的な帯域幅を増加させることを目的としてマルチプロセッサ内の情報量を制御するなど、共用システムバスへのアクセスを制御するためにも用いられ得る。

30

【0023】

次に、図1Bでは、プロセッサなどの要求デバイス102を有し、システムバス106によってユニバーサルコントローラ104に結合されるシステム100が示される。さらに、コントローラ104は共用リソース108に結合され、共用リソース108は、例えば、様々な形態を採り得るメモリ108であり、DRAM、SDRAM、SLDRAM、EDO、FPM、またはRDRAMなどである。例示の実施形態では、システムバス106は単方向アドレスバス106-1を含み、その単方向アドレスバス106-1はプロセッサ102によって出力されるメモリアドレス要求をユニバーサルコントローラ104に伝達する。さらにシステムバス106は単方向アドレスバス106-2も含み、その単方向アドレスバス106-2はアドレスバス106-1と連係してメモリアドレスに関するコマンドを伝達する。例えば、プロセッサ102がメモリ108における特定のメモリ位置に記憶されている実行可能な命令を要求する場合、プロセッサはコマンドバス106-2に対して読み取り要求(システムコマンドと称す)を出力し、実質的には同時に、アドレスバス106-1に対しても対応するメモリアドレス要求(システムアドレスと称す)の出力が行われる。システムアドレスおよびシステムコマンドの双方はコントローラ104に含まれる設定可能なシステムインターフェース110によって受け取られる。ここで、設定可能とは、受け取ったシステムコマンドおよびシステムアドレスが、メモリ108によって要求されるいかなる方法および形態のものでも、システムインターフェース110がそれを処理できるように、システムインターフェース110を設定し得るということを

40

50

意味するものである。これにより、プロセッサ 102 が各メモリ装置に対して個別の要求を発する必要がなくなり、プロセッサ 102 が必要としているデータを、コントローラ 104 に結合する任意の数および種類のメモリ装置に格納することができる。

【0024】

例示の実施形態において、システムインターフェース 110 は、受け取ったシステムコマンドおよびシステムアドレスをユニバーサルコマンド 200 と称されるものに変換するように設定されている。ユニバーサルコマンド 200 の例は図 2A に示される。1つの実施例では、共用リソースが DRAM 型メモリ装置 (SLDRAM、SDRAM、EDO DRAM などを含む) である場合、ユニバーサルコマンド 200 は、メモリ 108 の任意のメモリアクセス要求を実行するために必要な全ての動作を含む 5 つのデータフィールドから形成される。そのような動作は、データプレチャージフィールド 202 によって示されているプレチャージ動作を含み、このフィールドは、特定のロウを予めチャージしておく必要が有るかどうかを表示するために用いられる、プレチャージ動作を含む。他の動作として、データ活性化フィールド 204、データリードフィールド 206、データライトフィールド 208 およびデータリフレッシュフィールド 210 を含む。例えばここで、メモリ 108 が、メモリバンク 1 において現在アクティブである (すなわち、リードまたはライトが行われた後のオープンされている) メモリページ 1 を有し、且つ、続くプロセッサコマンドが、メモリバンク 1 のページ 2 上に記憶されているデータを読み取り、プロセッサ 102 に出力するよう要求しているとする。この場合、プロセッサ 102 によって要求されるコマンドを実行するためには、ページ 1 がクローズされ (すなわち、ページ 1 がプレチャージされる)、ページ 2 は活性化されなければならない。活性化が完了した後、ページ 2 からのリードが行われる。したがって、図 2B に示されるユニバーサルコマンド 212 は、データフィールド 202, 204, 206, 208, 210 を有するユニバーサルコマンド生成部 110 によって生成され、そのうちのデータフィールド 202, 204, 206 は “関連する動作の実行” を示す “1” に設定され、データフィールド 208, 210 は “関連する動作の不実行” (すなわち、“NOP”) を示す “0” に設定される。

【0025】

図 1B に戻る。メモリ 108 へのアクセスは複数の異なる要求デバイスによって共用されているため非常に動的であり、従ってメモリ 108 の状態は常に変化する。メモリの状態とは、特定のメモリ位置において特定の動作を間違いなく実行するためには、そのメモリ位置の状態を知る必要があることを意味する。例えば、特定のメモリページがクローズしている場合、リード動作を実行するためにはまずそのメモリページをオープンする必要がある。従って、特定のアドレス位置のその時点での状態を探知するために、特定のメモリ位置に対して実行された最も新しい動作が図 3 に示されるリソスタグ 300 によって識別される。本発明の 1 つの実施形態では、リソスタグ 300 は、特定のメモリアドレス位置を識別するために用いられるアドレスフィールド 302、そのアドレスフィールド 302 によって識別されたアドレスに最後に発行されたコマンドを識別するために用いられる最終発行コマンドフィールド 304、さらに最終コマンド発行時間データフィールド 306 を含む。例えば、メモリアドレス ADD_5 のリソスタグ 308 は時間 5 (5 システムクロック周期を示す) にページリードコマンドが発行されたことを示し、リソスタグ 310 は、同じメモリアドレス ADD_5 について、時間 10 にそのメモリページにページライトが実行されることを示している。メモリアドレス ADD_5 の状態を観測することにより、ユニバーサルコントローラ 104 は ADD_5 におけるメモリページが既にオープンしていること、従って、ページオープン操作は必要ないことを認知する。

【0026】

リソスタグバッファ 114 に記憶されているタグ 300 によって提供されるリソースの状態についての情報を元に、設定可能システムインターフェース 110 に結合されるコマンド順序付け器 116 は、ユニバーサルコマンド 200 のコマンド構成要素 202 ~ 210 のそれぞれの間における適切な時間間隔を提示し、それにより、図 2C に示されるような、コマンド構成要素 202 ~ 204、およびコマンド構成要素 204 ~ 206 がそれぞ

10

20

30

40

50

れ時間間隔 t_1 および t_2 となるよう構成された順序付け済みコマンド 220 が提供される。ここで、コマンド構成要素 208 ~ 210 は“NOP”型のフィールドであるため、順序付け済みコマンド 220 はこれらのフィールドについて如何なる参照も含んでおらず、構成要素 202 ~ 206 に必要なクロック周期と、 $t_1 + t_2$ に実質的に等しい時間周期と、の合計に実質的に等しい時間周期を必要とするのみである。これにより、コマンド順序付け器 116 は、プロセッサ 102 とメモリ 108 との間のコマンドおよびデータの最適な流れを提供し得る。

【0027】

発明の別の実施形態では、共用リソース 108 が SDRAM などのマルチバンク型メモリ装置である場合、または共用リソースがマルチチップモジュールなどのマルチ装置メモリである場合、リソースタグバッファ 114 は、例えば特定のバンクまたは装置においてオープンしているページ全てについてのリソースタグを記憶し得る。1つの実施例では、比較器（図示せず）がシステムアドレスにおけるバンク番号または装置識別子を探知し、ページアドレスおよびシステムアドレスをタグバッファ 114 の内容と比較する。比較結果が“一致”でない場合（すなわち、アドレスが適合しない場合）、ユニバーサルコントローラ 104 はタグバッファ 114 からのアドレスを用いて古いページをクローズしなければならない。さらに、新しいシステムコマンドに基づき新しいページをオープンしなければならない。

【0028】

複数の異なる装置がユニバーサルコントローラ 104 によって補助されている場合、特定の装置にのみ関連する動作パラメータであって、入力されるシステムアドレスにも関連している動作パラメータを選択し得ることが望ましい。ユニバーサルコントローラが複数の異なる装置を補助している場合に、ユニバーサルコントローラ 104 に結合されたアドレス空間コントローラ 120 が図 1C に示されている。例示の実施形態では、アドレス空間コントローラ 120 は入力されたシステムアドレスに関連する 1つの装置を示す装置特有のパラメータのみを選択する性能を有する。図 1D に示される具体的な実施例では、アドレス空間コントローラ 120 は比較器 122 を含み、その比較器 122 が、入力されたシステムアドレスと、入力されたアドレスに関連する装置（または同様に、メモリ領域）を識別する領域アドレス範囲バッファ 124 の内容とを比較する。特定の装置または領域が識別されると、装置パラメータレジスタ 126 および 128（それぞれがバッファ 124 に結合され、特定の装置ごとに特有のパラメータを含む）のグループ中 1つのレジスタが選択される。選択された装置パラメータレジスタは次いでそのシステムアドレスに対応する装置についての特定の動作パラメータを提示する。別の実施形態では、選択された装置パラメータレジスタの内容は LUT 118 に入力される。これにより、任意の数の異なる装置をユニバーサルコントローラ 104 によって補助することができ、各装置の特定の動作パラメータが識別されて、対応するユニバーサルコマンドの最適な順序付けに用いられる。

【0029】

ユニバーサルコントローラに結合された装置の 1つがビジー状態で新しいコマンドを受付られないといった場合に、コマンド列で待機中の任意の別のコマンドを選択し得ることが有益であるという点に留意すべきである。発明の別の実施形態では、装置による応答およびユニバーサルコントローラによる要求のそれぞれが、それぞれ関連する識別番号 150 を有しており、例示の実施形態では、図 1E に示されるように、その識別番号は 5 ビット長のデータワードである。識別番号 150 は、2 ビット長のグループセクタフィールド 152 および 3 ビット長の要求番号フィールド 153 を含むように形成される。グループセクタ (GS) は特定のシステム要求がどのグループに属するか（例えば、プロセッサ）を判定し、要求番号 (RN) は、グループセクタフィールド 152 によって識別された関連グループの要求または応答の番号を表す。その際、同一のランシーバからの連続する要求は連続した要求番号を有する。

【0030】

別の実施形態では、グループ優先順位セクタレジスタ154は応答グループまたは要求グループのそれぞれについての優先順位値を含み、より高い優先順位値を有する応答グループまたは要求グループが低い優先順位値のグループに先行する。これにより、優先順位値の低い要求や応答が次のクロック周期で処理され得ない場合に、その低順位値の要求または応答に先行して高順位値の要求または応答が処理され得る。いわゆる動ロック（ライブロック、Livelock）を防ぐため、動ロックカウンタレジスタ156は、低優先順位値の要求（または応答）に先行し得る、高優先順位値を有する連続する要求（または応答）の数の情報を含む。これにより、低優先順位値の要求（または応答）はクロック周期の多くの周期中放置されずにすむことになる。

【0031】

また別の留意点として、コマンドの流れとデータの流れの双方の制御を最適化するために、共用リソースのそれぞれがそれぞれに動作特性一式（例えば、DRAM型装置の場合はアクセス時間、CAS待ち時間など）を関連させている点があげられる。ユニバーサルコントローラ104によって1つ以上の共用リソースが補助されている場合、共用リソースのそれぞれは異なる動作特性の一式を有しており、別の実施形態では、その動作特性はコマンド順序付け器116に結合するルックアップテーブル（LUT）118に記憶される。コマンド順序付け器116は、リソースタグバッファ114に記憶されるリソースタグと連係してLUT118が提供する情報を用いて、コマンド構成要素202～210を適切に順序付け、順序付け済みコマンド220を形成する。これは、共用リソースが、マルチチップモジュールなど、まさにそれぞれが実質的に異なる動作特性を有するメモリ装置の集合である場合には特にあてはまる。

【0032】

次に、図4は、発明の実施形態に従ってユニバーサルコントローラが共用リソースへのアクセスを図るプロセス400の詳細を示したフローチャートである。このプロセス400はシステムが共用リソースへのアクセスを要求する402から開始する。共用リソースがDRAM型メモリ装置の場合、その動作はプレチャージ、リフレッシュ、クローズ、オープン、リードおよびライトを含む。例えば、プロセッサが、システムコマンド（すなわちページリード）と、要求されているページが記憶されているメモリ内の場所を示す関連のシステムアドレスとを生成することにより、共用メモリ内に記憶されるメモリページを要求する。これは、好ましい実施形態では、404において、例えば、共用メモリ内でのアクティブなメモリ位置に関連したリソースタグを用いてリソースの状態が判定される。次いで、406にて、共用リソース対する所望の要求を実行するために必要な、動作の順序付けについての判定が行われる。408では、ユニバーサルコントローラが、所望の要求を実行するために必要な動作の連続順序に基づいてユニバーサルコマンドを生成する。例えば、ページリード動作を実行するためには、前段階でオープンされているページをクローズし、新しいページを活性化し、それからリード動作が実行される必要がある。その際、これら全てはユニバーサルコマンド構成1つで理解される。共用リソースについてのリソースタグおよび特有の動作特性を用いてユニバーサルコントローラがユニバーサルコマンドを形成すると、次いで、410にて、ユニバーサルコントローラはユニバーサルコマンドのコマンド構成要素それぞれの間の適切な時間間隔を判定する。その後、412において、順序付け済みコマンドが共用リソースに対して発行される。この際、別の実施形態では物理ステージが用いられる。最後に414において、共用リソースは、例えば、システムアドレスに示される位置に記憶されているデータを提示することなどによって、順序付け済みコマンドに応答する。

【0033】

発明の別の実施形態では、図5に示されるプロセス500を用いて、ユニバーサルコントローラがリソースの状態（404）および実行する動作の順番（406）を決定する。このプロセス500は、502にてリソース区分識別子（すなわち、メモリアドレスレジスタ）をリソース識別子（すなわち、リソースタグアドレスフィールド202）と比較することによって開始される。504において“一致”の発生が確認された場合（すなわち、

10

20

30

40

50

新しいコマンドのアドレスがその時点でのタグアドレスフィールドと適合する場合)、続いて、506において次のコマンドが発行される。他方、新しいコマンドアドレスがその時点でのタグアドレスフィールドに適合しない場合(すなわち、不一致の場合)、次いで、508にて、古いページがまだオープンしているか否かの判定が行われる。古いページがオープンしている場合には、510にてそのページがクローズされ、512にて新しいページがオープンされる。しかし、508にて古いページがオープンしていないことが確認されると、次いで、512において新しいページがオープンされる。どちらの場合でも、一度新しいページがオープンされると、506にて次のコマンド(データ操作)が発行される。

【0034】

発明の別の実施形態において、ユニバーサルコントローラは、図6に示されるプロセス600に基づいて、連続動作のそれぞれの間における適切な時間間隔を決定する(410)。このプロセス600は、602において、ユニバーサルコントローラが、ある特定のリソースに対しての新しい一連のコマンドの最初のコマンドと、その時点までに同じリソースに対して発行されたコマンドのうち最も新しい一連のコマンドの最後のコマンドとを比較することによって開始される。604において、ユニバーサルコントローラは、新しいユニバーサルコマンドの最初のコマンド構成要素と、それ以前のもののうち最も新しいコマンドの最終コマンド要素とを比較することによって、ユニバーサルコマンド構成要素間の時間的な制約を判定する。別の実施形態では、ユニバーサルコントローラは表1に示されるような2次元配列の形態をとる2指標ルックアップテーブル(LUT)を用いる。その中で、配列の第一行は古い(すなわち今までのうちで最も新しい)コマンドを表し、配列の第一列は新しいコマンドを表す。例えば、表1を参照して、古いコマンドがページリードであった場合、且つ新しいコマンドがページクローズであった場合、それらの、ページクローズという新しいコマンドとページリードという古いコマンドとの交差位置には、それら2つの動作の間で許可される最低時間量(すなわち、発行に要する物理的な最低限の時間)が示される。通常、LUTに記憶される情報は共用リソースの製造者によって提供される。

【0035】

【表1】

表 1

古いコマンド

新しい
コマンド

	ページ クローズ	ページ オープン	リード	ライト
ページ クローズ			5φ	
ページ オープン				
リード				
ライト				

【0036】

特定のユニバーサルコマンド構成要素についてのリソースの物理的な制約が判定されると、606において、同じユニバーサルコマンドに更なるコマンド構成要素が存在しているか否かの判定が行われる。更なるコマンド構成要素が存在しない場合、608において、ユニバーサルコマンドおよび関連する構成要素の時間間隔に関する詳細が記憶される。他

方、ユニバーサルコマンドに更なるコマンド構成要素が含まれている場合、制御は604に戻され、その構成要素について、対応する物理的な時間的制約が判定される。

【0037】

しかし、例えば複数のメモリバンクを有する共用メモリ108内の物理ページの状態を観測するためには、非常に多くのリソースタグが必要となり、それに従い非常に多くのキャッシュメモリがリソースタグバッファ114のために必要となる。これでは、それぞれが遠隔に位置するメモリの特定のメモリページについての固有のリソースタグを検索するために非常に多くの時間を要することになり、その結果、ユニバーサルコントローラ104の全体的な動作速度を低減させることになる。図7Aに示される別の実施形態では、ページヒット/ミスコントローラ702がユニバーサルコントローラ104に含まれ、ページレジスタ704の数Mがマルチバンクメモリ706内のメモリバンク数N未満になるよう設定される。これは、M個のページレジスタ704中で、全てのバンクが対応できるがわけではないからである。この動作において、Mページレジスタ704のそれぞれは、オープンしているページのアドレスおよび状態データを記憶する。また、ランダムページレジスタ番号生成部708は、ページレジスタに対応する無作為なM以下の整数値を生成し、オープンしているページのアドレスと交換する。比較器710は、入力されたシステムアドレスと、M個のすべてのレジスタのバンク番号およびページアドレスとの比較を並行に行い、以下に示す4つの可能な結果のいずれかを得る。

【0038】

1) 比較器710がヒット(一致)を示す場合、要求されているバンクの所望のページがオープンされ、アクセスへの準備が整う。

【0039】

2) 比較器710がバンクのヒット(一致)およびページのミス(不一致)を示す場合、ユニバーサルコントローラ104は、ページレジスタからのページアドレスを利用して古いページをクローズし、且つ、システムアドレスからのページアドレスを利用して新しいページをオープンする必要がある。

【0040】

3) 比較器710がバンクおよびページの双方でのミスを示す場合、ユニバーサルコントローラ104は、ランダムページレジスタ番号生成部によって提供される番号のバンクにおける任意の古いページをクローズし、且つシステムアドレスを利用して新しいページをオープンする必要がある。その後、所望のバンクへのアクセスを行う。

【0041】

4) バンクおよびページの双方がミスで、しかし少なくとも1つのページレジスタが未使用である場合、そのレジスタが利用されて、新しいページがオープンされる。

【0042】

別の実施形態では、図7Bに示されるように、ランダムページレジスタ番号生成部708は、最も以前に用いられた(LRU)比較器712によって代替され、M個のレジスタ704のうちのどれが最も長く未使用であるか(すなわち、最も以前に用いられたか)を判定する。

【0043】

マルチバンクメモリ706における物理ページの状態の観測に加え、図8に示されるバンクアクセスコントローラ800は、マルチバンクメモリ706に含まれるメモリバンク数Nに対応するN個のバンクレジスタ802を含む。関連するバンクの情報が記憶されるバンクレジスタ802は、バンクの識別番号を規定するバンク番号フィールド804を含む。さらに、バンクレジスタ802は、バンク番号フィールド804中のバンク番号によって識別される特定のバンクの状態を示すバンクステータスフィールド806を含む。具体的な実施形態では、バンクステータスフィールド806は表2に示されるような値をとり得る。

【0044】

【表2】

10

20

30

40

50

表 2

バンクレジスタ要素	内容
バンク番号	バンクレジスタ内に格納されている情報に対するバンクを識別する
バンクステータス	以下のようにバンクの状態を識別する： “00” - 無効なエントリ “01” - バンクカウント値が0になるまでバンクカウント値を低下させる。バンクカウンタが1以上のとき、そのバンクへのアクセスが禁止される。 “10” - バンクはクローズされている。 “01” - バンクカウント値が0になるまでバンクカウント値を低下させる。バンクカウンタが1以上のとき、そのメモリ内の全バンクへのアクセスが禁止される。
バンクタイマ	バンクカウンタが1以上のとき、バンクステータス値に応じてメモリへのアクセスが禁止される。

【0045】

400 ~ 800 Mb/s / ピンの範囲の速度でバスデータを伝達するシンクロナスリンク DRAM (SLDRAM) などのパケット指向型高速メモリの発展に伴い、メモリアクセスの衝突に起因する問題は益々増加している。まず図9Aを参照すると、発明の実施形態による、例示的なSLDRAM型マルチプロセッサシステム900が示される。マルチプロセッサシステム900は、システムバス906によってコントローラ904に接続されるプロセッサ902を含む。ユニバーサルコントローラ904は、次いで、SLDRAMバスによってシンクロナスリンクDRAM (SLDRAM) 908およびSLDRAM 910に接続される。SLDRAMバスは単方向コマンドバス912および双方向データバス914で構成されている。留意点として、図9Aには2つのSLDRAMのみが示されているが、バス912および914によって、任意の数のSLDRAMがユニバーサルコントローラ904に接続され得ることがあげられる。別の場合では、SLDRAMは、SLDRAM 908のようなSLDRAMを任意の適切な数だけ含むバッファ付きモジュールの形態をとり得る。ユニバーサルコントローラ904をSLDRAM 908および910のそれぞれに接続する初期化/同期(I/S)バス916は、ユニバーサルコントローラ904によって生成される初期化信号および同期信号の信号経路を提供する。

【0046】

発明の別の実施形態では、ユニバーサルコントローラ904からのパケット化されたコマンド、アドレス、および制御情報が、コマンドバス912上をSLDRAM 908および910に選択的に伝達される。データバス914は、ユニバーサルコントローラ904からのパケット化されたライトデータをSLDRAM 908および910のいずれか選択された方へ伝達するよう設定される。または、データバス914が、SLDRAM 908および910のいずれか選択された方からのパケット化されたリードデータをユニバーサルコントローラ904に送り返すようにも形成される。留意点として、コマンドバス912およびデータバス914は通常、例えば400 MB/s / p、600 MB/s / p、800 MB/s / pなどの互いに同一の速度で動作することがあげられる。

【0047】

ユニバーサルコントローラ904により生成され、コマンドバス912によって伝達される複数の制御信号は、例えば、無差異の走行クロック信号(CCLK)、FLAG信号、コマンドアドレス信号CA, LISTEN信号、LINKON信号およびRESET信号を含む。通常、パケットコマンドは4つの連続する10ビットワードから成り、コマンドの最初のワードは、FLAG信号の最初のビットが“1”となっている。好ましい実施形態では、無差異の走行クロック信号CCLKの両端は、コマンドワードをラッチするため

10

20

30

40

50

に S L D R A M 9 0 8 および 9 1 0 によって使用される。S L D R A M 9 0 8 および 9 1 0 は、入力されたコマンドについてコマンドバス 9 1 2 を調べることにより、H レベルの L I S T E N 信号に応答する。または、S L D R A M 9 0 8 および 9 1 0 は、省電力スタンバイモードに入ることによって、L レベルの L I S T E N 信号に応答する。L I N K O N 信号および R E S E T 信号は、S L D R A M 9 0 8 および 9 1 0 のいずれかが選択された方を既知の所望の状態になるように停止するかまたは起動するためにそれぞれ使用される。

【 0 0 4 8 】

ここで議論の残された点に関しては、S L D R A M 9 0 8 は、適当と考えられる任意の数の S L D R A M がユニバーサルコントローラに接続され得ることが、よく理解された上でのみ論じられるであろう。上に論じた通り、S L D R A M 9 0 8 のような典型的な S L D R A M デバイスは、メモリ領域だけでなく、メモリバンク、カラム、ロウ、ビットによって階層的に編成されている。これらの各階層レベルが相互に異なった動作特性を持つことが実際に認められることに注意すべきである。そのような動作特性には、メモリアクセス時間、チップイネーブル時間、データ検索時間等のパラメータを含むが、これらに限定はされない。領域が、それぞれ異なるコマンド待ち時間やデータ待ち時間を持つ異なるメモリタイプやメモリグループといった異なるデバイスに定義されるのに対し、マルチバンクメモリ内のバンクが通常同じ動作特性を持つことは注記されるべきである。例えば、1 つのローカルメモリグループがメモリコントローラにダイレクトに接続でき、また、ローカルメモリグループに関するコマンド待ち時間やデータ待ち時間を仲介するドライバが増加させるボード上に位置する第二の非ローカルメモリグループに接続できる。他のケースでは、マルチチップモジュールを形成する様々なメモリチップのそれぞれは、異なるメモリ領域であると考えられる。

【 0 0 4 9 】

図 9 A のシステムに関してさらに具体的に述べると、S L D R A M 9 0 8 は、コマンドバス 9 1 2、データバス 9 1 4、I / S バス 9 1 6 によってそれぞれ個別にアクセス可能な 4 つのメモリチップ A、B、C、D を持つマルチチップモジュールである。メモリチップ A ~ D のそれぞれは、コマンドやデータパケットを最適にスケジューリングするために、異なった動作特性を持つことができ（通常は製造者によって与えられる）、ユニバーサルコントローラ 9 0 4 は特定の階層レベル、およびまたは、対応するメモリ領域の動作特性を使用することができる。

【 0 0 5 0 】

例として、図 9 B は、図 9 に示されたマルチプロセッサシステムに従った代表的 S L D R A M バストランザクションの典型的タイミング図を示している。演算の間、プロセッサは通常、例えばリードコマンド 9 5 0 やライトコマンド 9 5 2 のように、S L D R A M 9 0 8 の適当なメモリバンク（複数でも可）が応答するプロセッサコマンドパケットを生成する。通常、リードコマンド 9 5 0 やライトコマンド 9 5 2 は、それらが生成されるプロセッサ 9 0 2 の特定の要求に基づいて、システムバス 9 0 6 上でパイプライン化されており、S L D R A M の最適のパフォーマンスには適していない。システムクロック CL_{sys} （示されていない）は必要なタイミング信号を与える。

【 0 0 5 1 】

この例では、プロセッサ 9 0 2 a は、S L D R A M 9 0 8 のメモリチップ A に位置するメモリアドレス MA_1 を持つリードコマンド 9 5 0 を生成する。他方、プロセッサ 9 0 2 b は、同じく S L D R A M 9 0 8 のメモリチップ A に位置するメモリアドレス MA_2 を持つリードコマンド 9 5 2 を生成する。この例では、リードコマンド 9 5 0 は、ライトコマンド 9 5 2 の出力に優先するシステム・バス 9 0 6 への出力である。ユニバーサルコントローラ 9 0 4 は、まずリードコマンド 9 5 0 を受け取り、続いてそのコマンド自体とコマンドアドレス MA_1 に基づいて、ユニバーサルコントローラ 9 0 4 内に格納された終点アドレス特定情報を使って、コマンドの処理を始める。いったん最短発行時間が決定されると、ユニバーサルコントローラ 9 0 4 は次に、受信したプロセッサ・コマンド 9 5 0 に応じ

10

20

30

40

50

て、SLDRAMコマンドパケットリード960を生成し、それをコマンドバス912に送り出す。

【0052】

一般に、SLDRAMコマンドパケットは、表3に示したように、8バンク、1024ロウアドレス、128カラムアドレスを持つ64MのSLDRAMを表わす4つの10ビットワードとして編成されている。示されたように、バンクアドレス(BNK)は3ビット、ロウ・アドレス(ROW)は10ビット、カラム・アドレス(COL)は7ビットである。他にも多くの編成や密度が可能であり、図示された40ビットフォーマットの外、適切と規定されうる他のどんなフォーマットでも収容されうる。電源が入っている間、ユニバーサルコントローラ904は、バンク、ロウ、カラムの番号や、その時にユニバーサルコントローラ904によって格納される関連付けられた動作特性などの要因についてのSLDRAMのポーリングに基づいてコマンドパケットを編成する。

10

【0053】

コマンドパケットの最初のワードは、複数のチップIDビットを含む。SLDRAMは、ローカルIDに整合しないコマンドはどれでも無視する。チップIDは電源オンとともにユニバーサルコントローラ904によって、初期化信号と同期信号を用いて割り当てられる。このようにして、ユニバーサルコントローラ904は、セパレートチップイネーブル信号またはグルーロジックの生成によって、マルチプロセッサシステム900のそれぞれのSLDRAMに一対一対応でアドレス指定する。

20

【0054】

【表3】

表 3

SLDRAMコマンドパケット構造

FLAG	CA9	CA8	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
1	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	CMD5
0	CMD4	CMD3	CMD2	CMD1	CMD0	BNK2	BNK1	BNK0	RW9	RW8
0	ROW7	ROW6	ROW5	ROW4	ROW3	ROW2	ROW1	ROW0	0	0
0	0	0	0	COL6	COL5	COL4	COL3	COL2	COL1	COL0

30

【0055】

リードコマンド950とライトコマンド952はパイプライン化されているので、ユニバーサルコントローラ904は、リードコマンド950の受け取った後一定時間の後にライトコマンド952を受け取り(またはバッファに格納させておくこともできる)、次いで、ライトコマンド952に対応するSLDRAMコマンドパケットライト962を発行する。同一のバンク(A)が両方のコマンドにアクセスされるので前に発行されたリードコマンド960の干渉を避けるために、ユニバーサルコントローラ904は、MA₂の特定の特性データならびに最短発行時間を生成させるためのリードコマンド960の発行時間(すなわち、発行の時刻)と、ライト962のデータオフセットを使う。

40

【0056】

このようにして、ユニバーサルコントローラ904は少なくとも、コマンドやデータパケットのストリームの現在の状態ならびに特定の終点アドレスデバイスの動作特性に基づいて、SLDRAMコマンドパケットの発行をダイナミックにスケジューリングできる。

【0057】

次に、本発明の実施形態に従ったメモリコントローラ1000のブロックダイヤグラムを図解した図10に言及する。メモリコントローラ1000が図1に示されたユニバーサルコントローラ104の実施形態の1つにすぎず、したがって本発明の限界を制限するものと捉えるべきでないことは注記すべきである。メモリコントローラ1000はシステム・

50

インタフェース1002を含み、これはシステム・バス906を介してプロセッサ902をメモリ・スケジューラ1006(スケジューラと呼ぶ)に接続する。本発明の実施形態の1つにおいては、システムインタフェース1002は、プロセッサ902によって生成されたメモリコマンドパケットとそれに関連付けられた書き込みデータパケットの両方の、メモリコマンドパケットスケジューラ1004への伝送に備えるために配置されている。内部バッファがいっぱいで新しいコマンドが収容できないとスケジューラ1006が表示している状況では、システムインタフェース1002はスケジューラ1006が新しいコマンドを受け入れ準備完了と表示する時まで新しいコマンドをホールドする。

【0058】

シンクロナスリンク・メディアアクセスコントローラ(SLiMAC)1008は、スケジューラ1006とSLDRAM908間の物理的インタフェースを提供する。さらに具体的には、SLiMAC1008は、それぞれコマンドバス912とデータバス914を介して、SLiMAC1008をSLDRAM908に接続するコマンドインタフェース1010とデータインタフェース1012を含む。本発明の好適な実施形態では、コマンドインタフェース1010はメモリコマンドを、関連するコマンドクロックCLKとともにSLiMAC1008からSLDRAM908に伝送する。いくつかの実施形態では、通常200MHzで作動するコマンドクロック信号CLKを生成させるために、SLiMAC1008はインタフェースクロック信号ICK(およそ100MHzで作動できる)を使うクロックダブラを組み入れている。

【0059】

本発明の実施形態の1つでは、データインタフェース1012はデータバス914のデータの受信、送信両方をする。データバス914の幅は、必要とされる数のSLDRAMをサポートするのに十分な大きさにできることは注記すべきである。したがって、必要な帯域幅を供給するために、必要なだけ十分のデータインタフェースがSLiMACに含まれる。一例として、データバス914が32ビット幅(例えば、SLDRAMごとに16ビット)であれば、その場合、SLiMAC1008は、個々のSLDRAMに関連付けられた16ビットをそれぞれ制御できる2つのデータインタフェースを備えることができる。このようにして、SLiMAC1008に含まれたデータインタフェースのサイズは、それに接続されたSLDRAMの特定の構成に厳密に適合させることができる。

【0060】

コマンド・インタフェース1010を用いるのとほぼ同じ方法で、SLiMAC1008は、SLDRAM908からSLiMAC1008に送信されたリードデータに伴うデータクロック信号DCLKを供給することができる。本発明の実施形態の1つにおいては、データクロックDCLKは、インタフェースクロックICK周波数を約100MHzから約1000MHzに増加させるクロック・ダブラを使うことで生成される。また、インタフェースクロック信号ICK、コマンドクロック信号CLK、データクロック信号DCLKが全位相同期的であることも注記されるべきである。

【0061】

本発明の好適な実施形態においては、スケジューラ1006は、システムコマンドとそれに関連付けられたシステムアドレスデータを、接続されたシステムインタフェース1002から受信するように配列された制限ブロック1016を含む。制限ブロック1016はSLDRAMコマンドパケットデータと関連付けられたタイミング情報を再順序付けブロック1018に供給する。ライトバッファ1020はシステムインタフェース1002からリードデータを受け取る。スケジューラ1006によって指示されたように、リードデータは、リードデータをシステム・インタフェース1002に供給するように配列されたデータバス914に接続されたリードバッファ1022を通してデータインタフェース1002から伝送される。初期化/同期(I/S)バス914に接続されたI/Sブロック1024は、適切な初期化および/または同期信号をSLDRAM908に要求通りに供給する。

【0062】

動作中において、スケジューラ1006は、プロセッサ902によって生成されたパイプライン化されたメモリコマンドパケットを受け取る。通常、メモリコマンドパケットはメモリコマンドとそれに関連付けられたメモリアドレスとで構成されている。本発明の実施形態の1つでは、スケジューラ1006は、メモリコマンドとそれに関連付けられたデータパケット（もしあれば）が差し向けられた終点アドレスを決定するために、受信された新コマンドに関連付けられたメモリアドレスをデコードする。いったんデコードされれば、スケジューラ1006は、新しいSLDRAMコマンドパケットを送り出すために、そこに格納された終点アドレス特定デバイス特性データを、直前に送り出されたメモリコマンドに関連付けられた情報とともに使う。新しいSLDRAMコマンドパケットはコマンドバス912への出力であり、最終的にはSLDRAMコマンドパケットに含まれたチップIDによって識別されるSLDRAMへの出力である。

10

【0063】

スケジューリングプロセスの一部として、スケジューラ1006は、新しいコマンドの発行前に、直前に発行された要求されたコマンドの発行後の最短時間量を決定する。上述のように、例えばメモリバンクなどのSLDRAMのそれぞれの階層レベルは、異なった動作特性を持つことができる（通常、製造者によって与えられている）ので、スケジューラ1006は、初期化の間に、それがサービスするSLDRAMのそれぞれにポーリングする。いくつかの実施形態では、接続されたメモリデバイスが動作特性を決定するためのポーリングを許さない場合、メモリ特定パラメータ（タイミングなど）は制限ブロックレジスタ1016に直接書き込むことができる。いったんSLDRAMがポーリングされれば、スケジューラ1006は、後で適切なスケジューリングプロトコルを展開するために使用するデバイス特定情報を格納する。このようにして、スケジューラ1006は、ハードワイヤリングを必要とすることもなく、さらなる時間の浪費や経費のかかる手続もなく、どんな数やタイプのSLDRAMにも適応するスケジューリングサービスを供給することが可能である。

20

【0064】

図11は、本発明の実施形態に従った制限ブロック1100を模式的に図解したものである。制限ブロック1100が図10に示された制限ブロックの唯一の可能な実施形態ではなく、したがってこれに限定されるものではないことは注記されるべきである。制限ブロック1100は、プロセッサ902によって生成される新しいメモリコマンドに関連付けられた受信した新しいアドレス信号をデコードするために配列され、システムインタフェース1002に接続されたアドレス・デコーダ1102を含む。デコードされた新しいアドレス信号は、配列タグ・レジスタ1104への入力を供給し、この配列タグレジスタには、関連するSLDRAMメモリバンクのすべてのための、または場合によってはサブセットのみのためのステータスや他の関連情報が格納されている。配列タグレジスタ1104は、セクタ1106への入力を供給し、セクタ1106は、デコードされた新しいコマンドに基づいて、選択された仮想バンクに関連するデータをルックアップテーブル（LUT）1108に伝える。

30

【0065】

制限ブロック1100はまた、システムインタフェース1002に接続された領域比較器1110を含んでおり、この領域比較器1110は、受信した新しいアドレス信号を用いて、新しいコマンドアドレスが存在するメモリ領域を示す領域識別子を供給する。このようにして、制限ブロック1100は、メモリ領域特定特性データに少なくとも部分的には基づいて、新しいメモリコマンドのための最善のスケジューリングプロトコルを提供することが可能である。領域比較器1110は、新しいコマンド信号とともに、領域識別子をLUT 1108に入力として供給する。次にLUT 1108は、新しいコマンドとそれに関連付けられた新しいアドレスをSLDRAMコマンドパケットに変換するために使われる最小デルタ発行時間とデータオフセットを供給する。最小デルタ発行時間が、直前に発行された古いコマンドとの関係において新しいコマンドを発行する（クロックサイクルでの）デルタ時間を示していることは注記されるべきである。データオフセット時間は、

40

50

新しいコマンドの発行後に新しいコマンドに関連付けられたリードデータパケットを受信するための、クロックサイクルにおけるデルタ時間を表わしている。

【 0 0 6 6 】

本発明の一実施形態において、制限ブロック 1 1 0 0 は 1 6 個の配列タグバンクレジスタを含み、L U T 1 1 0 8 は、1 6 個の関連付けられたレジスタを持つ 4 個のタイミング領域それぞれのために 4 つの異なるパラメータセットを格納可能である。

【 0 0 6 7 】

図 1 2 は、本発明の実施形態の 1 つに従って、受信したプロセッサコマンドに回答する S L D R A M バス信号のタイミング図 1 2 0 0 である。表 4 が、様々な発生信号を識別することにより、制限ブロックによって実行されるスケジューリングプロセスを要約していることは注記されるべきである。また、メモリコマンドが { コマンド , アドレス } の形式を取り、ここでは “ コマンド ” は実行されるべき命令を表わし、 “ アドレス ” は関連付けられたメモリのロケーションを表わしていることも注記されるべきである。

10

【 0 0 6 8 】

次に表 4 と図 1 2 に言及する。システムクロックサイクル C_1 の間、最初のコマンド { オープンページ , 1 0 0 0 } がアドレスデコーダ 3 0 2 で受信され、並列的に領域比較器 1 1 1 0 で受信される。この例では、アドレスデコーダ 1 1 0 2 はオープンページコマンドアドレス “ 1 0 0 0 ” を “ 1 0 0 ” および “ 4 0 0 ” としてデコードし、それを領域比較器 1 1 1 0 がメモリ領域 0 内に含まれることを決定する。オープンページコマンドは受信されるべき最初のコマンドなので、仮想バンク $B_0 - 1 3$ のどれにも “ ヒット ” は無く、対応する置換カウンタは “ 0 ” に設定される。この実施形態では、置換カウンタは擬似ランダムカウンティング方法に基づいて更新されるのに対し、他の実施形態では、ランダム・カウンティングか他の適切な方法が使用される。最初のコマンド { オープンページ , 1 0 0 0 } はオープンタイプコマンドなので、関連付けられた最小デルタ発行時間もデータオフセットも無く、したがってアドレス 1 0 0 0 のページは最初のコマンドクロックサイクル C_1 で開かれる。

20

【 0 0 6 9 】

次のシステムクロックサイクル C_2 の間に、 { リード , 1 0 0 0 } コマンドが制限ブロック 1 1 0 0 で受信され、それをアドレスデコーダ 1 1 0 2 が 1 0 0 および 4 0 0 としてデコードし (すなわち、メモリアドレスロケーション 1 0 0 0 に前のクロックサイクルでオープンされたページをリードすること)、これらの値が、領域比較器 1 1 1 0 に領域識別子を領域 1 に設定させる。しかしながらこの場合には、 B_0 レジスタに格納されている前の、あるいは他の言い方では “ 古いコマンド ” が B_0 において “ ヒット ” することになり、これにより、セクタは、 “ リード ” を “ 古いコマンド ” 入力として L U T 1 1 0 8 に出力する。他の入力としては、領域比較器 1 1 0 4 により発せられた領域標識子 「 領域 1 」 と、リードである “ 新しいコマンド ” 入力とを含む。L U T 1 1 0 8 は、格納された特性データを用いて、3 つのコマンドクロックサイクル C_3 の最小デルタ発行時間を生成する。これは、少なくとも 3 つのコマンドクロックサイクルが { ページオープン , 1 0 0 0 } コマンドの発行と、それに関連付けられた { リード , 1 0 0 0 } コマンドとを分離しなければならないということを示している。

30

40

【 0 0 7 0 】

このようにして、制限ブロック 1 1 0 0 において受信されたそれぞれのメモリコマンドパケットは、L U T 1 1 0 8 に格納された特性データにしたがって、少なくともある程度は直前に発行されたコマンドに基づいて処理される。

【 0 0 7 1 】

次に、本発明の特定の実施形態に従った、制限ブロックから受信されたコマンドの再順序付けを説明する。図 1 3 A ~ 図 1 3 C は、時刻表 1 3 0 2 , 1 3 0 4 であり、単純なコマンド再順序付けの例を通して、本発明の特定の実施形態に従ったメモリコマンドの再順序付けによって実現される利点のいくつかを例示するのに役立つ。それぞれの時刻表は 2 つの異なるメモリバンクに対応する 4 つの読み出しコマンドを示している。C M D 0 と C

50

MD 1は、関連付けられたメモリのバンク 1 に差し向けられたリードコマンドである。CMD 2 と CMD 3 は、関連付けられたメモリのバンク 2 に差し向けられたリードコマンドである。時刻表 1302 は、コマンドがシステムプロセッサからメモリコントローラによって受信される順序で、メモリコントローラとメモリを接続するコマンドバス上に配置されたメモリコマンドを示している。CMD 0 が時間帯 0 を占め、CMD 1 が時間帯 3 を占め、CMD 2 が時間帯 4 を占め、CMD 3 が時間帯 7 を占めている。それぞれの時間帯は 1 つのクロックサイクルを表わしている。

【0072】

上に述べたように、同じメモリバンクへのコマンドは、前に発行されたコマンドの処理を行うために、発行の間に最短の遅延が必要である。これは図 13A に、一对のコマンドの間の二つの時間帯によって表わされている。見てとれるように、図 13A に示された順序で 4 つのリードコマンドがメモリに送られたとすると、コマンドバスは 4 つの使用可能なクロックサイクル、すなわち時間帯 1、2、5、6 の間は、使われずに過ぎることになる。以下で論じるように、この非効率の少なくとも幾分かは、本発明に従ったコマンドの再順序付けによって改善されるであろう。

10

【0073】

図 13B と図 13C の時刻表 1304 と 1306 はそれぞれ、本発明の特定の実施形態に従った図 13A のコマンドの再順序付けと、それにより得られる利点の少なくともいくつかを図解している。この例においては、データバスのコンフリクトは簡便を期して考慮されていない。しかしながら以下に論じるように、メモリコマンドの効果的な再順序付けのためには、そのような考慮に注意が払われねばならない。CMD 2 と CMD 3 が CMD 0 と CMD 1 とは異なったメモリバンクに差し向けられているという事実のため、二対のコマンドの間にあるメモリ・アクセス待ち時間は問題とならず、無視してもよい。すなわち、コマンドは時刻表 1304 に示されたように配置し直すことができ、CMD 2 を CMD 0 の直後の時間帯 1 に置き、CMD 3 を CMD 1 の直後の時間帯 4 に置く。これは、異なったメモリバンクに差し向けられているという事実により、CMD 0 と CMD 2 の発行の間、CMD 1 と CMD 3 の発行の間には遅延は必要ないからである。しかしながら、図 13C に示したように、同じバンクに差し向けられたコマンドの対の間には最短の遅延時間、例えば 2 クロックサイクルが維持されねばならないことが理解されるだろう。すなわち、コマンドの再順序付けは同じメモリバンクへの連続するコマンド間の遅延時間を縮減する試みは含んでいない。

20

30

【0074】

コマンドの再順序付けの結果は図 13C に示されている。ここでは 4 つのコマンドが 5 クロックサイクル内に発行され、時間帯 2 だけが使われずに過ぎている。もちろん、さらに別のメモリバンクに向かう 5 番目のメモリコマンドが時間帯 2 に挿入され、コマンドバスが活用される効率をさらに極限まで増大させることは理解されるだろう。

【0075】

図 14 は、本発明の特定の実施形態に従って構成されたメモリコントローラの一部のブロックダイアグラムである。再順序付け回路 1400 はシステム・プロセッサから入ってくるメモリコマンドシーケンス、すなわち 1、2、3 というコマンドシーケンスを受け取る。特定の実施形態によれば、メモリコマンドは制限回路（示されていない）を介して再順序付け回路 1400 に伝送され、制限回路は上述のように、選択されたコマンドに、関連付けられたメモリの同一の論理バンクに差し向けられた他のコマンドに応じて、発行時間の制約を課す。コマンドは、コマンドキュー 1402 内で再順序付けされ、そこからコマンドはメモリに向けて発行される。この例においては、コマンドは 1、3、2 の順に再順序付けされている。

40

【0076】

オリジナルのメモリコマンドシーケンス、すなわち 1、2、3 は、データ読み取り回路 1406 内の FIFO メモリに格納される。FIFO 1404 内のシーケンスは、コマンドが元々メモリコントローラによって受け取られた順序に対応するように、メモリから受け

50

取ったデータを再順序付けするために使われる。しかしながら、プロセッサのいくつかは順序通りのデータを期待するのに対し、他のプロセッサは順序通りでないデータを期待するので、F I F O 1 4 0 4 のスイッチを必要に応じてオン、オフすることにより、いかなるタイプのデータ順序もサポートされうることが注記されるべきである。これが必要なのは、プロセッサがコマンドを元々メモリコントローラに伝送した順序に対応する順序でデータを受け取ることを“期待する”からである。

【 0 0 7 7 】

さらに、メモリからのデータはメモリコントローラによって、プロセッサがメモリコマンドを伝送する元のシーケンスに対応しないシーケンスで受け取られるかもしれないので、第3のシーケンスがデータキュー1408に格納される。このシーケンス(この例では3、2、1)は、コマンドシーケンス1、3、2に対応するデータがデータ読み取り回路1406に受け取られるであろう順序を表わしている。データキューシーケンスは、コマンドキューシーケンスとメモリの様々な論理バンクに関連付けされた既知の待ち時間に基づいて、再順序付け回路1400によって算出される。メモリがデータキュー1408に格納されたシーケンス(すなわち3、1、2)でメモリコントローラにデータを伝送するとき、データは読み取りデータバッファ1410に格納され、F I F O 1 4 0 4 内の情報とデータキュー1408に基づいて、元のコマンドシーケンスの順序に対応する順序、すなわち1、2、3でプロセッサに伝送するように、再順序付けされる。

【 0 0 7 8 】

図15は、本発明の特定の実施形態に従って構成されたメモリコントローラ内の再順序付け回路1500のブロック図である。再順序付け回路1500は、システムプロセッサから受け取ったコマンドを格納し再順序付けするコマンドキュー1502を含む。コマンドキュー1502は、メモリ内の同一の論理バンクに向かうコマンドに関連付けられたコマンド発行時間制約とデータバス使用制約を使って、それぞれのコマンドの発行時間を計算し、コマンドを発行し、発行されたコマンドをキューから外す。

【 0 0 7 9 】

データキュー1504は、発行されたメモリコマンドに対応するデータ発生時刻を表わすデータ要素を格納し、キューへの新規入力それぞれに対する新しいデータ発生時刻を計算し、対応するメモリトランザクションが完了したときにキューエントリを外す。

【 0 0 8 0 】

比較器マトリクス1506は、衝突検出機能を実行する。この機能では、コマンドキュー1502から発行準備のできているコマンドのデータ発生時刻(マルチプレクサ1508を介して伝達される)が、データキュー1504に表わされている前に発行されたコマンドのデータ発生時刻と比較される。衝突が検出されれば、コマンドの発行が延期される。

【 0 0 8 1 】

図16は、図15の再順序付け回路1500のさらに詳細なブロック・ダイヤグラムである。図17のダイヤグラムに示されているように、コマンドキュー1502は、6つのコマンドキュー要素1602を含み、そのそれぞれは特定のメモリコマンドに関して61ビットの情報を格納している。コマンドフィールド1702は、メモリコマンドを特定する40ビットメモリコマンドパケットを含む。コマンド発行時間(Cd)フィールド1704は6ビットのフィールドであり、コマンドが発行される前のクロックサイクルのデルタ時間を表示する。フィールド1704の値は、上述の制限回路によって決定され、メモリ内の同一の論理バンクに対応する最新のメモリコマンドに関係する。すなわち、Cdフィールドの値は同一のバンクへの2つのコマンド間の待ち時間を表わす。それぞれのバンクに必要な待ち時間に関する情報は制限回路に格納され、大部分はメモリの物理的特性によって決定される。コマンドキュー内で、Cdフィールドはそれぞれのクロックサイクルに対して決定されるが、いくつかの例外がある。例えば、同一の論理バンクへの連続するコマンド間の待ち時間は変更されない。したがって、特定のバンクに向けられたコマンドのためのCdフィールドがゼロになり発行されない場合、最初のコマンドが発行されるまで、同じバンクへの他のコマンドのCdフィールドはデクリメントされない。

【 0 0 8 2 】

データ発生時刻 (D d) フィールド 1 7 0 6 は 6 ビットのフィールドであり、コマンドキューからのメモリコマンドの発行と、対応するデータの転送の間のクロックサイクルのデルタ時間を示す。D d フィールド 1 7 0 6 はコマンドキュー内では変更されてはならない。コマンドIDフィールド 1 7 0 8 は 5 ビット・フィールドであり、コマンドパケット 1 7 0 2 内のコマンドを一対一対応に識別する。この情報は、コマンドやデータの再順序付けが効を奏するように、どれがどのパケットか、どのデータがどのパケットに対応するか見失わないようにするため、F I F O の対応する情報とデータキューとともに使用される。論理バンク (B) フィールド 1 7 1 0 は 3 ビットのフィールドであり、メモリ内のどの論理バンクにコマンドパケットが向けられているかを識別する。最後に、バースト標識 (D b) フィールド 1 7 1 2 は 1 ビットのフィールドであり、要求された、または書き込まれたデータが、1 つまたは 2 つのクロックサイクルを占めることを示す。

10

【 0 0 8 3 】

図 1 6 に戻って述べると、コマンドキューの動作はコマンドキューコントローラ 1 6 0 4 によって制御されている。コントローラ 1 6 0 4 は、どのコマンドキュー要素 1 6 0 2 が使用可能かを見失わないようにして、空き位置識別部 1 6 0 6 を介して、入ってくるコマンドの特定のキュー要素への挿入を制御する。コントローラ 1 6 0 4 はまた、対応するコマンドが発行されたときに、コマンドキュー要素の情報をデータキュー 1 5 0 4 に挿入するのを容易にする。特定の実施形態によれば、コマンドバスやデータバス上の空き時間スロットの利用可能性に関係なく、コマンドはコマンドキュー 1 5 0 2 に挿入される。

20

【 0 0 8 4 】

コマンドは、その C d カウントがゼロで、かつ、データバス上に衝突がない場合、マルチプレクサ 1 6 0 8 を介して、コマンドキュー要素 1 6 0 2 のどの 1 つからでもコマンドバスに発行されうる。すなわち、コマンドバスおよび/またはデータバス上の空き時間スロットが識別されねばならない。コマンドがリードやライトではない場合 (したがってデータバス・リソースを必要としない場合)、コマンドバス時間スロットのみが必要とされる。コマンドがリードやライトである場合には、コマンドバスとデータバス両方のスロットが必要とされる。

【 0 0 8 5 】

コントローラ 1 6 0 4 のゼロコンパレータ 1 6 1 0 は、最初の決定、すなわち C d = 0 かどうかを決定するために使われる。減算器 1 6 1 2 は、上記の例外すなわち発行できない特定のコマンドに対して C d = 0 であることがない場合に、それぞれのクロックサイクル毎に、それぞれのコマンドキュー要素 1 6 0 2 に対する C d カウントから “ 1 ” を減じるために使われる。その場合、キューコントローラ 1 6 0 4 は、すべてのキュー要素に対する C d フィールドと B フィールドを用いて、同一の論理バンクへの全コマンドに対する C d カウントがデクリメントするのを防ぐマスク信号 (M) を発する。

30

【 0 0 8 6 】

特定の実施形態によれば、C d = 0 である 2 つのキュー要素があった場合、優先順位の最も高いもの (例えば、最も古いもの) が発行される。アドレスシフタ 1 6 1 4 は、以下で図 1 8 を参照しつつ詳細に論じるように、キュー内のコマンドの優先順位を決定する。他の特定の実施形態によれば、新しいコマンドがコマンドキューに到着して、その C d カウントがすでにゼロである場合、それはマルチプレクサ 1 6 0 8 を介して直接メモリに伝送される。新しいコマンドは、その C d カウントがゼロでない場合、または優先順位が上位で C d = 0 でコマンドキューに格納された他のコマンドがある場合、コマンドキュー要素 1 6 0 2 に格納される。しかしながら、コマンドキューが空の場合には、新しいコマンドはただちに発行される (C d がゼロに等しい場合)。

40

【 0 0 8 7 】

リードコマンドやライトコマンドに関しては、衝突は、発行準備ができているコマンドを含むコマンドキュー要素 1 6 0 2 の D d フィールドと D b フィールドを使って検出される。コマンドに対応するデータ発生時刻と持続時間はマルチプレクサ 1 5 0 8 を介して比較

50

器マトリクス1506に伝送され、マルチプレクサ1508はキューコントローラ1604によって制御される。すなわち、キューコントローラ1604は、コマンド発行時間すなわちCdがゼロであるキュー要素のデータ発生時刻と持続時間（1または2クロックサイクル）を伝送するために、マルチプレクサ1508を制御する。持続時間は1または2クロックサイクルであり、これは、加算器1616が、Dbビットをデータ発生時刻Ddに加算することによって、Dd+1に対する“0”（1クロックサイクルを表わす）または“1”（2クロックサイクルを表わす）を導き出すことにより得られる。次に、データ発生時刻および持続時間は、比較器マトリクス1506で、データキュー1504に格納された5つの以前に発行されたコマンドのデータ発生時刻および持続時間と比較される。特定の実施形態によれば、比較器マトリクス1506は2×10個の平行比較器マトリクスを含む。

10

【0088】

図18は、図16のアドレスシフタ1614の特定の実施形態のブロック図である。上述のように、アドレスシフタ1614はコマンドの優先順位を決定する。これもまた上述のように、新しいコマンドは、空き位置認識部1606にしたがって、任意のフリーコマンドキュー要素1602に挿入される。新しいコマンドが挿入されるコマンドキュー要素1602のアドレスは、最初の空き位置（A0～A5）に最上位の優先順位で挿入される。結果として、アドレスシフタのA0ポジションは、未発行の最も古いコマンド用のキュー要素アドレスを格納する。コマンドがコマンドキューから発行されるとき、アドレスシフタ1614の対応するエントリが外され、下位の優先順位のコマンド用のアドレスが上位の優先順位のポジションに変更される。上述のように、コマンドキュー内のコマンドに対するCdカウントがゼロに達したとき、コマンドは発行される。しかしながら、Cd=0のコマンドが1つ以上ある場合には、最も古いコマンド、すなわちアドレスシフタ1614内のアドレスのポジションによって示される優先順位が最上位のコマンドが発行される。

20

【0089】

図16のデータキュー1504は5つのキュー要素1652を含み、そのそれぞれは図19によって図解されているように、以前に発行されたメモリコマンドに関する12ビットの情報を含んでいる。データ発生時刻（Dd）フィールド1902は、コマンドキューからのコマンドの発行と、対応するデータの受信との間のクロックサイクルでのデルタ時間を示す6ビットのフィールドである。それぞれのデータキュー要素1652に対するDdカウントは、その値がゼロに到達するまで、減算器1654の1つを用いてクロックサイクル毎にデクリメントされる。Dd=0のとき、対応するデータはデータバス上にある。したがって、任意の時刻にDd=0を持つデータキュー要素1652はただ1つであることが理解されるだろう。Ddカウントがゼロに到達した後、対応するデータキュー要素内の情報はデータキュー1504から外される。

30

【0090】

コマンドIDフィールド1904は、データが対応する発行済みコマンドを一对一对応に識別する5ビットのフィールドである。この情報は、コマンドがメモリコントローラに最初に伝送された元の順序に対応するようにデータを再順序付けするのに役立つ。最後に、バースト標識（Db）フィールド1906は、データが1つまたは2つのクロックサイクルを占めていることを示す1ビットのフィールドである。

40

【0091】

図16に戻って論じると上述のように、データキュー要素1652のそれぞれに対するデータ発生時刻（Dd）および持続時間は、発行準備ができていないコマンド、すなわちCd=0のコマンドキュー1502内のコマンド用のDdおよび持続時間と、比較器マトリクス1506で比較される。持続時間は、1または2クロックサイクルであり、これは、加算器1616が、Dbビットをデータ発生時刻Ddに加算することによって、Dd+1に対する“0”（1クロックサイクルを表わす）または“1”（2クロックサイクルを表わす）を導き出すことによって得られる。比較によりデータバス上に衝突がないことがわ

50

かれは、コマンドがコマンドキューから発行される。

【0092】

データキューコントローラ1658は、データキュー1504の動作を制御する。空き位置識別部1660は、コマンドキューコントローラ1604とともに、データキュー要素1652内に新たなデータキュー要素情報を挿入することを容易にする。空き位置識別部はまた、対応するメモリアクセスが完了したときにデータキュー要素1652から情報を外すのを容易にする。ゼロコンパレータ1662とバースト識別部1664は、いつデータキュー要素1652のどれかに対するDdがゼロになるか、そしていつデータ転送がデータバスを占有しなくなるか、そしてそれゆえいつ対応する情報がデータキューから外されるべきかを決定するのに使われる。

10

【0093】

本発明の別の特定の実施形態によれば、衝突検出は2次元配列のコンパレータとマルチプレクサの使用を通して、さらに複雑になる。このアプローチは、上述の1次元アプローチよりさらにシリコン集約的で、発行準備のできたコマンドに対する1つの要素だけでなく、コマンドキュー内の全要素を見る。これは、以前に発行されたコマンドについてだけでなく、データバス上のデータパケットの順序についても、コマンドのスケジューリングを行う。

【0094】

新しいコマンドを挿入するためには、コマンドパイプの発行予定部分における2つの連続したステージのそれぞれの組み合わせが、それらの間に新しいコマンドを挿入できるか調べるために比較されなければならない。この比較は、コマンドが挿入されうる範囲を、実際に決定する。この範囲は以下に示すとおりである。

20

【0095】

$CLEN_x$ = コマンドの長さ

$$T_{cstart} = t_{cA} + CLEN_A \quad (1)$$

$$T_{cend} = t_{cB} \quad (2)$$

【0096】

ここで、 t_{cA} と t_{cB} は、連続するパイプライン要素AとBの発行時間である。パイプライン要素Aはパイプライン要素Bに先立ち、したがってその発行時間は2つの中の低い方である。挿入があるとすれば、当然AとBの要素の間には少なくとも1つのオープンスロットがなければならない。したがって：

30

【0097】

$$N = T_{cend} - T_{cstart} + 1 \quad (3)$$

(ここで、N = エlement A B間の発行スロットの数)

$$LEN <= t_{cB} - t_{cA} - CLEN_A \quad (4)$$

【0098】

ハードウェアでは、以下の条件を単純に実装することは容易である：

【0099】

$$(t_{cB} - CLEN_A) - (t_{cA} + CLEN_A) = > 0 \quad (5)$$

【0100】

範囲のスタートポイントとエンドポイントも、関連付けられたデータスロットの可能な範囲を特定する。この範囲は、オーバーラップがないかどうか、新しい範囲がどのようになるかを調べるために、データパイプ内のそれぞれの連続する要素と比較されなければならない。この比較には5つの異なるケースが存在する。

40

【0101】

ケース0：

このケースでは、データスロット t_{dA} と t_{dB} によって画される範囲は、2つの連続する要素MとNの範囲の完全に外部にある。このケースでは、したがって：

【0102】

$$t_{dA} + CLEN_A = > t_{dN} \quad (6)$$

50

または、 $DLEN_X =$ データの長さとする、

$$t_{dB} \leq t_{dM} + DLEN_M \quad (7)$$

【0103】

MとNのペアの間には可能なデータスロットはない。

【0104】

ケース1：

このケースでは、データスロット t_{dA} と t_{dB} によって画される範囲は、2つの連続する要素MとNの範囲の完全に内部にある。このケースでは、したがって：

【0105】

$$t_{dA} + CLEN_A > t_{dM} + DLEN_M \quad (8)$$

10

かつ、

$t_{dB} - CLEN + DLEN \leq t_{dN}$ (ここで、 $CLEN$ はスロット内の新しいコマンドの長さ、 $DLEN$ はスロット内の新しいデータの長さ) (9)

【0106】

このケースにおける最も早い可能なデータスロット時間は、対応するコマンドの発行時間 $t_{cA} + CLEN_A$ を持つ $t_{dA} + LENA_A$ である。

【0107】

ケース2：

このケースでは、データスロット t_{dA} と t_{dB} によって画される範囲は、要素Mの長さに一致する。このケースでは、したがって：

20

【0108】

$$t_{dA} + CLEN_A < t_{dM} + DLEN_M \quad (10)$$

かつ、

$$t_{dB} - CLEN + DLEN > t_{dM} + DLEN_M, \text{ かつ、 } t_{dB} - CLEN + DLEN < t_{dM} \quad (11)$$

【0109】

このケースにおける最も早い可能なデータスロット時間は、対応するコマンドの発行時間 $t_{cM} + CLEN_M - DATA_OFFSET$ を持つ $t_{dM} + DLEN_M + 1$ である。ここで $DATA_OFFSET$ は、コマンド発行時間とデータ占有時間の間の時間である。

【0110】

30

ケース3：

このケースでは、データスロット t_{dA} と t_{dB} によって画される範囲は、要素Nの長さに一致する。このケースでは、したがって：

【0111】

$$t_{dA} + CLEN_A > t_{dM} + DLEN_M \quad (12)$$

かつ、

$$t_{dA} + CLEN_A + DLEN < t_{dN} \quad (13)$$

【0112】

したがって、このケースにおける最も早い可能なデータスロット時間は、対応するコマンドの発行時間 $t_{cA} + CLEN_A + 1$ を持つ $t_{dA} + CLEN_M$ である。このケースにはケース1も含まれていることは注記すべきである。

40

【0113】

ケース4：

このケースでは、データスロット t_{dA} と t_{dB} によって画される範囲は、要素MとNによって定義される範囲を包含する。このケースでは、したがって：

【0114】

$$t_{dA} + CLEN_A < t_{dM} + DLEN_M \quad (14)$$

かつ、

$$t_{dB} - LENA > C t_{dN} \quad (15)$$

【0115】

50

したがって、このケースにおける最も早い可能なデータスロット時間は、対応するコマンドの発行時間 $t_{cM} + CLEN_A + DATA_OFFSET$ を持つ $t_{dM} + DLEN_M$ である。ここで $DATA_OFFSET = t_{dA} - t_{cA}$ である。

【0116】

最も早い可能なスロットが常に選ばれるようなスケジューリングという目的に照らすと、ケース1とケース3が同一になることは明らかである。従って、これらを合体したケースはケース3である。同様に、ケース2とケース4は、要求される結果が $t_{dM} + LLEN_M$ であるので、同一である。このケースでは、 t_{dM} が t_{dA} と t_{dB} によって与えられる範囲に一致することが端的に示されねばならない。さらに、入ってくるコマンドに対する、最も早い可能な発行時間 (t_c) とデータスロット (t_d) が考慮されねばならない。それぞれの

10

コマンドパイプのペアに対して、それぞれのデータパイプで行われなければならない比較は以下の通りである：

【0117】

【数1】

```

if(((tcB - CLEN) => (tcA + CLENA)) && (tc <= (tcA + CLENA))) {
    if(((tdA + CLENA) <= (tdM + DLENM)) && ((tdB - DLEN - (tdM + DLENM)) >=
0)) {
        td = tdM + DLENM;
        tc = tcA - tdA + tdM + DLENM;
    }
    else if(((tdN - (tdA + CLENA + DLENA)) >= 0) && (tdA + CLENA) >= (tdM +
DLENM)) {
        td = tdA + CLENA;
        tc = tcA + CLENA;
    }
    else {
        td = IMPOSSIBLE;
        tc = IMPOSSIBLE;
    }
}
else if(((tcB - CLEN) => tc) && (tc > (tcA + CLENA))) {
    if((td < (tdM + DLENM)) && ((tdB - DLEN - (tdM + DLENM)) >= 0)) {
        td = tdM + DLENM;
        tc = tc - td + tdM + DLENM;
    }
    else if(((tdN - (td + DLEN)) >= 0) && td >= (tdM + DLENM)) {
        td = td;
        tc = tc;
    }
    else {
        td = IMPOSSIBLE;
        tc = IMPOSSIBLE;
    }
}
else {
    td = IMPOSSIBLE;
    tc = IMPOSSIBLE;
}
}

```

【 0 1 1 8 】

したがって、コマンドパイプに関して必要な演算は：

【 0 1 1 9 】

【 数 2 】

$$t_{cB} - CLEN \Rightarrow t_{cA} + CLEN_A$$

$$t_{cB} - CLEN \Rightarrow t_c$$

$$t_c + CLEN \leq t_{cB}$$

$$t_c > t_{cA} + CLEN_A$$

$$t_c \leq t_{cA} + CLEN_A$$

【 0 1 2 0 】

同様に、データパイプに関して必要な演算は：

【 0 1 2 1 】

【数 3】

$$\begin{aligned}
 t_{dA} + CLEN_A &\leq t_{dM} + DLEN_M \\
 t_{dA} + CLEN_A &\geq t_{dM} + DLEN_M \\
 t_{dB} - DLEN &\geq t_{dM} + DLEN_M \\
 t_{dN} &\geq t_{dA} + CLEN_A + DLEN_A \\
 t_d &< t_{dM} + DLEN_M \\
 t_{dN} &\geq t_d + DLEN \\
 t_d &\geq t_{dM} + DLEN_M
 \end{aligned}$$

10

【 0 1 2 2 】

従って、この決定ロジックは、上で定義されたコンパレータのマトリクスで構成されている。最適の選択は最も早いコマンド発行時間であり、これは単純な優先順位エンコーダによって決定される。

【 0 1 2 3 】

再順序付けパイプ制御ロジックは、コマンドパイプとデータパイプのそれぞれの要素について、どの演算がなされるべきかをダイナミックに決定しなければならない。

20

【 0 1 2 4 】

待機中のコマンドパイプでは、それぞれのパイプ要素は4つの動作が可能である。前の要素からの読み出し（パイプは前進する）、現内容の保持（パイプはホールドする）、次の要素からの読み出し（パイプはバックアップする）、入ってくるコマンドバスからの読み出しである。4つのケースによって定義されたパイプの様々なポイントに、多数組の条件が存在しうる。S L i M A C への発行元の要素は、要素0と定義され、それに対し発行から最も遠い要素は要素Mと定義される。再順序付け決定ロジックが、カレントパイプラインにおける最適の挿入点が要素N - 1とNの間であることを発見した場合には、要素Nへの挿入が行われる。

30

【 0 1 2 5 】

ケース1 - ホールド：

S L i M A C への発行や新しいコマンドの挿入がない場合、パイプはホールドする。

【 0 1 2 6 】

ケース2 - ホールド&インサート：

このケースでは、S L i M A C への発行はないが、新しいコマンドのパイプへの挿入がある。挿入が要素Nに生じるならば、パイプは要素0から要素N - 1までをホールドし、要素Nに挿入し、要素N + 1から要素Mまでをバックアップする。

【 0 1 2 7 】

ケース3 - 発行：

このケースでは、要素0からS L i M A C への発行があり、パイプの他の部分は、要素0が要素1の内容を、要素1が要素2の内容を、というように、要素M - 1が要素Mの内容を含むようになるまで、先送りされる。

40

【 0 1 2 8 】

ケース4 - 発行&挿入：

このケースでは、要素0からS L i M A C への発行があり、要素Nでの挿入がある。このケースでは、要素0からN - 2では先送り動作が行われ、要素N - 1では挿入動作が行われ、要素NからMはホールドされることになる。先送りは、直後の要素からのデータを格納することになる要素に行われ、要素Nでの挿入は（その要素はカレントパイプの要素N - 1と要素Nの間に挿入されることになる）、実際には、挿入された要素が、更新された

50

パイプのN - 1の位置に行くことを意味している。

【0129】

図20は、図15に示された衝突検出システム1500の別の実現である衝突検出システム2000を図解している。この実施形態では、衝突検出システム2000は、ターゲット応答制限に基づいた最適のコマンドシーケンスが得られるようにコマンドを再順序付けして、初期化コントローラとターゲットサブシステム間のデータ転送用の最適のスロットを決定する。コマンドの再順序付けはデータバス上の異なったデータパケット同志の衝突を引き起こしてはならないので、特定のコマンドに関係するコマンドデータ転送がデータコンフリクトを引き起こす場合この特定のコマンドの発行を差し止める衝突検出器2002が必要となる。この実施形態では、衝突検出システム2000は、コマンドキュー2004と結合される衝突検出器2002を備える。

10

【0130】

この実施形態では、衝突検出器2002は“発行予定の”コマンド(コマンドキュー2004に格納されている)と“発行済みの”コマンド(データキュー2006に格納されている)の間のすべての可能なデータ衝突を検出する。この実施形態では、それぞれマルチプレクサ2008と結合されているN個のコマンドキュー2004がある。N個のコマンドキュー2004のそれぞれは、発行予定のそれらのコマンドと、コマンドがターゲットデバイスに発行された後、データ転送がユニバーサルコントローラ104とターゲットデバイス(すなわち共有リソース)108間のデータバス上にいつ現れるかを示す時間ファクタ“ d_time_{ND} ”と、データ・バースト転送を示すバーストビット(b_{ND})と、リード/ライトビット($r w_{ND}$)と、を格納するように構成されている。この実施形態では、データキュー2006は、すでにターゲットデバイスに発行済みの要求に対して、データ転送がユニバーサルコントローラ104とターゲットデバイス(すなわち共有リソース)108間のデータバス上にいつ現れるかを示す時間ファクタ“ d_time_D ”を格納する。コマンドキュー2006はまた、バーストビット(b_{ND})、リード/ライトビット($r w_{ND}$)も格納する。

20

【0131】

より好ましい実施形態では、衝突検出システム2000は、発行予定のコマンドを格納し再順序付けするように構成されたキュー&リンクコントローラユニット2010を備える。キュー&コントローラユニット2010は、新しいコマンドの発行時間と、データがデータバス上に現れる時間も計算する。キュー&コントローラユニット2010はまた、コマンドキューから発行されたデータをデータキューに転送し、同時にコマンド発行後にコマンドキューからそれを外す。キュー&コントローラユニット2010はまた、メモリへのアクセス完了後にデータキューからデータ要素を外す。

30

【0132】

図21を参照すると、ターゲットデバイスへのすべてのリード/ライトコマンドは、データパケット転送に関連付けられている。ターゲットデバイスへのコマンド発行の前に、新しいデータパケットND(New Data)が、衝突を起こさずにデータキューへの挿入が可能か調べるために、そのタイミング情報に従ってチェックされる。図21に示されたこの例では、発行されたデータパケットDはすでにデータキューの中に位置を占めており、新しいデータパケットNDは発行済みデータパケットDに対して比較される。発行済みデータパケットDと新しいデータパケットNDが両方ともバーストアクセスを表わしていることには留意されたい。したがってこの例では、データ衝突を起こさずに発行済みデータパケットDに配慮して新しいデータパケットNDが位置を占める可能性が2つある。新しいデータパケットNDは発行済みデータパケットDの左側か右側に位置を占めることができる。

40

【0133】

この特定の例は、非バーストデータ転送とバーストデータ転送(すなわち4データストリーム)の両方をサポートするメモリコントローラの衝突検出を図解している。データバスの双方向的性質のため、1クロックサイクルが連続するリード-ライトまたはライト-リ

50

ード転送の間に挿入されなければならない。

【0134】

多くの可能な結果があることは留意されるべきであり、そのいくつかを以下に列挙する。

【0135】

1) NDがDの後または前に置かれる場合、衝突は起こらない。

【0136】

2) 連続するリード - ライトまたはライト - リードデータ転送の間には、1クロックサイクルが挿入されなければならない。コマンドキューとデータキューのすべての要素は、その動作が“リードデータ”(rw = 0)か“ライトデータ”(rw = 1)かを示す“rw”ビットを格納している。

10

【0137】

3) データパケットは、1データストリーム(非バースト転送)か4データストリーム(バースト転送)で構成されている。コマンドキューとデータキューのすべての要素は、その動作が“バースト転送”(バースト = 1)か“非バースト転送”(バースト = 0)かを示す“バースト”ビットを格納する。

【0138】

発行予定データパケットと発行済みデータパケットのペアのそれぞれにおいて、発行予定コマンドについて行われるべき比較は以下のとおりである：

【0139】

【数4】

20

```

// 変数の初期化
collision = NO;

// コマンドキューからの新規パケットの終端はバーストビットに
// 応じて決定される
if (burstND = 1) then d_time_endND = d_timeND + 3
else d_time_endND = d_timeND
10

for i=1 to last_element_from_Data_Queue
begin
// コマンドキューからのパケットの終端はバーストビットに
// 応じて決定される
if (burstD[i] = 1) then d_time_endD[i] = d_timeD[i] + 3
else d_time_endD[i] =
d_timeD[i]
20

// 2つの連続するリード/ライトまたはライト/リードの間に1クロック
// が発生しなければならない
if (rwD[i] = rwND) then
begin
d_time_endD[i] = d_time_endD[i] + 1
d_time_endND = d_time_endND + 1
end
30

// 衝突検出
if NOT((d_timeND > d_time_endD[i]) or (d_timeD[i] >
d_time_endND)) collision = YES;
end.

```

【0140】

本発明のさらに別の実施形態では、連続する2つのメモリアクセス間の時間を予測する装置と方法が開示されている。この装置と方法は、新しいコマンドに対する最も早い“コマンド発行時間”の高速計算を可能とするものである。図22について述べれば、特定のページに発行された最新のコマンドとそのメモリへの予測された次のアクセス間の時間を格納するN個のページタイマ2022を持つ予測システム2200を図解したものである。40

同じページへの次のアクセスは“クローズ”、“オープン”、“リード”、“ライト”の場合がある。入ってくる新しいコマンド(例えば、リード)は、特定のページへのアクセスが発行前にどのくらい待たなければならないかを表示する1つの特定のページタイマを選ぶ。この新しいコマンドは次に、タイミングルックアップテーブル2204の中から、このコマンド(リード)と同じページへの可能な次のアクセス(クローズ、オープン、ライト、リード)の間に挿入されるべき適切な内容を選ぶ。タイマの解像度は1クロックサイクルである。

【0141】

タイミングルックアップテーブル - データは、コマンド発行後データバス上のデータがど 50

のくらいのサイクル有効かを示す時間を格納する。新しいコマンドが非アクティブの場合には、すべてのサイクル毎に、すべてのページタイマの値が“0”に達するまでになる。

【0142】

次に図23を参照すれば、本発明のさらに別の実施形態において、本発明の実施形態に従ったデバイスアクセス優先順位決定部2302を持つデバイスコントローラ2300が示されている。この実施形態では、優先順位決定部2302は、リクエストコントローラユニット2304に結合されたデバイス要求をいくつでも受け取り格納するのに適したリクエストキュー2303を含む。リクエストコントローラユニット2304は一部には、リクエストキュー2303の任意の位置から特定の応答をフェッチし、フェッチした応答を複数の共有デバイス108のうちの適切な1つに伝達するために使われる。この実施形態では、優先順位決定部2302はまた、応答キュー2306を含む。この応答キューは、応答コントローラユニット2308に結合された共有デバイス108のいずれからでも応答を受け取り格納するように構成されており、応答コントローラユニット2308は、格納された応答から、要求しているデバイス102に送られる特定の応答を選択するのに利用される。

10

【0143】

好ましい実施形態では、図1Eに示されているように、要求とそれに関連付けられた応答とが同じID番号150を持つように、応答と要求がID番号150とそれぞれ関連付けられている。上述のように、ID番号150は、5つのデータビットを含み、その1番目と2番目のデータビットは、その特定の応答/要求が属する要求元デバイスのグループ(マルチプロセッサコンピューティング環境におけるプロセッサの1グループなど)を識別するグループセクタフィールド152である。さらに上述のように、要求番号フィールド(RN)153は、グループセクタフィールド152によって識別された要求元デバイスのグループに関連付けられた要求および/または応答の番号を表しており、例えば、同一の要求デバイスからの連続する要求が連続する要求番号フィールド153を持つように表わされている。

20

【0144】

動作中は、リクエストコントローラ2304とレスポンスコントローラ2308の双方は、グループ優先順位セクタレジスタ154と、動ロックカウンタレジスタ156と、再順序付けセクタ2312とをそれぞれ組み入れている。グループ優先順位セクタレジスタ154は、RN152によって識別された特定の要求/応答グループに関する優先順位情報を含む。それについて1つの実施形態では、“3”という値が最上位の優先順位を表わし、“0”という値が最下位の優先順位を表わしており、上位の優先順位の要求が優先順位の下の要求を飛び越えることができる。

30

【0145】

動ロック状況を回避するために、動ロックカウンタレジスタ156は、何個の連続する優先順位上位の要求(または応答)が、優先順位下位の要求(または応答)を飛び越せるかに関する情報を含んでいる。動ロックカウンタレジスタ156が、優先順位上位のリクエストが優先順位下位のリクエストを飛び越える状況においてのみアクティブとなることは注記されるべきである。実際、適当なキューに優先順位下位の要求(または応答)がない場合には、動ロックカウンタレジスタ156は非アクティブとなる。

40

【0146】

以上、本発明の実施形態の幾つかを詳細に説明してきたが、本発明の趣旨と範囲から逸脱することなく他の多くの特定の形式で本発明を実施し得ることは理解されよう。したがって、ここに示した例は説明のためのものであり、これらに限定されるものではない。本発明は、ここに示された細部に限定されることなく、前記特許請求の範囲内で変形可能である。

【図面の簡単な説明】

【図1A】 本発明の実施形態によるユニバーサルコントローラの一般的な使用例を示す図である。

50

【図 1 B】 図 1 A に示されるユニバーサルコントローラの具体的な使用例を示す図である。

【図 1 C】 本発明の実施形態によるユニバーサルコントローラに結合されたアドレス空間コントローラを示す図である。

【図 1 D】 図 1 C に示されるアドレス空間コントローラの具体的な使用例を示す図である。

【図 1 E】 本発明の実施形態による例示的な要求 / 応答識別番号を示す図である。

【図 2 A】 本発明の実施形態による一般的なユニバーサルコマンドを示す図である。

【図 2 B】 図 2 A と同種のユニバーサルコマンドで、メモリページリードコマンドに適する具体的なユニバーサルコマンドを示す図である。

【図 2 C】 図 2 B の例示的なコマンドにおいて、コマンド構成要素間に適当な時間間隔を設けることによって形成される一連のコマンドの例を示す図である。

【図 3】 本発明の実施形態によるリソースタグを示す図である。

【図 4】 本発明の実施形態に従って、ユニバーサルコントローラが共用リソースにアクセスするプロセスの詳細を表すフローチャートである。

【図 5】 本発明の実施形態に従って、ユニバーサルコントローラがリソースの状態および実行すべき動作の順序を決定するプロセスを示す図である。

【図 6】 本発明の実施形態による工程に基づき、ユニバーサルコントローラが連続する動作の間における適切な時間間隔を決定する工程を示す図である。

【図 7 A】 本発明の実施形態によるページヒット / ミスコントローラを示す図である。

【図 7 B】 本発明の実施形態によるページヒット / ミスコントローラを示す図である。

【図 8】 本発明の実施形態によるバンクアクセスコントローラを示す図である。

【図 9 A】 本発明の実施形態による例示的な S L D R A M 型マルチプロセッサシステムを示す図である。

【図 9 B】 図 9 A に示されるマルチプロセッサシステムによる例示的な S L D R A M バスの処理の流れを示すタイミング図である。

【図 10】 本発明の実施形態によるメモリコントローラのブロック図である。

【図 11】 本発明の実施形態による制限ブロックのブロック図である。

【図 12】 本発明の実施形態による例示的な S L D R A M コマンドのタイミング図である。

【図 13 A】 本発明の具体的な実施形態による、メモリコマンドの再順序付けの流れを示した図である。

【図 13 B】 本発明の具体的な実施形態によるメモリコマンドの再順序付けの流れを示した図である。

【図 13 C】 本発明の具体的な実施形態によるメモリコマンドの再順序付けの流れを示した図である。

【図 14】 本発明の具体的な実施形態により構成されたメモリコントローラの一部を示すブロック図である。

【図 15】 本発明の具体的な実施形態により構成された再順序付け回路のブロック図である。

【図 16】 図 15 の再順序付け回路のより詳細なブロック図である。

【図 17】 本発明の具体的な実施形態によるコマンドキュー要素の内容を示す図である。

【図 18】 アドレスシフタの具体的な実施形態を示すブロック図である。

【図 19】 本発明の具体的な実施形態によるデータキュー要素の内容を示す図である。

【図 20】 図 15 に示される衝突探知システムの別の実施例としての衝突探知システムを示す図である。

10

20

30

40

50

【図21】 ターゲットデバイスに対するリード/ライトコマンドのそれぞれがどのようにデータパケットの伝達に関連しているかを示す例示的なタイミング図である。

【図22】 特定のページに最後に発行されたコマンドとそのメモリへの予測される次期アクセスとの間の時間を記憶するNページタイマーを有する予測システムを示す図である。

【図23】 本発明の実施形態によるデバイスアクセス優先順位決定器を有するデバイスコントローラを示す図である。

【図24】 本発明の実施形態による制限ブロックにより実行されるスケジューリングプロセスを要約する表4を示す図である。

【図1A】

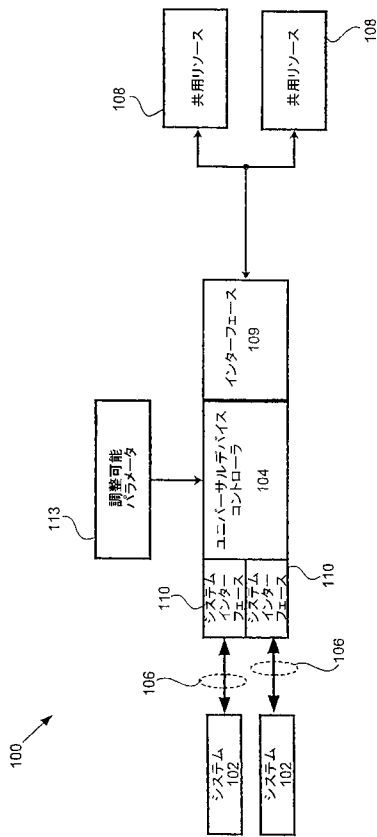


Fig. 1A

【図1B】

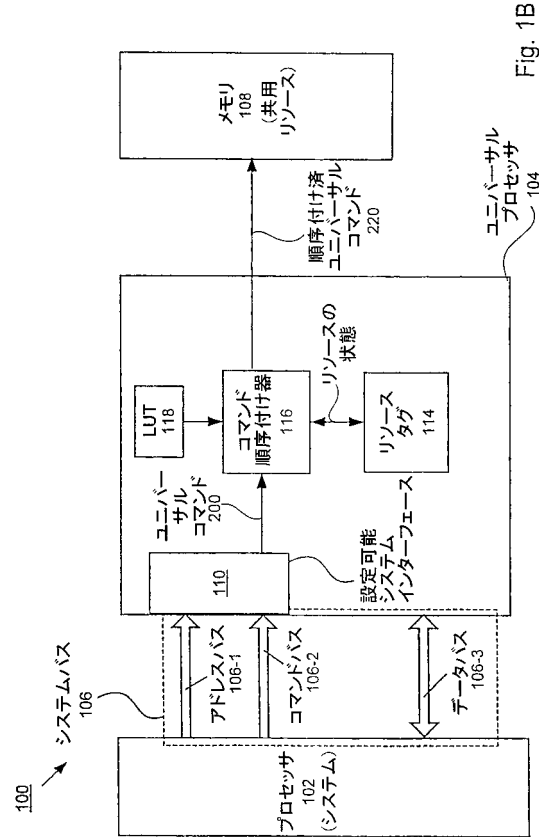


Fig. 1B

【図1C】

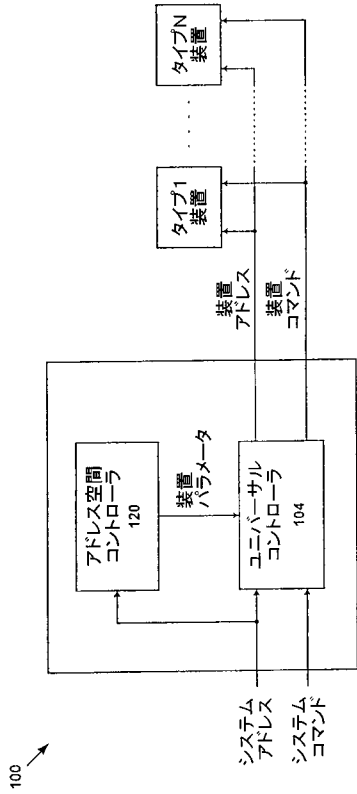


Fig. 1C

【図1D】

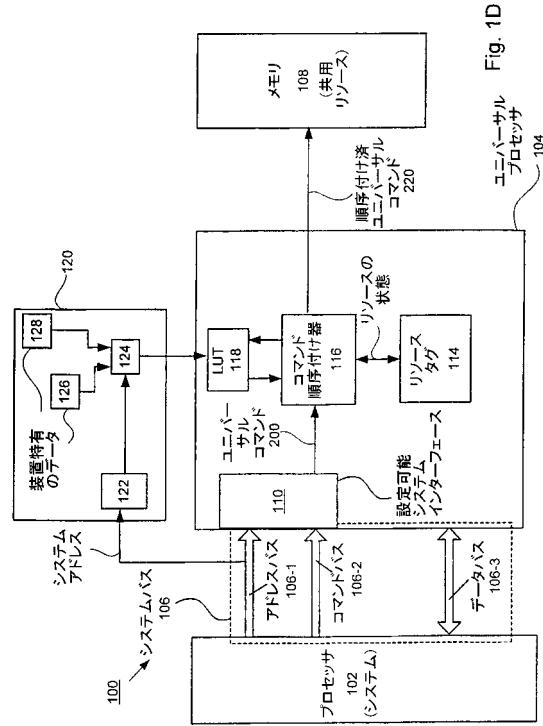
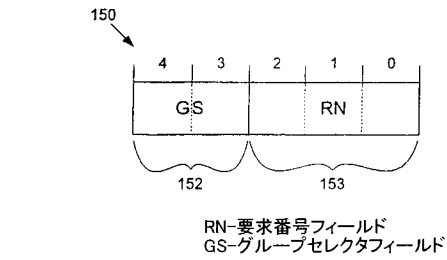


Fig. 1D

【図1E】



【図2A】

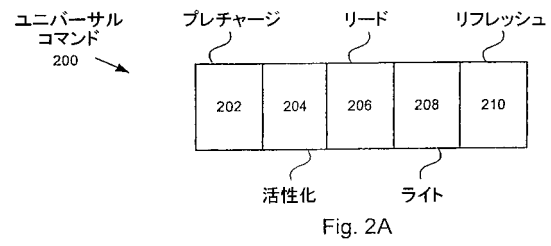


Fig. 2A

【図2B】

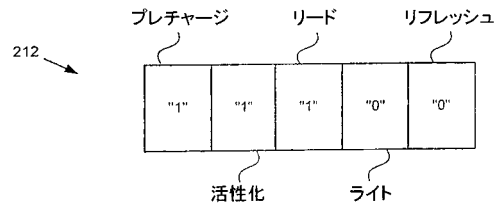


Fig. 2B

【図2C】

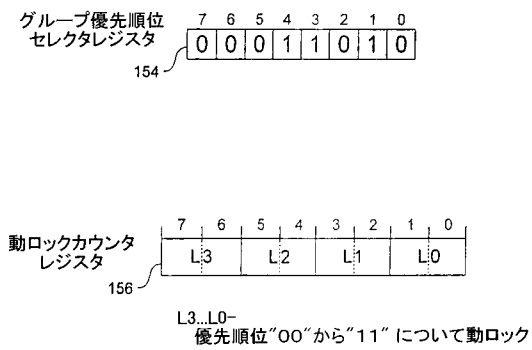


Fig. 1E

【図2C】

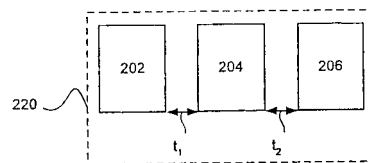


Fig. 2C

【図3】

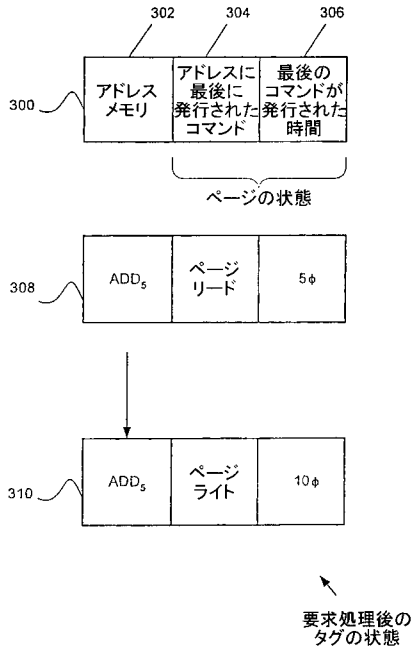


Fig. 3

【図4】

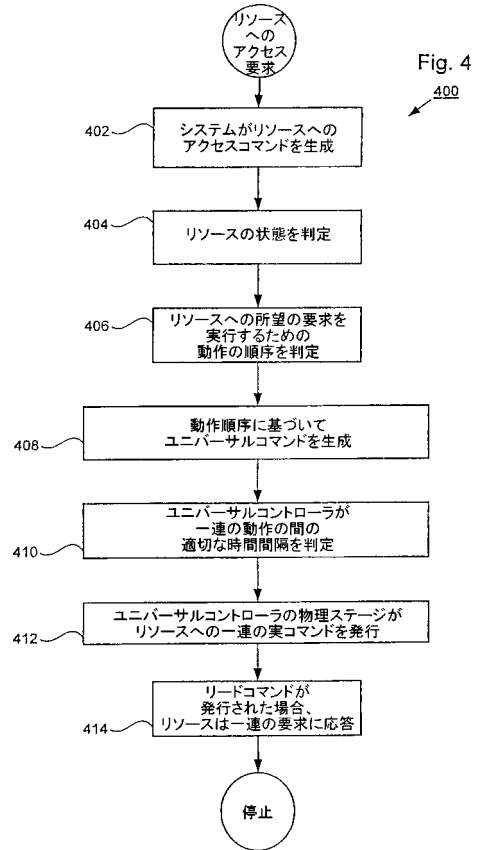


Fig. 4

【図5】

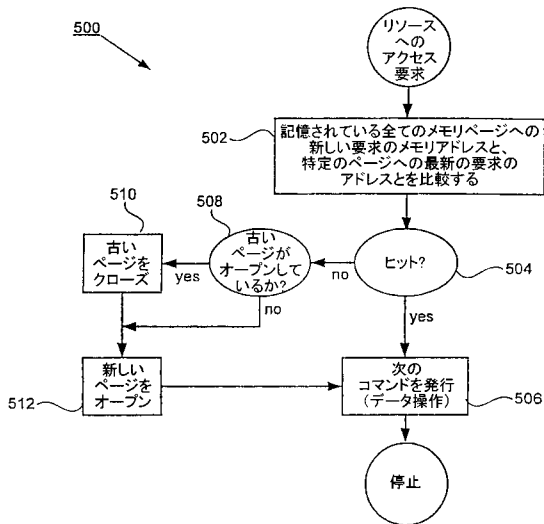


Fig. 5

【図6】

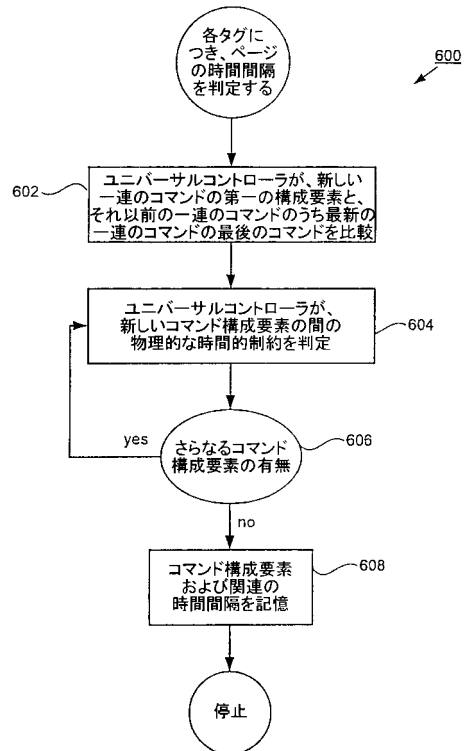


Fig. 6

【図7A】

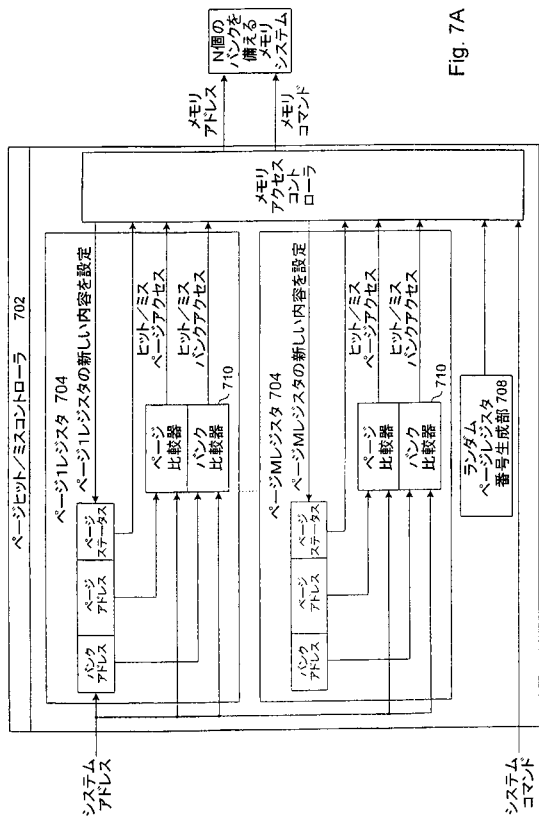


Fig. 7A

【図7B】

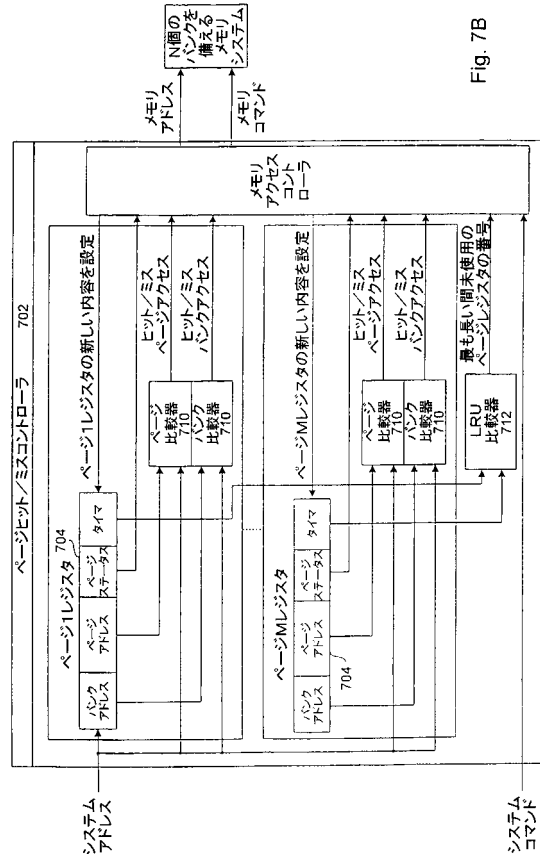


Fig. 7B

【図8】

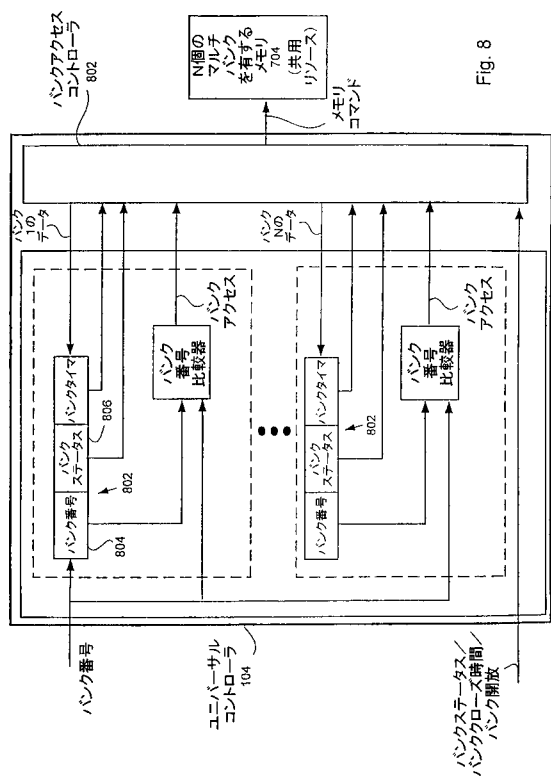


Fig. 8

【図9A】

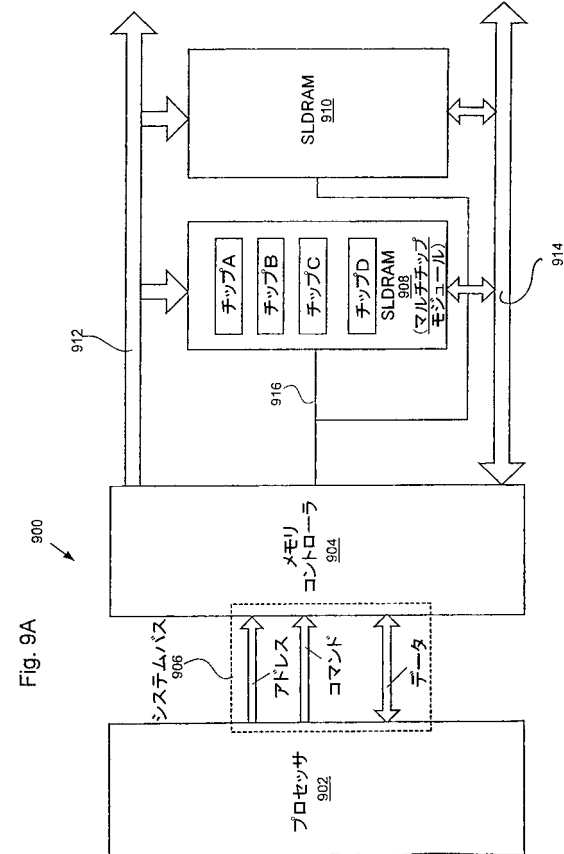


Fig. 9A

914

【 図 9 B 】

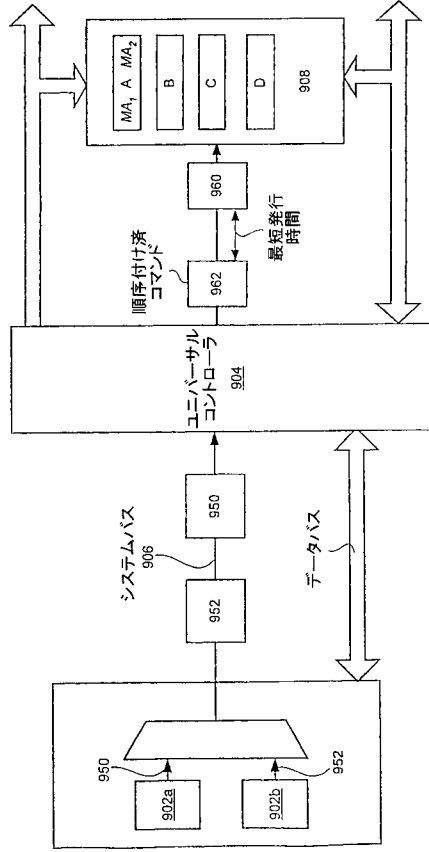


Fig. 9B

【 図 10 】

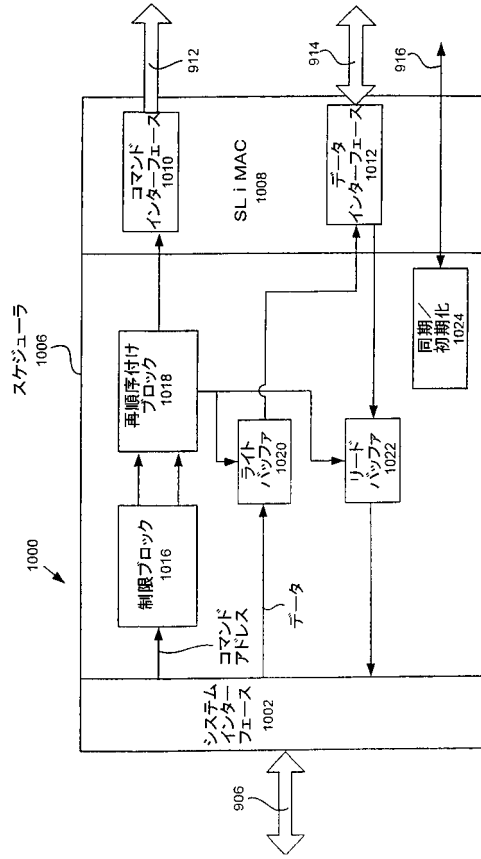


Fig. 10

【 図 11 】

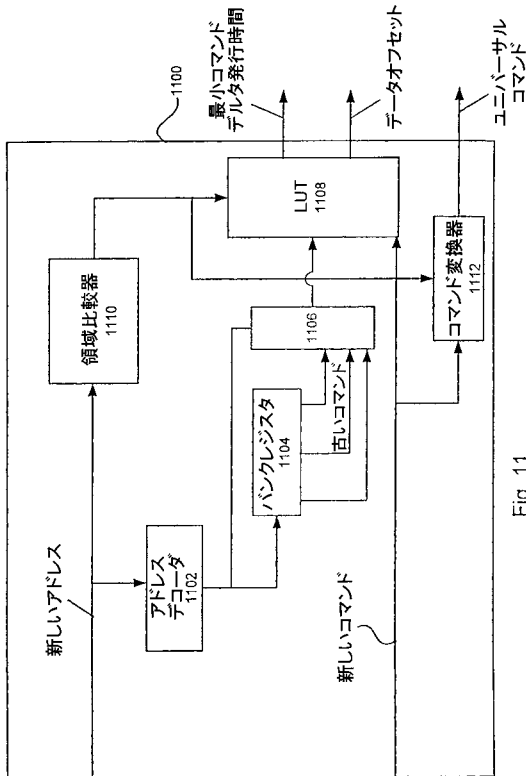


Fig. 11

【 図 12 】

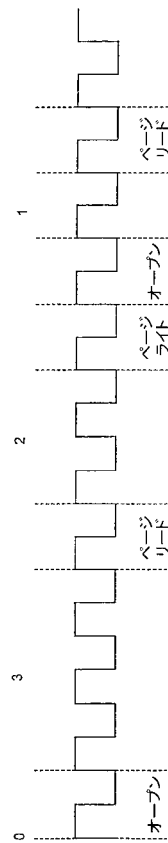


Fig. 12

【 図 1 3 A 】

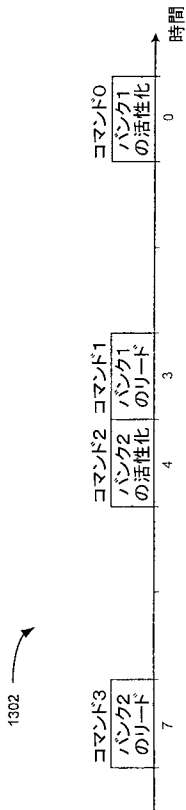


Fig. 13A

【 図 1 3 B 】

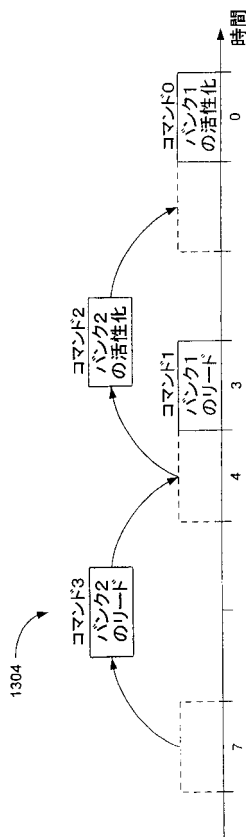


Fig. 13B

【 図 1 3 C 】

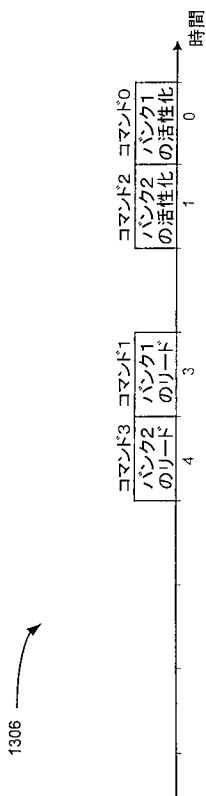


Fig. 13C

【 図 1 4 】

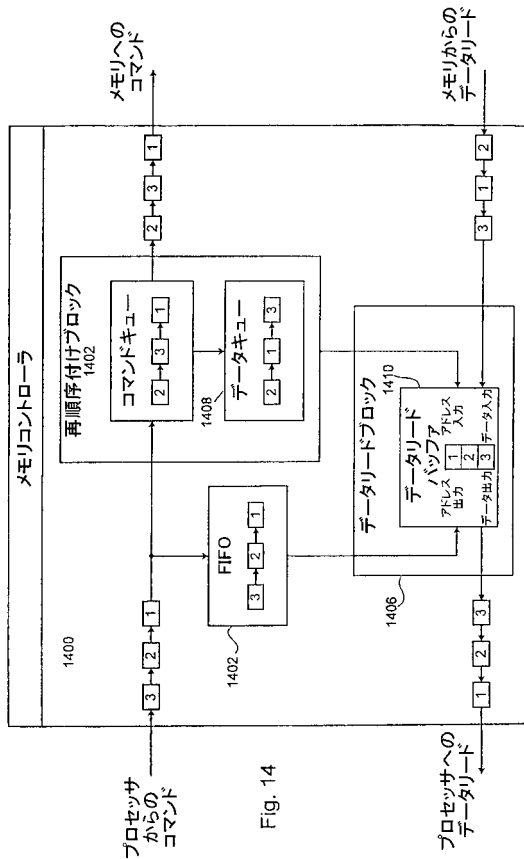


Fig. 14

【図15】

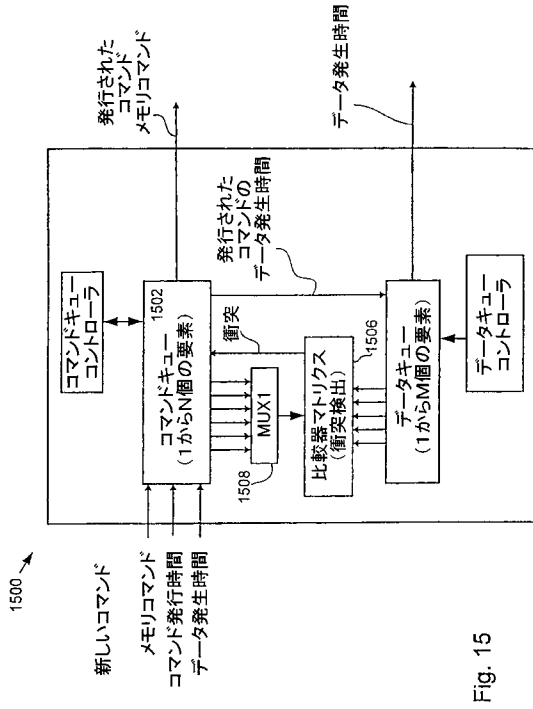


Fig. 15

【図16】

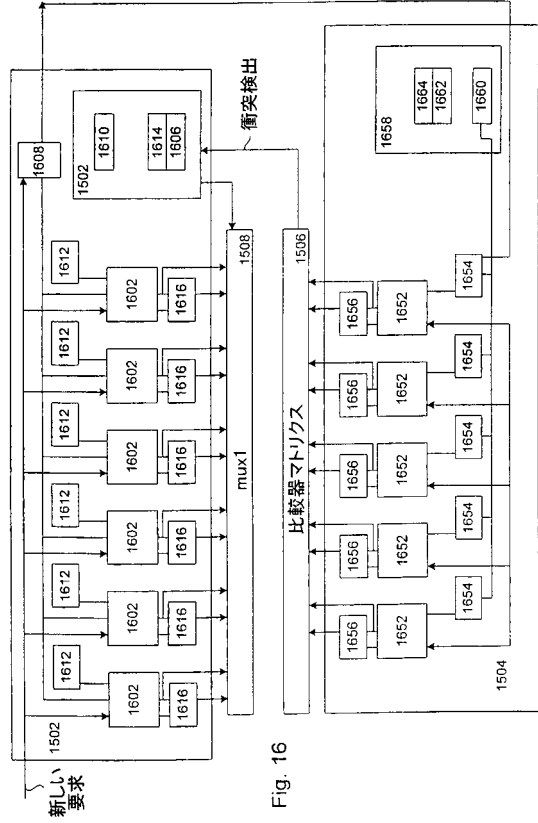


Fig. 16

【図17】

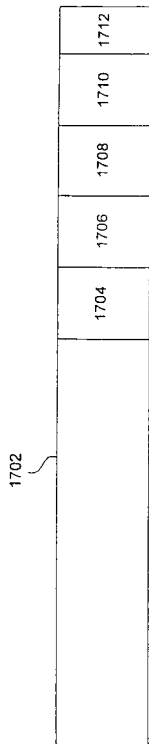


Fig. 17

【図18】

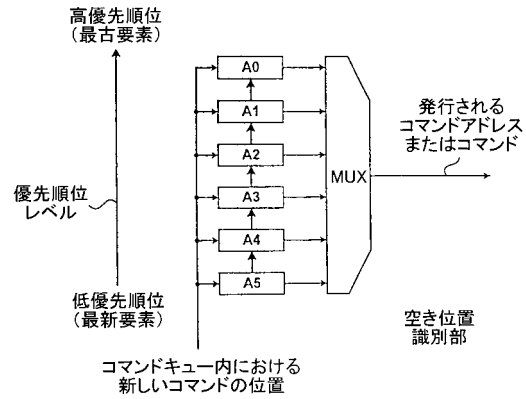


Fig. 18

【図19】

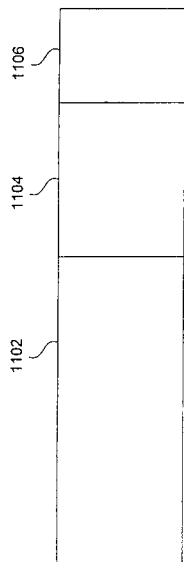


Fig. 19

【図21】

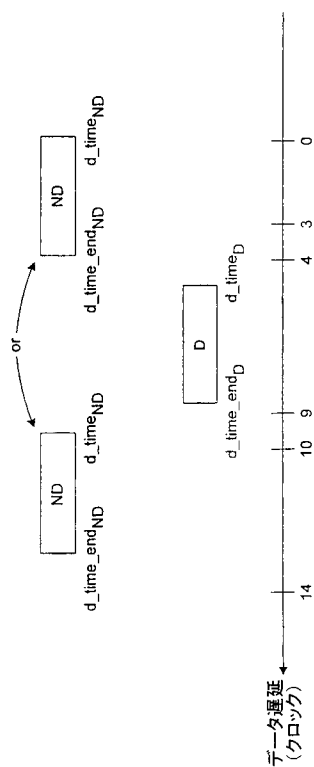


Fig. 21

【図20】

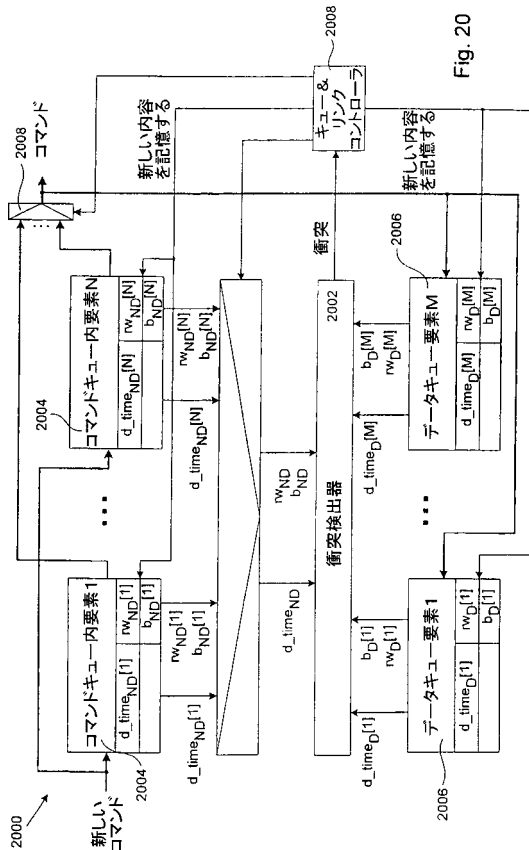


Fig. 20

【図22】

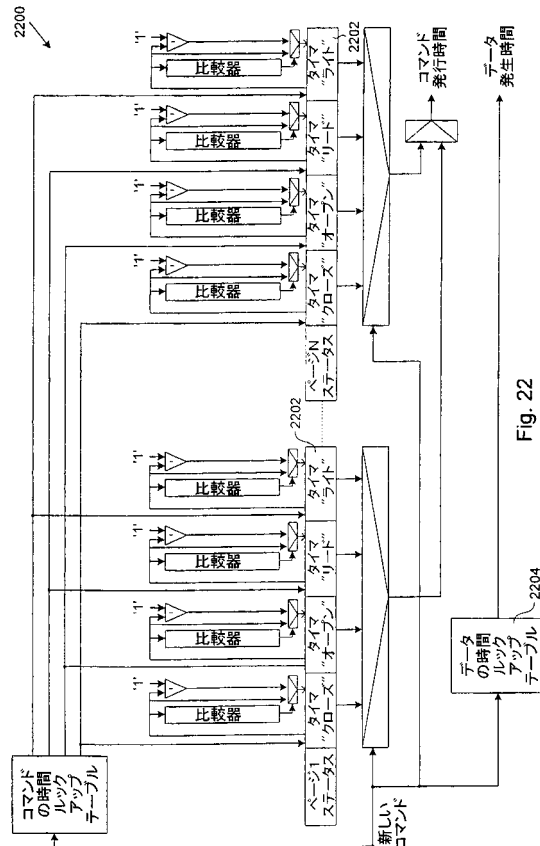


Fig. 22

【図 23】

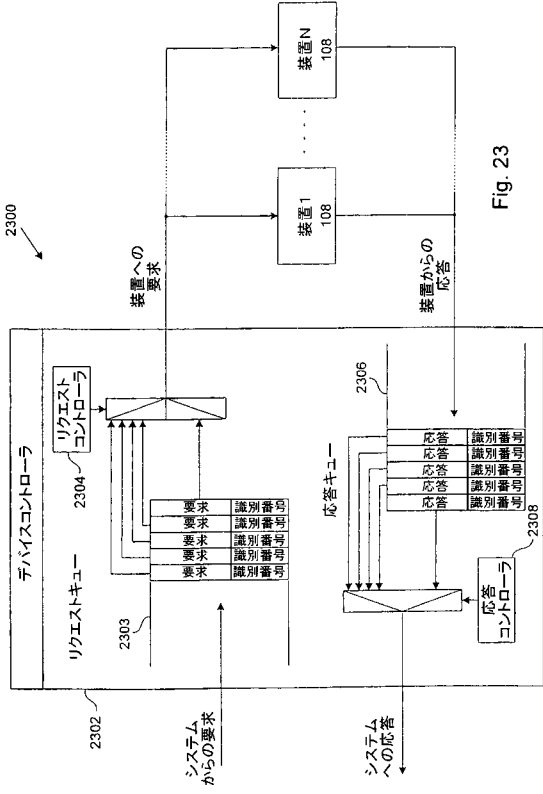


Fig. 23

【図 24】

表 4

入力コマンド	領域	仮想バンク				代替	LUT 入力	ΔT
		0	1	2	3			
オープン@1000	100	400			0		0	
ページリード@1000	100	400	100 オープン		1	オープン 領域0	3φ	
ページライト@1001	100	400	100 ページ リード		2	ページリード ページライト 領域0	2φ	
オープン@1001	10	200	100 ページ ライト		3	オープン	0	
ページリード@1001	100	400	100 ページ リード	200 オープン	0	ページライト ページリード 領域0	1	

Fig. 24

フロントページの続き

(72)発明者 シャベルスキー・ピオトル
アメリカ合衆国 カリフォルニア州 9 5 0 5 4 サンタ・クララ, ミル・クリーク・レーン, 5 9
0, アpartment 1 0 3

審査官 多賀 実

(56)参考文献 特開平 0 3 - 0 5 1 9 4 8 (J P , A)
特開平 1 0 - 2 2 8 4 1 7 (J P , A)
特開平 0 3 - 2 9 6 1 5 4 (J P , A)
特開昭 6 2 - 0 0 6 3 5 7 (J P , A)
特開平 0 2 - 0 2 8 7 3 0 (J P , A)
特開平 1 0 - 2 1 4 2 5 2 (J P , A)
特開平 0 4 - 0 2 4 8 4 3 (J P , A)
特開平 0 4 - 0 5 4 5 4 4 (J P , A)

(58)調査した分野(Int.Cl., DB名)

G06F 3/06

G06F 12/00-12/06

G06F 13/16-13/18

G06F 13/362