

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号

特許第7155103号

(P7155103)

(45)発行日 令和4年10月18日(2022.10.18)

(24)登録日 令和4年10月7日(2022.10.7)

(51)国際特許分類

F I

G 0 6 F 11/14 (2006.01)

G 0 6 F 11/14 6 1 0

G 0 6 F 12/00 (2006.01)

G 0 6 F 12/00 5 7 1 A

G 0 6 F 11/07 (2006.01)

G 0 6 F 11/07 1 4 0 N

G 0 6 F 11/07 1 9 3

請求項の数 15 (全17頁)

(21)出願番号	特願2019-501916(P2019-501916)	(73)特許権者	591016172
(86)(22)出願日	平成29年7月11日(2017.7.11)		アドバンスト・マイクロ・デバイス
(65)公表番号	特表2019-527424(P2019-527424		・インコーポレイテッド
	A)		ADVANCED MICRO DEVI
(43)公表日	令和1年9月26日(2019.9.26)		CES INCORPORATED
(86)国際出願番号	PCT/US2017/041559		アメリカ合衆国 9 5 0 5 4 カリフォル
(87)国際公開番号	WO2018/013584		ニア州、 サンタ クララ、 オーガスティ
(87)国際公開日	平成30年1月18日(2018.1.18)		ン ドライブ 2 4 8 5
審査請求日	令和2年7月6日(2020.7.6)	(74)代理人	100108833
(31)優先権主張番号	62/363,075		弁理士 早川 裕司
(32)優先日	平成28年7月15日(2016.7.15)	(74)代理人	100111615
(33)優先権主張国・地域又は機関			弁理士 佐野 良太
	米国(US)	(74)代理人	100162156
(31)優先権主張番号	15/375,076		弁理士 村雨 圭介
(32)優先日	平成28年12月9日(2016.12.9)	(72)発明者	ジェームズ アール . マグロ
	最終頁に続く		最終頁に続く

(54)【発明の名称】 DDRメモリエラーリカバリ

## (57)【特許請求の範囲】

## 【請求項1】

メモリアクセス要求を受信して記憶するコマンドキュー(520)と、

前記コマンドキューに接続され、メモリコマンドのシーケンスをメモリチャネル(130/140)に提供するアービタ(538)と、

前記メモリコマンドのシーケンスを前記メモリチャネル(130/140)に記憶し、メモリが第1タイプである場合、前記メモリチャネル(130/140)からの応答を未だ受信していないメモリアクセスコマンドを記憶し続け、前記メモリが第2タイプである場合、所定の信号の受信に基づいてエラーが発生した可能性がある、以前に発行された連続するメモリコマンドのシーケンスを記憶し続けるリプレイキュー(530)と、

を備えるメモリコントローラ(500)であって、

対応するメモリコマンドがエラーなく完了したことが応答によって示された場合、前記リプレイキュー(530)は、さらなるアクションを行わずに、前記対応するメモリコマンドを削除し、

エラーが応答によって示され、前記メモリチャネルが前記第1タイプのメモリを備える場合、前記リプレイキュー(530)は、前記エラーが発生した前記対応するメモリコマンドをリプレイし、前記メモリが前記第2タイプの場合、前記リプレイキューは、前記所定の信号の受信に基づいて前記エラーが発生した可能性がある、以前に発行された連続するメモリコマンドのシーケンスをリプレイする、

メモリコントローラ(500)。

10

20

## 【請求項 2】

前記対応するメモリコマンドのエラー訂正コード（ECC）を計算し、計算したECCを、前記メモリチャネル（130 / 140）から返されたECCと比較するために前記リプレイキュー（530）に接続されたECCチェックブロック（542）をさらに備える、請求項1のメモリコントローラ（500）。

## 【請求項 3】

メモリアクセス要求を受信し記憶するために前記コマンドキュー（520）に接続されたメモリインタフェースキュー（514）をさらに備え、

前記メモリインタフェースキュー（514）は、前記メモリチャネル（130 / 140）にディスパッチされたコマンドと、前記コマンドに応じて前記メモリチャネル（130 / 140）から受信した応答と、を提供するために前記リプレイキュー（530）に接続されている、

請求項1のメモリコントローラ（500）。

## 【請求項 4】

前記リプレイキュー（530）は、前記メモリインタフェースキュー（514）からの応答の受信に応じて、エラー訂正コードを計算し、前記エラー訂正コードを前記メモリチャネル（130 / 140）から受信したエラー訂正コードと比較することによって、前記応答にエラーが存在することを判別する、

請求項3のメモリコントローラ（500）。

## 【請求項 5】

前記リプレイキュー（530）は、前記メモリインタフェースキュー（514）からのエラーを含む応答の受信に応じて、前記対応するメモリコマンドと、前記対応するメモリコマンドの後に前記メモリインタフェースキュー（514）が前記メモリチャネル（130 / 140）に提供した追加のコマンドと、をリプレイする、

請求項3のメモリコントローラ（500）。

## 【請求項 6】

メモリアクセス要求を受信して記憶するコマンドキュー（520）と、

前記コマンドキュー（520）に接続され、メモリコマンドのシーケンスをメモリチャネル（130 / 140）に提供するアービタ（538）と、

前記コマンドキュー（520）に接続され、メモリアクセス要求を受信して記憶するメモリインタフェースキュー（514）と、

前記メモリインタフェースキュー（514）に接続され、前記メモリコマンドのシーケンスを前記メモリチャネル（130 / 140）に記憶し、メモリが第1タイプである場合、前記メモリチャネル（130 / 140）からの応答を未だ受信していないメモリアクセスコマンドを記憶し続け、前記メモリが第2タイプである場合、所定の信号の受信に基づいてエラーが発生した可能性がある、以前に発行された連続するメモリコマンドのシーケンスを記憶し続けるリプレイキュー（530）と、

を備えるメモリコントローラ（500）であって、

前記メモリコントローラ（500）は、エラーを検出しない限り通常状態（610）に留まり、前記メモリインタフェースキュー（514）は、前記通常状態において、前記アービタ（538）によって選択されたコマンドを前記コマンドキュー（520）から受信し続け、

前記メモリコントローラ（500）は、エラーと、前記メモリチャネルが前記第1タイプのメモリを備えることと、を検出したことに応じてリカバリ状態（630 / 640）に移行し、前記リプレイキュー（530）は、前記リカバリ状態において、前記エラーが発生した対応するメモリコマンドをリプレイし、前記メモリが前記第2タイプの場合、前記リプレイキューは、前記所定の信号の受信に基づいて前記エラーが発生した可能性がある、以前に発行された連続するコマンドのシーケンスをリプレイする、

メモリコントローラ（500）。

## 【請求項 7】

10

20

30

40

50

前記メモリコントローラ(500)は、前記リプレイキュー(530)が、メモリエラーを発生させた全てのメモリコマンドをさらなるエラーなしにリプレイするまで、前記リカバリ状態(630/640)に留まる、

請求項6のメモリコントローラ(500)。

【請求項8】

前記メモリコントローラ(500)は、コマンド及びアドレスエラーに応じてエラーリカバリ状態(640)に移行し、データエラーに応じてコマンドリプレイ状態(630)に移行する、

請求項6のメモリコントローラ(500)。

【請求項9】

前記メモリコントローラ(500)は、エラーが発生した可能性があるコマンドのリプレイに応じて、前記エラーリカバリ状態(640)から前記コマンドリプレイ状態(630)に移行し、

前記メモリコントローラ(500)は、全てのリプレイされたコマンドに対する応答をエラーなしに受信したことに応じて、前記コマンドリプレイ状態(630)から前記通常状態(610)に移行する、

請求項8のメモリコントローラ(500)。

【請求項10】

前記メモリコントローラ(500)は、エラーが発生した可能性があるコマンドのリプレイに応じて、前記エラーリカバリ状態(640)から前記コマンドリプレイ状態(630)に移行し、

前記メモリコントローラ(500)は、さらなるエラーに応じて、前記さらなるエラーがコマンド/アドレスエラーである場合に前記エラーリカバリ状態(640)に移行し、それ以外の場合に前記コマンドリプレイ状態(630)に移行する、

請求項8のメモリコントローラ(500)。

【請求項11】

前記メモリコントローラ(500)は、前記対応するメモリコマンドをリプレイして所定回数のエラーを受信したことに応じて、システムエラーを示す、

請求項7のメモリコントローラ(500)。

【請求項12】

メモリアクセス要求を提供するメモリアクセスエージェント(110/210/220)と、

メモリシステム(120)と、

請求項1～11の何れかのメモリコントローラ(290/500)と、

を備えるデータ処理システム(100)であって、

前記メモリコントローラ(290/500)は、前記メモリアクセスエージェント(110/210/220)及び前記メモリシステム(120)に接続されている、

データ処理システム(100)。

【請求項13】

メモリアクセス要求を受信して記憶することと、

記憶されたメモリアクセス要求からのメモリコマンドのシーケンスをメモリチャネル(130/140)に提供することと、

前記メモリコマンドのシーケンスを前記メモリチャネル(130/140)に記憶し、メモリが第1タイプである場合、前記メモリチャネル(130/140)からエラーなしの応答を未だ受信していないメモリコマンドを記憶し続け、前記メモリが第2タイプである場合、所定の信号の受信に基づいてエラーが発生した可能性がある、以前に発行された連続するメモリコマンドのシーケンスを記憶し続けることと、

前記メモリチャネル(130/140)から受信した応答にエラーが発生したか否かを検出することと、

メモリエラーが検出されなかったことに応じて通常状態(610)に留まり、前記通常

10

20

30

40

50

状態(610)の間、前記メモリが前記第1タイプである場合に、記憶されたメモリアクセス要求からコマンドを提供し続け、前記メモリが前記第2タイプである場合に、前記所定の信号の受信に基づいてエラーが発生した可能性がある、以前に発行された連続するメモリコマンドのシーケンスを提供し続けることと、

エラーの検出に応じてリカバリ状態(630/640)に移行し、前記リカバリ状態(630/640)の間、記憶されたメモリコマンドを、前記エラーが発生した対応するメモリコマンドからリプレイし始めることと、を含み、

前記リプレイすることは、前記メモリチャネルが前記第1タイプのメモリを備える場合に、前記エラーが発生した可能性がある前記対応するメモリコマンドをリプレイし、又は、前記メモリが前記第2タイプの場合に、前記所定の信号の受信に基づいて前記エラーが発生した可能性がある、以前に発行された連続するコマンドのシーケンスをリプレイすることを含む、

方法。

#### 【請求項14】

前記リカバリ状態(630/640)に移行する前に、少なくとも1つのメモリがパワーダウン状態から起動したという確認を待つことをさらに含む、

請求項13の方法。

#### 【請求項15】

メモリエラーを生じた全てのメモリコマンドがさらなるエラーなしにリプレイされるまで前記リカバリ状態(630/640)に留まることをさらに含む、

請求項13の方法。

#### 【発明の詳細な説明】

#### 【背景技術】

#### 【0001】

コンピュータシステムは、通常、安価で高密度のダイナミックランダムアクセスメモリ(DRAM)チップをメインメモリとして使用する。今日販売されている多くのDRAMチップは、半導体技術協会(JEDEC: Joint Electron Devices Engineering Council)によって公表された様々なダブルデータレート(DDR)DRAM規格と互換性がある。DDR DRAMは、様々な低電力モードを設けることによって、高性能及び低電力動作の両方を提供する。

#### 【0002】

最新のDDRメモリコントローラは、保留中のメモリアクセス要求を記憶するためのキューを維持して、保留中のメモリアクセス要求を、生成又は記憶された順序から外れた順序で選択するのを可能にし、これによって効率を高める。例えば、メモリコントローラは、所定のメモリバンクの同じ行への複数のメモリアクセス要求をキューから取り出し、これらをメモリシステムに連続的に発行することによって、現在の行をプリチャージして他の行をアクティブにするオーバーヘッドを回避することができる。

#### 【0003】

DDRメモリシステムは、例えばパリティビット、巡回冗長コード(CRC)、エラー検出コード(EDC)、又は、DDR DRAM内にデータと共に記憶される他のエラー訂正コード(ECC)等のように、エラー検出及びリカバリのための様々なメカニズムを含む。DDRメモリコントローラは、メモリアクセスを実行する場合に、記憶されたCRC又はECCビットを、メモリアクセスに伴って計算されたCRC又はECCビットと比較する。DDRメモリコントローラは、エラーの検出に応じて、可能であればエラーを訂正し、エラーをオペレーティングシステムに報告する。そして、オペレーティングシステムは、実行すべき是正措置を決定する。しかし、DRAMバスは、2400メガヘルツ(MHz)等の比較的高いクロックレートで動作し、クロックサイクルの両遷移でデータを転送する。データレートが高いため、DDRメモリバスは、偶発的なランダムエラー又はメモリバス上の「グリッチ」の影響を受けやすい。既知のDDRメモリコントローラは、これらのエラーを検出及び訂正するメカニズムを有するが、このようなメカニズムは、オ

10

20

30

40

50

ペレーティングシステムコールによって引き起こされるオーバーヘッドのためにシステムパフォーマンスを著しく低下させる。

【図面の簡単な説明】

【0004】

【図1】いくつかの実施形態による、データ処理システムのブロック図である。

【図2】図1のデータ処理システムでの使用に適したアクセラレーテッドプロセッシングユニット（APU：accelerated processing unit）のブロック図である。

【図3】いくつかの実施形態による、図2のAPUでの使用に適したメモリコントローラ及び関連する物理インタフェース（PHY）のブロック図である。

【図4】いくつかの実施形態による、図2のAPUでの使用に適した他のメモリコントローラ及び関連するPHYのブロック図である。

10

【図5】いくつかの実施形態による、メモリコントローラのブロック図である。

【図6】図5のメモリコントローラの有限状態マシンの動作に関連する状態図である。

【発明を実施するための形態】

【0005】

以下の説明では、異なる図面において同じ参照番号を使用して、類似又は同一の項目を示している。特に断りのない限り、「接続された」という単語及びこれに関連する動詞形は、当技術分野において既知の手段による直接接続及び間接的な電気接続の両方を含む。特に断りのない限り、直接接続の説明は、適切な形態の間接的な電気接続を使用する代替の実施形態をも意味している。

20

【0006】

一形態では、メモリコントローラは、コマンドキューと、アービタと、リプレイキューと、を含む。コマンドキューは、メモリアクセス要求を受信して記憶する。アービタは、メモリコマンドのシーケンスをメモリチャネルに提供するために、コマンドキューに接続されている。リプレイキューは、メモリコマンドのシーケンスをメモリチャネルに記憶し、メモリチャネルからの応答を未だ受信していないメモリアクセスコマンドを記憶し続ける。対応するメモリコマンドがエラーなく完了したことが応答によって示された場合、リプレイキューは、さらなるアクションを行わずに、対応するメモリコマンドを削除する。対応するメモリコマンドがエラーを伴って完了したことが応答によって示された場合、リプレイキューは、少なくとも当該対応するメモリコマンドをリプレイする。

30

【0007】

他の形態では、メモリコントローラは、コマンドキューと、アービタと、メモリインタフェースキューと、リプレイキューと、を含む。コマンドキューは、メモリアクセス要求を受信して記憶する。アービタは、メモリコマンドのシーケンスをメモリチャネルに提供するために、コマンドキューに接続されている。メモリインタフェースキューは、メモリアクセス要求を受信して記憶するために、コマンドキューに接続されている。リプレイキューは、メモリインタフェースキューに接続されており、メモリコマンドのシーケンスをメモリチャネルに記憶し、メモリチャネルからの応答を未だ受信していないメモリアクセスコマンドを記憶し続ける。メモリコントローラは、エラーを検出しない限り通常状態に留まり、メモリインタフェースキューは、通常状態において、アービタによって選択されたコマンドをコマンドキューから受信し続ける。メモリコントローラは、エラーの検出に応じてリカバリ状態に移行し、リプレイキューは、リカバリ状態において、少なくとも1つの対応するメモリコマンドをメモリインタフェースキューに送信することによって、少なくとも1つの対応するメモリコマンドをリプレイする。

40

【0008】

さらに他の形態では、データ処理システムは、メモリアクセスエージェントと、メモリシステムと、メモリコントローラと、を含む。メモリアクセスエージェントは、メモリアクセス要求を提供する。メモリコントローラは、メモリアクセスエージェント及びメモリシステムに接続されている。メモリコントローラは、コマンドキューと、アービタと、リプレイキューと、を含む。コマンドキューは、メモリアクセス要求を受信して記憶する。

50

アービタは、メモリコマンドのシーケンスをメモリシステムに提供するために、コマンドキューに接続されている。リプレイキューは、メモリコマンドのシーケンスをメモリチャネルに記憶し、メモリチャネルからの応答を未だ受信していないメモリアクセスコマンドを記憶し続ける。対応するメモリコマンドがエラーなく完了したことが応答によって示された場合、リプレイキューは、さらなるアクションを行わずに、対応するメモリコマンドを削除する。対応するメモリコマンドがエラーを伴って完了したことが応答によって示された場合、リプレイキューは、少なくとも当該対応するメモリコマンドをリプレイする。

#### 【 0 0 0 9 】

さらに他の形態では、方法は、メモリアクセス要求を受信して記憶することを含む。記憶されたメモリアクセス要求からのメモリコマンドのシーケンスがメモリチャネルに提供され、メモリチャネルからエラーなしの応答を未だ受信していないメモリコマンドが記憶され続ける。メモリエラーが発生したか否かは、メモリチャネルから受信した応答において検出される。メモリコントローラは、メモリエラーが検出されないことに応じて、通常状態に留まり、通常状態の間、記憶されたメモリアクセス要求の中からコマンドが提供され続ける。エラーの検出に応じてリカバリ状態に移行し、リカバリ状態の間、記憶されたメモリコマンドは、エラーが発生した対応するメモリコマンドからリプレイし始める。

#### 【 0 0 1 0 】

図 1 は、いくつかの実施形態による、データ処理システム 1 0 0 のブロック図である。データ処理システム 1 0 0 は、概して、アクセラレーテッドプロセッシングユニット ( A P U ) の形態のデータプロセッサ 1 1 0 と、メモリシステム 1 2 0 と、周辺機器相互接続エクスプレス ( P C I e ) システム 1 5 0 と、ユニバーサルシリアルバス ( U S B ) システム 1 6 0 と、ディスクドライブ 1 7 0 と、を含む。データプロセッサ 1 1 0 は、データ処理システム 1 0 0 の中央処理装置 ( C P U ) として動作し、現代のコンピュータシステムにおいて有用な様々なバス及びインタフェースを提供する。これらのインタフェースには、2つのダブルデータレート ( D D R x ) メモリチャネルと、P C I e リンクへの接続用の P C I e ルートコンプレックスと、U S B ネットワークへの接続用の U S B コントローラと、S A T A ( Serial Advanced Technology Attachment ) 大容量記憶デバイスへのインタフェースと、が含まれる。

#### 【 0 0 1 1 】

メモリシステム 1 2 0 は、メモリチャネル 1 3 0 と、メモリチャネル 1 4 0 と、を含む。メモリチャネル 1 3 0 は、本例において別々のランクに対応する代表的な D I M M 1 3 4 , 1 3 6 , 1 3 8 を含む、D D R x バス 1 3 2 に接続されたデュアルインラインメモリモジュール ( D I M M ) のセットを含む。同様に、メモリチャネル 1 4 0 は、代表的な D I M M 1 4 4 , 1 4 6 , 1 4 8 を含む、D D R x バス 1 4 2 に接続された D I M M のセットを含む。

#### 【 0 0 1 2 】

P C I e システム 1 5 0 は、データプロセッサ 1 1 0 内の P C I e ルートコンプレックスに接続された P C I e スイッチ 1 5 2 と、P C I e デバイス 1 5 4 と、P C I e デバイス 1 5 6 と、P C I e デバイス 1 5 8 と、を含む。P C I e デバイス 1 5 6 は、システム基本入出力システム ( B I O S ) メモリ 1 5 7 に接続されている。システム B I O S メモリ 1 5 7 は、例えば読み出し専用メモリ ( R O M ) 、フラッシュ E E P R O M ( electrically erasable programmable R O M ) 等の様々な不揮発性メモリタイプの何れかであってもよい。

#### 【 0 0 1 3 】

U S B システム 1 6 0 は、データプロセッサ 1 1 0 内の U S B マスタに接続された U S B ハブ 1 6 2 と、U S B ハブ 1 6 2 にそれぞれ接続された代表的な U S B デバイス 1 6 4 , 1 6 6 , 1 6 8 と、を含む。U S B デバイス 1 6 4 , 1 6 6 , 1 6 8 は、例えばキーボード、マウス、フラッシュ E E P R O M ポート等のデバイスであってもよい。

#### 【 0 0 1 4 】

ディスクドライブ 1 7 0 は、S A T A バスを介してデータプロセッサ 1 1 0 に接続され

10

20

30

40

50

ており、オペレーティングシステム、アプリケーションプログラム、アプリケーションファイル等のための大容量ストレージを提供する。

【0015】

データ処理システム100は、メモリチャネル130及びメモリチャネル140を設けることによって、最新のコンピューティングアプリケーションでの使用に適している。各メモリチャネル130、140は、例えばDDRバージョン4（DDR4）、低電力DDR4（LPDDR4）、グラフィックスDDRバージョン5（GDDR5）及び高帯域幅メモリ（HBM）等の最新のDDRメモリに接続されてもよいし、将来のメモリ技術に適応されてもよい。これらのメモリは、高いバス帯域幅及び高速動作を提供する。同時に、これらは、ラップトップコンピュータ等のバッテリー駆動アプリケーションの電力を節約する低電力モードも設けており、組み込み型サーマルモニタリングも提供する。

10

【0016】

図2は、図1のデータ処理システム100での使用に適したAPU200のブロック図である。APU200は、概して、中央処理装置（CPU）コアコンプレックス210と、グラフィックスコア220と、ディスプレイエンジン230のセットと、メモリ管理ハブ240と、データファブリック250と、周辺コントローラ260のセットと、周辺バスコントローラ270のセットと、システム管理ユニット（SMU）280と、メモリコントローラ290のセットと、を含む。

【0017】

CPUコアコンプレックス210は、CPUコア212と、CPUコア214と、を含む。本例において、CPUコアコンプレックス210は2つのCPUコアを含むが、他の実施形態では、CPUコアコンプレックス210は任意の数のCPUコアを含んでもよい。各CPUコア212、214は、制御ファブリックを形成するシステム管理ネットワーク（SMN）及びデータファブリック250に対して双方向に接続されており、メモリアクセス要求をデータファブリック250に提供することができる。各CPUコア212、214は、単一コアであってもよいし、例えばキャッシュ等の特定のリソースを共有する2つ以上の単一コアを有するコアコンプレックスであってもよい。

20

【0018】

グラフィックスコア220は、例えば、頂点処理、フラグメント処理、シェーディング、テクスチャブレンド等のグラフィックス操作を、高度に統合された並列形式で実行することの可能な高性能グラフィックス処理ユニット（GPU）である。グラフィックスコア220は、SMN及びデータファブリック250に対して双方向に接続されており、メモリアクセス要求をデータファブリック250に提供することができる。これに関して、APU200は、CPUコアコンプレックス210及びグラフィックスコア220が同じメモリ空間を共有するユニファイドメモリアーキテクチャ、又は、CPUコアコンプレックス210及びグラフィックスコア220がメモリ空間の一部を共有するメモリアーキテクチャをサポートしてもよいが、グラフィックスコア220は、CPUコアコンプレックス210がアクセスできない専用のグラフィックスメモリを使用してもよい。

30

【0019】

ディスプレイエンジン230は、グラフィックスコア220によって生成されたオブジェクトをレンダリング及びラスターライズして、モニタに表示する。グラフィックスコア220及びディスプレイエンジン230は、メモリシステム120の適切なアドレスに一樣に変換されるために共通のメモリ管理ハブ240に対して双方向に接続されており、メモリ管理ハブ240は、かかるメモリアクセスを生成し、メモリシステムから返された読出しデータを受信するために、データファブリック250に対して双方向に接続されている。

40

【0020】

データファブリック250は、任意のメモリアクセスエージェントとメモリコントローラ290との間でメモリアクセス要求及びメモリ応答をルーティングするためのクロスバースイッチを含む。また、データファブリック250は、システム構成に基づくメモリアクセスの宛先と、仮想接続毎のバッファとを判断するためのシステムメモリマップであっ

50

て、BIOSによって定義されたシステムメモリマップを含む。

【0021】

周辺コントローラ260は、USBコントローラ262と、SATAインタフェースコントローラ264と、を含み、これらの各々が、システムハブ266及びSMNバスに対して双方向に接続されている。これらの2つのコントローラは、APU200で使用可能な周辺コントローラの単なる例示である。

【0022】

周辺バスコントローラ270は、システムコントローラ（即ち「サウスブリッジ」（SB））272と、PCIeコントローラ274と、を含み、これらの各々が、入出力（I/O）ハブ276及びSMNバスに対して双方向に接続されている。また、I/Oハブ276は、システムハブ266及びデータファブリック250に対して双方向に接続されている。したがって、例えば、CPUコアは、データファブリック250がI/Oハブ276を介してルーティングするアクセスを通じて、USBコントローラ262、SATAインタフェースコントローラ264、SB272、又は、PCIeコントローラ274内のレジスタをプログラムすることができる。

【0023】

SMU280は、APU200上のリソースの動作を制御し、それらの間の通信を同期させるローカルコントローラである。SMU280は、APU200上の様々なプロセッサのパワーアップシーケンシングを管理し、リセット、イネーブル及び他の信号を介して複数のオフチップデバイスを制御する。SMU280は、APU200の各コンポーネントにクロック信号を提供するために、図2に示されていない1つ以上のクロック源（例えば位相同期ループ（PLL）等）を含む。また、SMU280は、様々なプロセッサ及び他の機能ブロックの電力を管理し、CPUコア212、214及びグラフィックスコア220から測定された電力消費値を受信して、適切な電力状態を判断してもよい。

【0024】

また、APU200は、様々なシステムモニタリング及び省電力機能を実装する。特に、1つのシステムモニタリング機能は、サーマルモニタリングである。例えば、SMU280は、APU200が高温になると、CPUコア212、214及び/又はグラフィックスコア220の周波数及び電圧を低減させてもよい。APU200が非常に高温になった場合には、APU200が完全にシャットダウンされてもよい。サーマルイベントは、SMU280によって、外部センサからSMNバスを介して受信されてもよく、SMU280は、これに応じてクロック周波数及び/又は電源電圧を低下させてもよい。

【0025】

図3は、いくつかの実施形態による、図2のAPU200での使用に適したメモリコントローラ300及び関連する物理インタフェース（PHY）330のブロック図である。メモリコントローラ300は、メモリチャネル310と、パワーエンジン320と、を含む。メモリチャネル310は、ホストインタフェース312と、メモリチャネルコントローラ314と、物理インタフェース316と、を含む。ホストインタフェース312は、メモリチャネルコントローラ314を、スケーラブルデータポート（SDP）を介してデータファブリック250に双方向に接続する。物理インタフェース316は、メモリチャネルコントローラ314を、DDR-PHYインタフェース仕様（DFI）に準拠するバスを介してPHY330に双方向に接続する。パワーエンジン320は、SMNバスを介してSMU280に双方向に接続されており、APB（Advanced Peripheral Bus）を介してPHY330に双方向に接続されており、メモリチャネルコントローラ314にも双方向に接続されている。PHY330は、例えば図1のメモリチャネル130又はメモリチャネル140等のメモリチャネルに対する双方向接続を有する。メモリコントローラ300は、単一のメモリチャネルコントローラ314を使用した単一のメモリチャネル用のメモリコントローラの例示であり、以下にさらに説明するメモリチャネルコントローラ314の動作を制御するためのパワーエンジン320を有する。

【0026】

10

20

30

40

50



図4は、いくつかの実施形態による、図2のA P U 2 0 0での使用に適した別のメモリコントローラ400及び関連するP H Y 4 4 0, 4 5 0のブロック図である。メモリコントローラ400は、メモリチャネル410, 420と、パワーエンジン430と、を含む。メモリチャネル410は、ホストインタフェース412と、メモリチャネルコントローラ414と、物理インタフェース416と、を含む。ホストインタフェース412は、メモリチャネルコントローラ414を、S D Pを介してデータファブリック250に双方向に接続する。物理インタフェース416は、D F I仕様に準拠しており、メモリチャネルコントローラ414をP H Y 4 4 0に双方向に接続する。メモリチャネル420は、ホストインタフェース422と、メモリチャネルコントローラ424と、物理インタフェース426と、を含む。ホストインタフェース422は、メモリチャネルコントローラ424を、別のS D Pを介してデータファブリック250に双方向に接続する。物理インタフェース426は、D F I仕様に準拠しており、メモリチャネルコントローラ424をP H Y 4 5 0に双方向に接続する。パワーエンジン430は、S M Nバスを介してS M U 2 8 0に双方向に接続されており、A P Bを介してP H Y 4 4 0, 4 5 0に双方向に接続されており、メモリチャネルコントローラ414, 424にも双方向に接続されている。P H Y 4 4 0は、例えば図1のメモリチャネル130等のメモリチャネルに対する双方向接続を有する。P H Y 4 5 0は、例えば図1のメモリチャネル140等のメモリチャネルに対する双方向接続を有する。メモリコントローラ400は、2つのメモリチャネルコントローラを有するメモリコントローラの例示であり、共有のパワーエンジン430を使用して、以下にさらに説明するように、メモリチャネルコントローラ414及びメモリチャネルコントローラ424の各々の動作を制御する。

#### 【0027】

図5は、いくつかの実施形態による、メモリコントローラ500のブロック図である。メモリコントローラ500は、メモリチャネルコントローラ510と、電力コントローラ550と、を含む。メモリチャネルコントローラ510は、インタフェース512と、キュー514と、コマンドキュー520と、アドレス生成器522と、コンテンツアドレス可能メモリ(C A M)524と、リプレイキュー530と、リフレッシュロジックブロック532と、タイミングブロック534と、ページテーブル536と、アービタ538と、エラー訂正コード(E C C)チェックブロック542と、E C C生成ブロック544と、データバッファ(D B)546と、を含む。

#### 【0028】

インタフェース512は、外部バスを介したデータファブリック250との第1双方向接続と、出力と、を有する。メモリコントローラ500において、この外部バスは、「A X I 4」として知られている、英国ケンブリッジのA R M H o l d i n g s, P L Cによって仕様化されたアドバンストエクステンシブルインタフェースバージョン4と互換性があるが、他の実施形態では、他のタイプのインタフェースであってもよい。インタフェース512は、F C L K(又はM E M C L K)ドメインとして知られる第1クロックドメインから、U C L Kドメインとして知られるメモリコントローラ500の内部の第2クロックドメインへのメモリアクセス要求を変換する。同様に、キュー514は、U C L Kドメインから、D F Iインタフェースに関連するD F I C L Kドメインへのメモリアクセスを提供する。

#### 【0029】

アドレス生成器522は、データファブリック250からA X I 4バスを介して受信したメモリアクセス要求のアドレスを復号化する。メモリアクセス要求は、正規化されたアドレスとして表わされる物理アドレス空間内のアクセスアドレスを含む。アドレス生成器522は、正規化されたアドレスを、メモリシステム120内の実際のメモリデバイスをアドレス指定し、関連するアクセスを効率的にスケジュールするのに使用可能なフォーマットに変換する。このフォーマットは、メモリアクセス要求を特定のランク、行アドレス、列アドレス、バンクアドレス及びバンクグループに関連付ける領域識別子を含む。システムB I O Sは、起動時に、メモリシステム120内のメモリデバイスにクエリしてその

10

20

30

40

50

サイズ及び構成を判断し、アドレス生成器 5 2 2 に関連する構成レジスタのセットをプログラムする。アドレス生成器 5 2 2 は、構成レジスタに記憶された構成を使用して、正規化されたアドレスを適切なフォーマットに変換する。コマンドキュー 5 2 0 は、データ処理システム 1 0 0 内のメモリアクセスエージェント（例えば、CPU コア 2 1 2, 2 1 4 及びグラフィックスコア 2 2 0 等）から受信したメモリアクセス要求のキューである。コマンドキュー 5 2 0 は、アドレス生成器 5 2 2 によって復号化されたアドレスフィールドと、アクセスタイプ及びサービス品質（QoS）識別子を含むメモリアクセスをアービタ 5 3 8 が効率的に選択するのを可能にする他のアドレス情報と、を記憶する。CAM 5 2 4 は、例えばライトアフターライト（WAW）及びリードアフターライト（RAW）順序付けルール等の順序付けルールを実施するための情報を含む。

10

#### 【0030】

リプレイキュー 5 3 0 は、例えば、アドレス及びコマンドパリティ応答、DDR4 DRAM の書込み巡回冗長検査（CRC）応答、又は、GDDR5 DRAM の書込み及び読出し CRC 応答等の応答を待つアービタ 5 3 8 によって取り出されたメモリアクセスを記憶するための一時的なキューである。リプレイキュー 5 3 0 は、ECC チェックブロック 5 4 2 にアクセスして、返された ECC が正しいか否か又はエラーを示しているか否かを判別する。リプレイキュー 5 3 0 は、何れかのサイクルでパリティ又は CRC エラーの場合にアクセスがリプレイされるのを可能にする。

#### 【0031】

リフレッシュロジック 5 3 2 は、メモリアクセスエージェントから受信した通常の読出し及び書込みメモリアクセス要求とは別に生成される様々なパワーダウン、リフレッシュ、及び、終端抵抗（ZQ）較正サイクルのためのステートマシンを含む。例えば、メモリバンクがプリチャージパワーダウンにある場合には、リフレッシュサイクルを実行するために定期的に起動されなければならない。リフレッシュロジック 5 3 2 は、オートリフレッシュコマンドを定期的に生成して、DRAM チップ内のメモリセルのチャージオフストレージキャパシタのリークによって生じるデータエラーを防止する。さらに、リフレッシュロジック 5 3 2 は、ZQ を定期的に較正して、システム内の熱変化によるオンダイ終端抵抗のミスマッチを防止する。また、リフレッシュロジック 5 3 2 は、どの場合に DRAM デバイスを別のパワーダウンモードにするのかを決定する。

20

#### 【0032】

アービタ 5 3 8 は、コマンドキュー 5 2 0 に双方向に接続されており、メモリチャネルコントローラ 5 1 0 の中心部分である。アービタ 5 3 8 は、メモリバスの利用を改善するために、インテリジェントなアクセススケジューリングによって効率を改善する。アービタ 5 3 8 は、タイミングブロック 5 3 4 を使用して、コマンドキュー 5 2 0 内の特定のアクセスの発行に適しているか否かを DRAM タイミングパラメータに基づいて判断することによって、適切なタイミング関係を実施する。例えば、各 DRAM は、同じバンクへの起動コマンド間の最小指定時間（「 $t_{RC}$ 」として知られる）を有する。タイミングブロック 5 3 4 は、リプレイキュー 5 3 0 に双方向に接続されており、このタイミングパラメータ及び JEDEC 仕様で指定された他のタイミングパラメータに基づいて適格性を判断するカウンタのセットを維持する。ページテーブル 5 3 6 は、リプレイキュー 5 3 0 に双方向に接続されており、アービタ 5 3 8 のメモリチャネルの各バンク及びバンクのアクティブページに関する状態情報を維持する。

30

40

#### 【0033】

ECC 生成ブロック 5 4 4 は、インタフェース 5 1 2 から受信した書込みメモリアクセス要求に応じて、書込みデータに従って ECC を計算する。DB 5 4 6 は、受信したメモリアクセス要求の書込みデータ及び ECC を記憶する。アービタ 5 3 8 が、メモリチャネルにディスパッチするための対応する書込みアクセスを選ぶと、DB 5 4 6 は、結合した書込みデータ/ECC をキュー 5 1 4 に出力する。

#### 【0034】

電力コントローラ 5 5 0 は、アドバンストエクステンシブルインタフェースバージョン 1

50

( A X I ) へのインタフェース 5 5 2 と、 A P B インタフェース 5 5 4 と、パワーエンジン 5 6 0 と、を含む。インタフェース 5 5 2 は、 S M N への第 1 双方向接続であって、図 5 に別に示された「 E V E N T \_ n 」と付されたイベント信号を受信するための入力を含む第 1 双方向接続と、出力と、を含む。 A P B インタフェース 5 5 4 は、インタフェース 5 5 2 の出力に接続された入力と、 A P B を介して P H Y に接続するための出力と、を有する。パワーエンジン 5 6 0 は、インタフェース 5 5 2 の出力に接続された入力と、キュー 5 1 4 の入力に接続された出力と、を有する。パワーエンジン 5 6 0 は、構成レジスタ 5 6 2 のセットと、マイクロコントローラ (  $\mu$  C ) 5 6 4 と、セルフリフレッシュコントローラ ( S L F R E F / P E ) 5 6 6 と、信頼性のある読出し / 書込みトレーニングエンジン ( R R W / T E ) 5 6 8 と、を含む。構成レジスタ 5 6 2 は、 A X I バスを介してプ  
 ログラムされており、メモリコントローラ 5 0 0 内の様々なブロックの動作を制御するための構成情報を記憶する。したがって、構成レジスタ 5 6 2 は、図 5 に詳細に示されていないこれらのブロックに接続された出力を有する。セルフリフレッシュコントローラ 5 6 6 は、リフレッシュロジック 5 3 2 によるリフレッシュの自動生成に加えて、リフレッシュの手動生成を可能にするエンジンである。信頼性のある読出し / 書込みトレーニングエンジン 5 6 8 は、 D D R インタフェース読出しレイテンシトレーニング及びループバックテスト等の目的のために、連続的なメモリアクセスストリームをメモリ又は I / O デバイスに提供する。

10

#### 【 0 0 3 5 】

メモリチャネルコントローラ 5 1 0 は、関連するメモリチャネルへのディスパッチのためにメモリアクセスを選択することを可能にする回路を含む。アドレス生成器 5 2 2 は、所望のアービトレーションの決定を行うために、アドレス情報を、メモリシステム内のランク、行アドレス、列アドレス、バンクアドレス及びバンクグループを含むプリデコードされた情報に復号化し、コマンドキュー 5 2 0 は、プリデコードされた情報を記憶する。構成レジスタ 5 6 2 は、受信したアドレス情報をアドレス生成器 5 2 2 がどのように復号するのかを決定するために、構成情報を記憶する。アービタ 5 3 8 は、復号化されたアドレス情報と、タイミングブロック 5 3 4 によって示されたタイミング適格性情報と、ページテーブル 5 3 6 によって示されたアクティブページ情報と、を使用して、例えば Q o S 要件等の他の基準を遵守しながらメモリアクセスを効率的にスケジューリングする。例えば、アービタ 5 3 8 は、メモリページを変更するのに必要なプリチャージ及びアクティブ  
 化コマンドのオーバーヘッドを避けるために、オープンページへのアクセスを優先し、或るバンクへのオーバーヘッドアクセスを、他のバンクへの読出し及び書込みアクセスをインタリーブすることによって隠す。特に、アービタ 5 3 8 は、通常動作中に、異なるページを選択する前にプリチャージされる必要があるまで、異なるバンク内のページオープン  
 を維持することを決定してもよい。

20

30

#### 【 0 0 3 6 】

アービタ 5 3 8 は、タイミングブロック 5 3 4 を使用して、保留中のアクセスのタイミング適格性を判別し、次に、効率性及び公平性の両方を保証する基準のセットに基づいて、コマンドキュー 5 2 0 から適格なアクセスを選択する。アービタ 5 3 8 は、効率性及び公平性の両方を保証する 2 つのメカニズムをサポートする。第 1 に、アービタ 5 3 8 は、  
 コマンドキュー 5 2 0 に記憶されたメモリアクセス要求の属性と、書き込みが許可されているのと同時に読み出しが許可される条件を制御するためのプログラム可能な閾値と、を調べることによって読み出し / 書き込みトランザクション管理を実行して効率性及び公平性の両方を保証する。第 2 に、アービタ 5 3 8 は、特定のタイプのアクセスのストリークがメモリバスを無期限に保持するのを許可しないように保証するストリークカウンタを含む。これらの 2 つのメカニズムについて説明する。

40

#### 【 0 0 3 7 】

( D D R メモリエラーリカバリ )

上述したように、リプレイキュー 5 3 0 は、アドレス及びコマンドパリティ応答、 D D R 4 D R A M の書き込み巡回冗長検査 ( C R C ) 応答、又は、 G D D R 5 D R A M の書

50

き込み及び読み出しCRC応答等の応答を待っているアービタ538によって選択されたメモリアクセスを記憶する一時的なキューである。リプレイキュー530は、ECCチェックブロック542にアクセスして、返されたECCが正しいかどうか又はエラーを示しているかどうかを判別する。リプレイキュー530は、これらのサイクルのうち1つのサイクルにパリティ又はCRCエラーがあった場合に、アクセスがリプレイされることを可能にする。また、リプレイキュー530は、現在のDDR DRAMで利用可能なエラー報告メカニズムを利用して、リプレイに関する決定を行う。メモリエラーが通常一時的であって、メモリチャネルが直ぐにリカバリすると仮定することによって、リプレイキュー530は、冗長で破壊的なリカバリシーケンスを回避するための適切なバックアップ及びリプレイメカニズムを提供する。

10

#### 【0038】

いくつかのデバイスは、転送時のデータ保護をサポートする（例えば、エラー検出及び訂正（EDC）を伴うGDDR5読み出し及び書き込みデータ転送、書き込みCRCによって保護されたDDR4書き込みデータ転送等）。GDDR5デバイスは、CRCデータを転送するための単方向EDCバスを提供する。これにより、要求が読み出しか書き込みかにかかわらず、EDC値がデバイスからコントローラに常に伝えられる。GDDR5の読み出し応答データの転送中に、EDCバンドルが、パラメータ`tcrcr1`に基づいて、応答データと共に又は応答データの直後に返される。書き込みデータの転送中、EDCバンドルが、GDDR5デバイスが書き込みデータを受信した後に（受信した書き込みデータからEDC値が計算されるため）返される。読み出し時に、メモリコントローラ500は、受信した読み出しデータ応答からEDC値を計算し、EDC値を、その読み出しから受信したEDCデータと比較する。書き込み時に、メモリコントローラ500は、予想EDC値を計算し、書き込みデータサイクル後にGDDR5から返されたEDCパケットと後で比較するために、計算した予想EDC値をメモリコントローラ500に一時的に記憶する。予想書き込みデータEDC値は、メモリコントローラ500内のEDCキューロジック（EDCQ）に記憶される。

20

#### 【0039】

メモリコントローラ500は、レイテンシを短縮するために「早期応答」をサポートし、リプレイキュー530は、EDC応答が返される時間に対して「早期」に早期応答をメモリチャネルに返す。この「早期応答」サポートは、EDCの結果が「不良」に戻った場合に応答が「キャンセル」されることを要求する。メモリコントローラ500は、EDCの結果が不良に戻った場合に、「早期応答」パケット及び「応答キャンセル」をメモリチャネルに応答する。EDCが「良好」と返された場合、さらなる措置が不要となる。読み出し要求又は書き込み要求が失敗すると、リプレイキュー530は、サイクル要求の再試行を実行する。書き込み要求は、発行時にメモリチャネルに確認応答される。書き込みが失敗した場合、リプレイキュー530は、メモリチャネルとは無関係にコマンドを再試行し、書き込みデータの順序を維持する。

30

#### 【0040】

DDR4デバイスは、書き込みコマンドでのCRCチェックのみをサポートする。CRC情報は、最新の2ビット時間の間に書き込みデータと共に送信される。よって、GDDR5のEDCとは異なり、CRC情報がDDR4デバイスでチェックされ、デバイスは、エラー検出時にALERT#信号をアサートする。ALERT#信号はオープンドレインであり、DRAM MEMCLK又は任意の内部コントローラクロックと非同期であるとみなされ、これにより同期を必要とすることから、連続するバーストのシーケンスにおいてエラーを引き起こした特定の書き込みトランザクションを識別するメモリコントローラ500の能力が制限され、メモリコントローラ500は、以前に発行された様々な書き込みコマンドをリプレイして、失敗した書き込みのリプレイを確実にする。GDDR5及びDDR4の両方のメモリに対して、リプレイキュー530は、書き込み及び読み出しトランザクションをリプレイし、失敗したサイクルが正常に完了するまで、コマンドキュー520から発行されている新たなトランザクションを停止する。

40

50

## 【 0 0 4 1 】

リプレイキュー 5 3 0 の具体的な実施形態及び動作について説明する。

## 【 0 0 4 2 】

図 6 は、図 5 のメモリコントローラ 5 0 0 の有限状態マシン 6 0 0 の動作に関連する状態図である。有限状態マシン 6 0 0 は、N O R M A L 状態 6 1 0 と、「W A I T \_ A C K」と付された待機確認状態 6 2 0 と、「C M D \_ R E P」と付されたコマンドリプレイ状態 6 3 0 と、「E R R \_ R E C」と付されたエラーリカバリ状態 6 4 0 と、によって定義される。メモリコントローラ 5 0 0 は、D R A M エラーが検出されない限り、N O R M A L 状態 6 1 0 に留まる。この状態の間、アービタ 5 3 8 は、メモリシステムへのメモリアクセスコマンドの順序付けの制御を保持し、その通常の優先順位ルールに従ってコマンドキュー 5 2 0 からコマンドを選択することによって、通常のトラフィックを送信する。メモリコントローラ 5 0 0 は、リプレイキュー 5 3 0 から返された D R A M エラーを検出すると、N O R M A L 状態 6 1 0 を離れ、一時的に W A I T \_ A C K 状態 6 2 0 に移行する。次に、キュー 5 1 4 は、アービタ 5 3 8 によって選択されたコマンドの受け付けを停止し、選択されたりカバリフェーズにおいてリプレイキュー 5 3 0 からのコマンドの受信を開始する。リプレイキュー 5 3 0 は、メモリアクセスコマンドの発行を制御し、D R A M タイプ及びエラータイプに基づいてリカバリシーケンスを実行する。例えば、D D R 4 システムにおけるコマンド / アドレスエラーの場合、メモリコントローラ 5 0 0 は、E R R \_ R E C 状態 6 4 0 に移行し、コマンドが正しく提供されたことを保証するために、リプレイキュー 5 3 0 が最後のリカバリコマンドを送信した後の待機時間までこの状態に留まる。このとき、メモリコントローラ 5 0 0 は、C M D \_ R E P 状態 6 3 0 に移行する。C M D \_ R E P 状態 6 3 0 において、リプレイキュー 5 3 0 は、エラーが発生した可能性のある 1 つ以上のコマンドをリプレイする。C M D \_ R E P 状態 6 3 0 は、エラーが発生した可能性がある最後のコマンドをリプレイキュー 5 3 0 が送信したときに終了する。

## 【 0 0 4 3 】

例えば、典型的なシーケンスは以下のように進行する。

## 【 0 0 4 4 】

1 ) 起動時に、有限状態マシン 6 0 0 は、N O R M A L 状態 6 1 0 で開始する。

## 【 0 0 4 5 】

2 ) D D R 4 システムの場合、リプレイキュー 5 3 0 はエラー ( A L E R T \_ n = 0 ) をサンプリングし、G D D R 5 システムの場合、リプレイキュー 5 3 0 がキュー 5 1 4 から返された C R C エラーを受信すると、リプレイキュー 5 3 0 は、キュー 5 1 4 からの制御を要求し、有限状態マシン 6 0 0 は、W A I T \_ A C K 状態 6 2 0 に移行する。

## 【 0 0 4 6 】

3 ) アービタ 5 3 8 及びキュー 5 1 4 は、リカバリ要求を確認する前に任意の D R A M をパワーダウンモードから起動して、リプレイキュー 5 3 0 が制御している間、動的なパワーダウンを無効にする必要がある。

## 【 0 0 4 7 】

4 ) キュー 5 1 4 が要求を確認すると、状態マシン 6 0 0 は、E R R \_ R E C 状態 6 4 0 に移行してコマンド / アドレスエラーリカバリシーケンスを実行するか、C M D \_ R E P 状態 6 3 0 に直接移行する。

## 【 0 0 4 8 】

5 ) コマンド / アドレスエラーリカバリが完了すると、状態マシン 6 0 0 は C M D \_ R E P 状態 6 3 0 に移行する。

## 【 0 0 4 9 】

6 ) C M D \_ R E P 状態 6 3 0 において、エラートランザクションがリプレイされ、メモリシステムに再送される。さらなるエラーが発生した場合、状態マシン 6 0 0 は、C M D \_ R E P 状態 6 3 0 に留まるか、コマンド / アドレスエラーの場合には E R R \_ R E C 状態 6 4 0 に移行する。

## 【 0 0 5 0 】

10

20

30

40

50

7) リプレイが完了してクールダウン時間が経過すると、リプレイキュー530は、トランザクションの制御をアービタ538に解放し、有限状態マシン600は、NORMAL状態610に戻り、メモリコントローラ500は、再び通常のコマンドを提供する用意を整える。メモリコントローラ500がメモリコマンドをリプレイし、所定の回数だけエラーを受信した場合、システムエラーを示す。エラーが繰り返し発生するということは、メモリバス上の一時的な状態ではなく、実際のシステム障害を示しているため、オペレーティングシステムによって他の是正措置を取る必要がある。

#### 【0051】

図5のメモリコントローラは、ハードウェア及びソフトウェアの様々な組み合わせによって実装されてもよい。例えば、ハードウェア回路は、プライオリティエンコーダ、有限ステートマシン、プログラマブルロジックアレイ(PLA)等を含んでもよい。例えば、アービタ538は、保留中のコマンドの相対的なタイミング適格性を評価するために、記憶されたプログラム命令を実行するマイクロコントローラを用いて実装することができる。この場合、いくつかの命令は、マイクロコントローラによって実行されるために、非一時的なコンピュータメモリ又はコンピュータ可読記憶媒体に記憶されてもよい。様々な実施形態では、非一時的なコンピュータ可読記憶媒体は、磁気若しくは光ディスク記憶デバイス、例えばフラッシュメモリ等のソリッドステート記憶デバイス、又は、他の不揮発性メモリデバイスを含む。非一時的なコンピュータ可読記憶媒体に記憶されたコンピュータ可読命令は、ソースコード、アセンブリ言語コード、オブジェクトコード、又は、1つ以上のプロセッサによって解釈及び/若しくは実行可能な他の命令フォーマットであってもよい。

#### 【0052】

図5のメモリコントローラ又はその一部(例えば、リプレイキュー530)は、プログラムによって読出され、集積回路を製造するために直接的若しくは間接的に使用されるデータベース又は他のデータ構造の形態のコンピュータアクセス可能なデータ構造によって記述されてもよいし表現されてもよい。例えば、このデータ構造は、例えばVerilog又はVHDL等の高水準設計言語(HDL)におけるハードウェア機能の動作レベル記述であってもよいし、レジスタ転送レベル(RTL)記述であってもよい。記述は、ゲートのリストを含むネットリストを合成ライブラリから生成するために当該記述を合成し得る合成ツールによって読出されてもよい。ネットリストは、集積回路を含むハードウェアの機能を表すゲートのセットを含む。そして、ネットリストを配置及びルーティングして、マスクに適用される幾何学的形状を記述するデータセットを生成してもよい。マスクは、集積回路を製造するために様々な半導体製造工程で使用されてもよい。或いは、コンピュータアクセス可能な記憶媒体上のデータベースは、所望により、ネットリスト(合成ライブラリ有り若しくは無し)又はデータセットであってもよいし、グラフィックデータシステム(GDS)IIデータであってもよい。

#### 【0053】

特定の実施形態について説明してきたが、これらの実施形態に対する様々な変更が当業者には明らかであろう。例えば、メモリチャネルコントローラ510及び/又は電力コントローラ550の内部アーキテクチャは、様々な実施形態において変更することができる。メモリコントローラ500は、DDR×メモリ以外の他のタイプのメモリ(例えば、高帯域幅メモリ(HBM)、RAMバスDRAM(RDRAM)等)にインタフェースすることができる。図示した実施形態は、別々のDIMMに対応する各ランクのメモリを示したが、他の実施形態では、各DIMMは複数のランクをサポートすることができる。

#### 【0054】

したがって、添付の特許請求の範囲によって、開示された実施形態の範囲に含まれる、開示された実施形態の全ての変更を包含することが意図される。

10

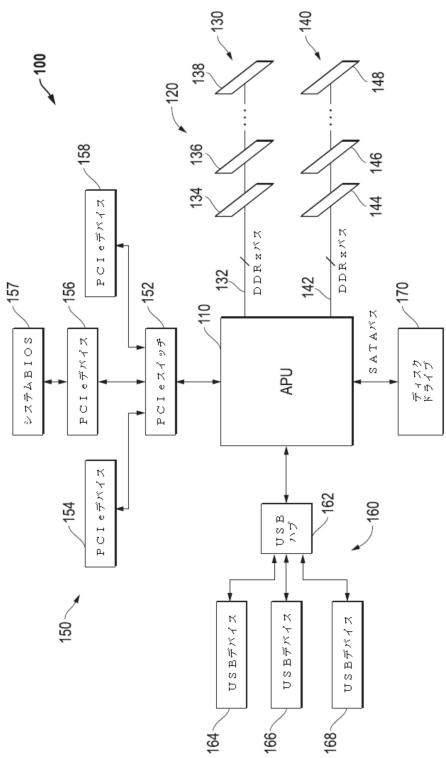
20

30

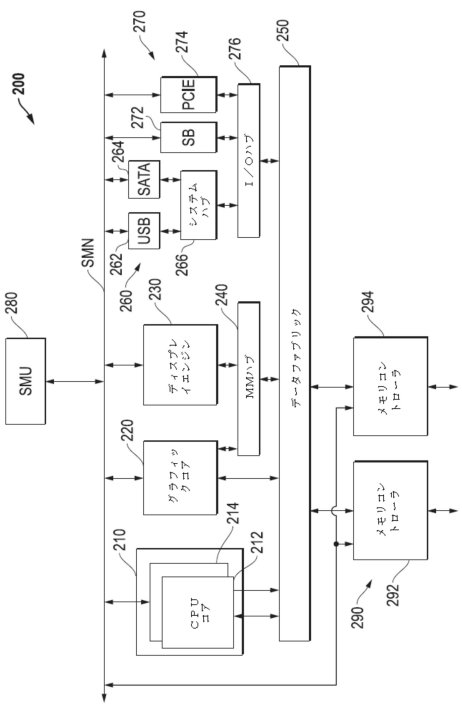
40

【図面】

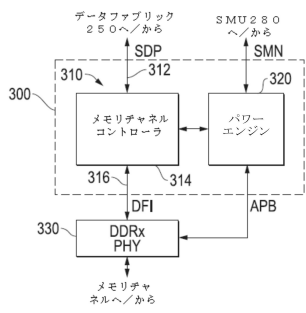
【図 1】



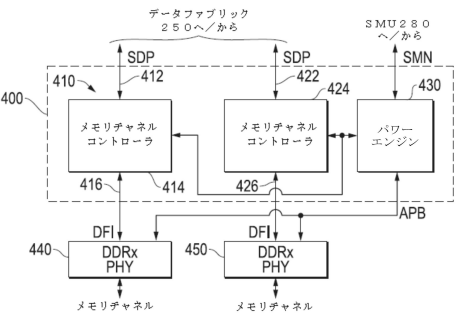
【図 2】



【図 3】



【図 4】



10

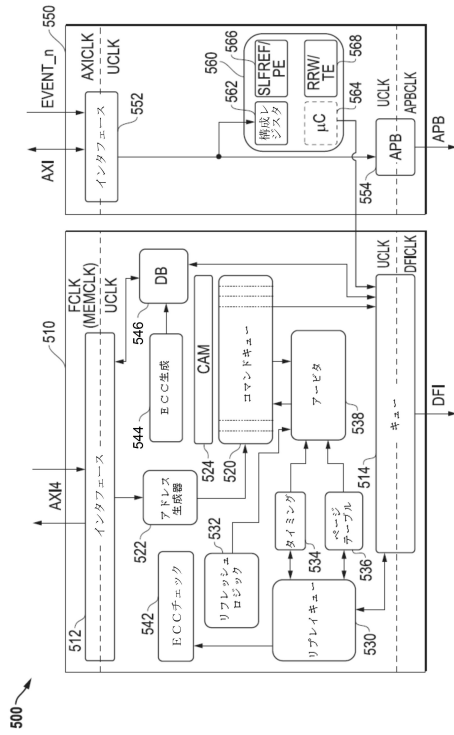
20

30

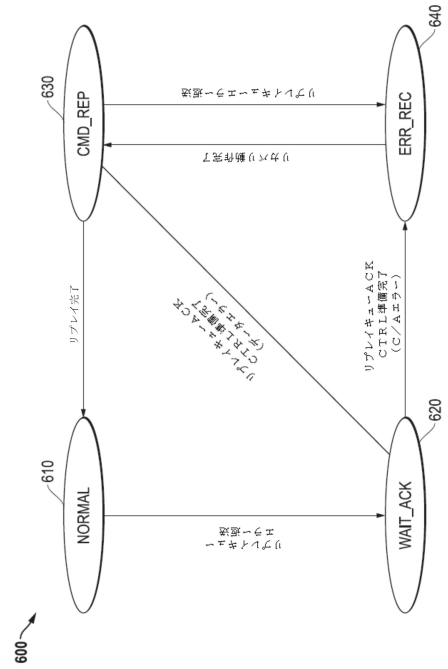
40

50

【 図 5 】



【 図 6 】





## フロントページの続き

## (33)優先権主張国・地域又は機関

米国(US)

アメリカ合衆国 7 8 7 3 5 テキサス州、オースティン、サウスウェスト パークウェイ 7 1 7 1

## (72)発明者 ルイファ ベン

アメリカ合衆国 9 5 0 5 4 カリフォルニア州、サンタ クララ、オーガスティン ドライブ 2 4 8 5

## (72)発明者 アンソニー アサロ

カナダ国 L 3 T 7 X 6 オンタリオ州、マーカム、1 コマース バリー ドライブ イースト

## (72)発明者 ケダーナス バラクリシュナン

アメリカ合衆国 7 8 7 3 5 テキサス州、オースティン、サウスウェスト パークウェイ 7 1 7 1

## (72)発明者 スコット ピー . マーフィ

アメリカ合衆国 9 5 0 5 4 カリフォルニア州、サンタ クララ、オーガスティン ドライブ 2 4 8 5

## (72)発明者 ユピン ヤオ

中華人民共和国 2 0 1 2 0 3 シャンハイ、プードン ディストリクト、チャンドン ロード 1 3  
8 7 番、チャンジャン ハイテック パーク、リバー フロント ハーバー ビルディング # 4 8

審査官 金田 孝之

## (56)参考文献 特表 2 0 0 9 - 5 2 7 8 1 9 ( J P , A )

米国特許第 0 8 3 6 5 0 1 5 ( U S , B 1 )

米国特許出願公開第 2 0 1 1 / 0 0 4 0 9 2 4 ( U S , A 1 )

特開 2 0 1 6 - 0 7 6 0 7 6 ( J P , A )

## (58)調査した分野 (Int.Cl. , D B 名)

G 0 6 F 1 1 / 1 4

G 0 6 F 1 1 / 0 7

G 0 6 F 1 2 / 0 0

G 1 1 C 2 9 / 5 2