

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5955323号
(P5955323)

(45) 発行日 平成28年7月20日 (2016. 7. 20)

(24) 登録日 平成28年6月24日 (2016. 6. 24)

(51) Int. Cl.

F I

G 0 6 F 12/00 (2006.01)

G 0 6 F 12/00 5 6 4 C

請求項の数 20 (全 23 頁)

(21) 出願番号	特願2013-529258 (P2013-529258)	(73) 特許権者	591016172
(86) (22) 出願日	平成23年9月13日 (2011. 9. 13)		アドバンスト・マイクロ・ディバイス・
(65) 公表番号	特表2013-543612 (P2013-543612A)		インコーポレイテッド
(43) 公表日	平成25年12月5日 (2013. 12. 5)		ADVANCED MICRO DEVI
(86) 国際出願番号	PCT/US2011/051345		CES INCORPORATED
(87) 国際公開番号	W02012/037086		アメリカ合衆国、94088-3453
(87) 国際公開日	平成24年3月22日 (2012. 3. 22)		カリフォルニア州、サニペール、ピー・
審査請求日	平成26年8月19日 (2014. 8. 19)		オウ・ボックス・3453、ワン・エイ・
(31) 優先権主張番号	12/910, 412		エム・ディ・プレイス、メイル・ストップ
(32) 優先日	平成22年10月22日 (2010. 10. 22)		・68 (番地なし)
(33) 優先権主張国	米国 (US)	(74) 代理人	100108833
(31) 優先権主張番号	61/382, 089		弁理士 早川 裕司
(32) 優先日	平成22年9月13日 (2010. 9. 13)	(74) 代理人	100111615
(33) 優先権主張国	米国 (US)		弁理士 佐野 良太

最終頁に続く

(54) 【発明の名称】 構成可能な電力状態をもつダイナミックRAMPHYインタフェース

(57) 【特許請求の範囲】

【請求項 1】

メモリデバイスのための物理メモリインタフェースを制御するための方法であって、
第1および第2の電力コンテキストを格納することと、

前記メモリデバイスからのデータ読取りおよび前記メモリデバイスへのデータ書込みの
ためにタイミング遅延を提供するように構成された複数の調整可能な遅延素子を提供する
ことであって、各調整可能な遅延素子が、前記第1および第2の電力コンテキストのうち
の選択された1つに応答する遅延時間を有する、複数の調整可能な遅延素子を提供するこ
とと、

電力コンテキスト変更要求を受信することと、

前記電力コンテキスト変更要求に基づいて、前記第1および第2の電力コンテキストの
うちの1つを選択することと
を含む方法。

【請求項 2】

前記第1および第2の電力コンテキストのうちの1つを選択する前に、前記メモリデバ
イスに関連したセルフリフレッシュモードを選択するように構成されたセルフリフレッシ
ュ出力を生成することをさらに含む、請求項1に記載の方法。

【請求項 3】

前記第1の電力コンテキストを格納するように構成された第1のセットのレジスタを提
供することと、

10

20

前記第 2 の電力コンテキストを格納するように構成された第 2 のセットのレジスタを提供することと、

前記電力コンテキスト変更要求に応答して、前記第 1 および前記第 2 のセットのレジスタのうちの 1 つを選択することと
をさらに含む、請求項 1 に記載の方法。

【請求項 4】

前記電力コンテキスト変更要求に応答して、前記物理インタフェースの少なくとも 1 つのドライバに対する選択可能なドライブ強度を調整することをさらに含む、請求項 1 に記載の方法。

【請求項 5】

前記電力コンテキスト変更要求に応答して、前記物理インタフェースの少なくとも 1 つの受信器に対する選択可能な終端インピーダンスを調整することをさらに含む、請求項 1 に記載の方法。

【請求項 6】

第 1 のメモリトレーニング段階を経て前記第 1 の電力コンテキストを生成することと、
第 2 のメモリトレーニング段階を経て前記第 2 の電力コンテキストを生成することとをさらに含む、請求項 1 に記載の方法。

【請求項 7】

電力制御インタフェース (A C P I) S 3 電力状態から復帰すると、前記第 1 および第 2 の電力コンテキストのうちの少なくとも 1 つを復元することをさらに含む、請求項 1 に記載の方法。

【請求項 8】

前記第 1 および第 2 の電力コンテキストをサウスブリッジ上のメモリ位置から取り出すことをさらに含む、請求項 1 に記載の方法。

【請求項 9】

前記第 1 および第 2 の電力コンテキストをノースブリッジ上のメモリ位置から取り出すことをさらに含む、請求項 1 に記載の方法。

【請求項 10】

メモリデバイスのための物理メモリインタフェースであって、
第 1 の電力コンテキストおよび第 2 の電力コンテキストを受信するように構成された複数のレジスタと、

前記第 1 および第 2 の電力コンテキストのうちの 1 つを選択するように構成された選択回路と、

前記メモリデバイスからのデータ読取りおよび前記メモリデバイスへのデータ書込みのためにタイミング遅延を提供するように構成された複数の調整可能な遅延素子であって、各調整可能な遅延素子が、前記第 1 および第 2 の電力コンテキストのうちの前記選択された 1 つに応答する遅延時間を有する、複数の調整可能な遅延素子とを含む、物理メモリインタフェース。

【請求項 11】

前記選択回路が、前記第 1 および第 2 の電力コンテキストのうちの前記選択された 1 つをメモリ位置から取り出す、請求項 10 に記載の物理メモリインタフェース。

【請求項 12】

前記第 1 の電力コンテキストを格納するように構成された第 1 のセットのレジスタと、
前記第 2 の電力コンテキストを格納するように構成された第 2 のセットのレジスタとをさらに含む、

前記選択回路が前記第 1 と第 2 のセットのレジスタとの間で選択するように構成されている、請求項 10 に記載の物理メモリインタフェース。

【請求項 13】

各々が、前記第 1 および第 2 の電力コンテキストのうちの前記選択された 1 つに応答して選択可能なドライブ強度を有する、複数のドライバをさらに含む、請求項 10 に記載の

10

20

30

40

50

物理メモリインタフェース。

【請求項 1 4】

各々が前記第 1 および第 2 の電力コンテキストのうちの前記選択された 1 つに応答して選択可能な終端インピーダンスを有する複数の受信器をさらに含む、請求項 1 0 に記載の物理メモリインタフェース。

【請求項 1 5】

前記第 1 の電力コンテキストが第 1 のメモリトレーニング段階を経て決定され、かつ、前記第 2 の電力コンテキストが第 2 のメモリトレーニング段階を経て決定される、請求項 1 0 に記載の物理メモリインタフェース。

【請求項 1 6】

物理メモリデバイスの複数のチャネルのためのインタフェースをさらに含む、請求項 1 0 に記載の物理メモリインタフェース。

【請求項 1 7】

前記物理メモリインタフェースが、中央処理装置 (C P U) のダイ上に配置されている、請求項 1 0 に記載の物理メモリインタフェース。

【請求項 1 8】

電力制御インタフェース (A C P I) S 3 電力状態から復帰すると、前記第 1 および第 2 の電力コンテキストのうちの少なくとも 1 つを取り出すように構成されたメモリインタフェースをさらに含む、請求項 1 0 に記載の物理メモリインタフェース。

【請求項 1 9】

前記第 1 および第 2 の電力コンテキストをメモリ位置からロードするように構成されたメモリインタフェースをさらに含む、請求項 1 0 に記載の物理メモリインタフェース。

【請求項 2 0】

前記物理メモリインタフェースに結合されたメモリデバイスをさらに含み、前記メモリデバイスが、前記第 1 および第 2 の電力コンテキストのうちの前記選択された 1 つに関連する前記タイミング遅延を用いて、データの読取りおよび書込みのために構成されている、請求項 1 0 に記載の物理メモリインタフェース。

【発明の詳細な説明】

【技術分野】

【 0 0 0 1】

本発明は、ダイナミックランダムアクセスメモリ (D R A M) デバイスと直接インタフェースをとる物理層を含むメモリサブシステムに関する。

【 0 0 0 2】

〔関連出願の相互参照〕

本願は、2010年9月13日に出願された、米国仮出願第61/382,089号、および2010年10月22日に出願された、米国非仮出願第12/910,412号の利益を主張し、それらは、本明細書中に完全に明記されているかのように、参照により組み込まれる。

【背景技術】

【 0 0 0 3】

典型的なメモリシステムは、メモリコントローラとメモリデバイスとの間でデータを伝送するために、非同期クロック方式または同期クロック方式のいずれかを使用する。同期クロックは、メモリデバイスが、制御入力に応答する前にクロック信号を待ち、従って、コンピュータのシステムバスと同期されることを意味する。同期ダイナミックランダムアクセスメモリ (S D R A M) は、通常非同期メモリデバイスよりも高いクロック速度をサポートするので、幅広く使用される。

【 0 0 0 4】

ダブルデータレート (D D R) S D R A M は、クロック信号の立ち上がりエッジおよび立ち下がりエッジの両方でデータを伝送する。かかるメモリデバイスは、より低いクロック周波数を使用するが、電気データおよびクロック信号のタイミングの厳格な制御を必要

10

20

30

40

50

とする。かかるデバイスの第1のバージョン(DDR1)は、同じクロック周波数で動作するシングルデータレート(SDR)SDRAMの帯域幅のほぼ2倍を達成した。DDR2およびDDR3 SDRAMデバイスは、DDR1デバイスに関する後続の改善である。どのタイプのDDRメモリが使用されているか(DDR1/DDR2/DDR3)に関わらず、物理インタフェース(PHY)が、メモリコントローラとDDR SDRAMデバイスとの間に直接結合される。PHYインタフェースは、一般に、DDR SDRAMデータストロープのタイミング要件を処理するための回路機構を含む。典型的なPHYインタフェースの実装は、メモリ性能レベルまたは要求される電力を迅速に調整するための機構を提供しない。

【発明の概要】

10

【課題を解決するための手段】

【0005】

物理メモリインタフェース(PHY)が提供される。PHYは、メモリコントローラと物理メモリデバイスとの間のインタフェースをとる。PHYインタフェースは、第1の電力コンテキストおよび第2の電力コンテキストを受信するように構成されたコマンドおよび状態レジスタ(CSR)を含む。選択回路が提供される。選択回路は、第1と第2の電力コンテキストの間で切り替えるように構成される。PHYインタフェースは、複数の調整可能な遅延素子を含み、各々が、選択された電力コンテキストに応答する遅延時間を有する。電力コンテキスト間での切替えは、1つまたは複数の調整可能な遅延素子の調整という結果になる。

20

【0006】

別の実施形態では、PHYインタフェースは、第1の電力コンテキストを格納するように構成された第1のセットのCSRおよび第2の電力コンテキストを格納するように構成された第2のセットのCSRを含む。PHYインタフェースは、各々が、選択された電力コンテキストに応答する選択可能なドライブ強度を有する複数のドライバも含む。PHYインタフェースは、各々が、選択された電力コンテキストに応答する選択可能な終端インピーダンスを有する複数の受信器も含む。電力コンテキスト間での切替えは、1つまたは複数のドライバ/受信器のドライブ強度および/または終端インピーダンスの調整という結果になり得る。

【0007】

30

第1および第2の電力コンテキストは、BIOSトレーニング手順を経て決定され得る。かかる手順は、複数の段階をもち得る。例えば、第1の電力コンテキストは、第1のメモリトレーニング段階を経て決定され得、第2の電力コンテキストは、第2のメモリトレーニング段階を経て決定され得る。

【0008】

PHYインタフェースは、CSRへの読取り/書込みアクセスを可能にするように構成されている構成バスを含み得る。PHYインタフェースは、物理メモリデバイスの複数のチャンネルをサポートするようにも構成され得る。PHYインタフェースは、中央処理装置(CPU)のダイ上を含め、様々な場所に配置され得る。

【図面の簡単な説明】

40

【0009】

【図1】典型的なコンピュータシステムの一部を示す。

【図2】PHYの高レベルアーキテクチャを示す。

【図3】PHYインタフェース構成バス上のアドレス、コマンド(例えば、読取り、書込み)、およびデータを示すタイミング図である。

【図4】CSRアドレス指定の内訳を示す。

【発明を実施するための形態】

【0010】

図1は、中央処理装置(CPU)12、コア理論チップセット14、16およびメインメモリ28へのメモリインタフェース18を含む、コンピュータシステム10の一部を示

50

す。コア理論チップセットは、ノースブリッジ14（または統合メモリコントローラ）とサウスブリッジ（または入出力コントローラハブ）16との間で分割され得る。メモリコントローラ18は、しばしばノースブリッジ14に配置される。かかる回路機構は、単一チップまたはCPU内など、様々な場所に物理的に配置され得ることを理解すべきである。メモリコントローラ18は、通常、メインメモリ30との間を行き来するデータの論理的な流れを管理する。ダイナミックランダムアクセスメモリは、様々なメモリ記憶セル内のチャージを保持するために定期的なリフレッシュ信号も必要とし得る。メモリコントローラ18は、一般に、これらのリフレッシュ操作を管理する。CPUアーキテクチャおよび/またはオペレーティングシステムに応じて、メインメモリ30は、データを32ビットまたは64ビットデータ単位で伝送し得る。他のサイズのデータ単位がサポートされ得ることを理解すべきである。いくつかのメモリシステムは、ブロック20、24、および30で示される、複数のチャネル（例えば、2つ以上の独立したメモリコントローラ）を含む。明確にするために、本明細書に含まれる例は、単一のメモリチャネルのみを示し得る。本開示の範囲から逸脱することなく、複数のチャネルが使用され得ることを理解すべきである。

【0011】

Phyインタフェース22は、メモリコントローラ18と物理メモリデバイスとの間に存在する。Phyインタフェースは、一般に、中央処理装置内に配置されるが、他の場所に配置され得る。明確にするために、Phyインタフェース22は、図1では別個のブロックとして示す。Phyインタフェースは、通常、メインメモリデータストロープのタイミング要件を処理するための回路機構を含む。本開示では、ダブルデータレート（DDR）という用語は、DDR1/DDR2/DDR3および/またはかかるメモリデバイスの次の世代を包含する。DDRメモリデバイスは、一般に、電子デバイス技術合同協議会（JEDEC: Joint Electron Devices Engineering Councils）規格に準拠する。これらの規格は、DDRメモリデータがどのようにアクセスされ（読み取られ）、格納される（書き込まれる）のかを定義する。Phyインタフェースは、他のメモリタイプおよび/または他のメモリ規格と共に動作するように構築され得ることを理解すべきである。典型的なDDR SDRAMメモリデバイスへのインタフェースは、主として2つの信号クラス、DQ（データ）32およびDQS（データストロープ）34を通して達成される。他の信号は、メモリクロック（MEMCLK）38、ならびにアドレスおよびコマンド信号（通常、ADDR/CMD 36として示す）を含む。典型的なメモリデバイスは、図に示されていない、追加の信号を使用し得ることを理解すべきである。かかる信号は、明確さのみのために省略されている。例えば、電力および接地信号は示されていない。かかる信号は、典型的な実施態様に含まれるであろうことを理解すべきである。

【0012】

読取り操作中、DDR SDRAMは、DQおよびDQSを、一般に「エッジアライン（edge aligned）」と呼ばれる方法で、同時に発行する。メモリコントローラが、DDR SDRAMから送信されているデータを正しく取得するため、Phyインタフェース22は、遅延ロックスループ（DLL）などの遅延回路を利用して、有効なデータウィンドウまたは「データアイ（data eye）」の間に、DQ信号を正しくラッチするためにDQS信号が使用され得るように、DQS信号を遅延させる。同様に、Phyインタフェース22は、データのDDR DRAMへの書込みをサポートするためにも遅延回路を利用し得る。データ読取りのため、DQS 34を遅延させる必要がある。データ書込みのため、DQSおよびDQ 34、32を遅延させる必要がある。Phyは、DQS 34を、エッジアラインよりむしろ、DQ 32データアイの中央に合わせる。DQS 34は、書込み平準化（write leveling）のために遅延されて、データアイの中央要件を満足する。他の遅延も使用され得る（例えば、データバスの3状態制御の読取り/書込みのため）。Phyインタフェース22は、以下でさらに詳述する、遅延タイミング、ドライブ強度、および様々な他のパラメータを制御するために利用さ

10

20

30

40

50

れる複数のコマンドおよび状態レジスタ (C S R) 4 2 を含む。かかる回路機構は、チャネルベース毎でも重複され得ることを理解すべきである。

【 0 0 1 3 】

P h y インタフェースは、また、送信器のドライブ強度および受信器の終端インピーダンスの調整または選択も行い得る。固定のタイミング遅延、送信器ドライブ強度および受信器終端インピーダンスを使用するよりも、これらのパラメータは、コンピュータシステムが作動されるたびに調整され得る。これは、通常、トレーニングプログラムの支援を受けて達成される。トレーニングプログラムは、通常、基本入出力システム (B I O S) メモリデバイス 2 6 に格納されるが、装置ハードウェア内にも実装され得る。トレーニングプログラムは、パワーオンセルフテスト (P O S T) 中にアルゴリズムを実行し、それが、多数のメモリインタフェース信号に関連した、適切なタイミング遅延、ドライブ強度および終端インピーダンスを決定する。これらのパラメータは、P h y への、および P h y からの様々な信号経路の全体的なタイミングを定義する複数のレジスタで P h y インタフェース内に保存される。別の方法では、これらのパラメータは、別の場所 (例えば、ノースブリッジ 1 4 またはサウスブリッジ 1 6 内) に格納され得る。

10

【 0 0 1 4 】

図 2 は、P h y インタフェース 2 2 の高レベルアーキテクチャを示す。P h y インタフェースは、通常、クロックソース (例えば、P L L 7 0) およびコマンドおよび状態レジスタ (C S R) 4 2 を含む制御インタフェースを含む。P h y インタフェース 2 0 は、D Q 3 2、D Q S 3 4、アドレス / コマンド線 3 6 およびクロック入力 3 8 を含め、デュアルインラインメモリモジュール (D I M M) などの典型的なメモリデバイス上の大多数のピンに対して物理接続を提供する。

20

【 0 0 1 5 】

典型的なメモリデバイスは、クロック可能 (セルフリフレッシュ) 入力 4 0 も備え得る。クロック可能入力 4 0 は、メモリデバイスをセルフリフレッシュモードにするために使用される。このモードでは、メモリデバイスは、必要に応じて内部のリフレッシュサイクルを生成するために、オンチップタイマーを使用する。外部クロックもこの間、停止され得る。この入力は、メインメモリデータを失うことなくメモリコントローラを無効にできるので、通常、電源ダウンモードに関連して使用される。

【 0 0 1 6 】

図 2 に示すように、D Q 3 2 (データ) および D Q S 3 4 (ストロープ) 線は双方向である。各 D I M M は、複数の D Q 線 (例えば、6 4 データビット) および複数の D Q S 線を有することを理解すべきである。これらの線の各々は、P h y インタフェース 2 2 内に、関連するドライバ 5 2、5 6 および受信器 5 4、5 8 を有する。図 2 では、明確にするために、単一のドライバ / 受信器の対を示す。各ドライバは、選択可能または調整可能なドライブ出力を有する。同様に、各受信器は、選択可能または調整可能な終端インピーダンス (例えば、オンダイ終端 (o n d i e t e r m i n a t i o n)) を有する。D Q および D Q S 線 3 2、3 4 は、遅延ロックループ (D L L) 7 2、7 4、7 6 などの遅延素子にも関連する。遅延素子 (例えば、7 0 ~ 7 6)、ドライブ強度または終端インピーダンス値に対する調整は、以下でさらに詳細に説明するように、適切な値を、関連する C S R 4 2 の適切なフィールド内にプログラミングすることによって達成される。C S R 間の論理結合および遅延素子、ドライブ強度または終端インピーダンスにおける調整は、一般に、点線で示される。

30

40

【 0 0 1 7 】

この例では、ドライバ 5 2 および 5 6 は、D L L 7 2 および 7 4 とそれぞれ関連する。受信器 5 8 は、D L L 7 6 と関連する。前述のように、D L L は、読取りおよび書込み操作のために、適切なタイミング遅延を提供するように調整される。P h y インタフェースは、平準化の有無にかかわらず、読取りおよび書込み操作を実行するようにも構成され得る。平準化を行う D D R 3 D I M M S へのメモリ書き込み操作中に、P h y インタフェースは、各 D R A M チップにおいて、D Q S がメモリクロック 5 8 と合体する (c o

50

a l e s c e) のが見られるように、D I M M に進む各 D Q S の開始を遅延させる。平準化を行う読取り操作中、P h y インタフェースは、フライバイポロジによって取り込まれた遅延の補償も行い得る。

【 0 0 1 8 】

より高いデータレートでの操作の信号品位の問題に起因して、P h y インタフェースは、D L L 設定をバーストごと（またはトランザクション）のベースで動的に変更し得る。P h y インタフェースは、システム内の各 D I M M に対して D Q および D Q S に対する遅延設定の最適組 (t u p p l e) を格納し得る。アクセスされている D I M M に応じて、P h y インタフェースは、適切な D L L 設定を取り出し、それらを適用する。

【 0 0 1 9 】

P h y インタフェースは、所望の性能レベルに基づき、その要求される電力を調整し得る。電力コンテキストをいつ変更すべきかの決定は、様々な情報源に由来し得ることを理解すべきである。例えば、オペレーティングシステムは、コンテキスト変更が要求されることを（例えば、アイドル時間の設定期間後、ユーザーコマンド、時間スケジュールまたは同様のものによって）判断し得る。別の方法では、コンテキスト変更がいつ要求されるかを判断するためにハードウェアが使用され得る。コンテキスト変更は、異なる電力状態に関連する、異なるセットの P h y インタフェースパラメータ間で切り替えることにより達成される。例えば、高電力状態（例えば、より高いメモリ速度）および低電力状態（例えば、より低いメモリ速度）。各電力状態は、P h y インタフェースパラメータまたはコンテキストに関連するセット（つまり、各信号線に対する遅延素子設定、ドライブ強度および終端インピーダンス）を有する。以降で詳述するように、電力状態間の切替えは、いくつかの方法で達成され得る。本明細書で開示するように、複数の電力状態間の切替えは、任意のメモリタイプに適用でき、以下の例で使用されるように、D D R メモリとの使用に限定されないことを理解すべきである。

【 0 0 2 0 】

この例では、メモリコントローラ 1 8 は、3 2 ビットの、時間インターリーブされた、一方向の構成バス 8 0 を介して、C S R 4 2 にアクセスし得る。アドレスおよびコマンド（例えば、読取り、書込み、何もしない）は、図 3 に示すように、第 1 のパイプステージ内に送信され、その後、第 2 のパイプステージ内のデータが続く。C S R は、他の通信リンクを介してアクセスできることを理解すべきである。

【 0 0 2 1 】

この例では、C S R アドレス空間は 1 6 ビット幅で、6 5 , 5 3 6 の一意の 1 6 ビットレジスタの空間を可能にする。かかる大空間を可能にする代わりに、アドレスは、次の機能を可能にするためにマッピングされる：すなわち、チップレット識別、イントラチップレットブロードキャスト (i n t r a - c h i p l e t b r o a d c a s t) 、補償ブロードキャスト (c o m p e n s a t i o n b r o a d c a s t) 、チップレットインスタンス識別 (D 3 D B Y T E 、 D 3 C L K 、 および D 3 C M P は、2 回以上配置されるチップレットである) 。図 4 は、C S R アドレス指定に対する内訳を示す。

【 0 0 2 2 】

C S R の一部のみに所与の電力状態に関連する値を含む。電力状態間での低遅延の切替えを容易にするため、各電力状態に対して電力コンテキスト依存の C S R のセットが提供される。図 2 に戻ると、C S R の第 1 のセットが第 1 の電力コンテキストと関連する P h y P S [0] 4 4 。C S R の別のセットが第 2 の電力コンテキストと関連する P h y P S [1] 4 6 。P h y P S [n] 4 8 によって示されるように、3 つ以上の電力コンテキストをサポートするために、C S R の追加のセットが提供され得ることを理解すべきである。C S R の異なるセット間で選択するために、マルチプレクサまたは選択回路 5 0 が提供される。P h y インタフェースは、電力コンテキスト依存でない C S R も含む。かかる C S R は、全ての電力状態で適用可能である。

【 0 0 2 3 】

各 P h y P S 内の各 C S R に対するプログラム可能フィールドの要約を以下の表 1 に示

10

20

30

40

50

す。

【表 1】

番号	説明
1	DDR PLLレート；
2	DLL（遅延ロックループ）周波数固有のプログラミング；
3	DQS（ストローブビット）ドライブ強度（DS）；
4	DQ（データビット）DS
5	MemClk DS
6	アドレス&コマンド（Addr，バンク，RAS，CAS，WE，AddrParity）DS
7	チップ選択（Cs）&オンダイ終了（Odt）DS
8	クロック可能（Cke）DS
9	ProcOdt強度
10	ProcOdtOff
11	バイトごとのDIMMごとのチャネルごとのTxDqsDll制御
12	バイトごとのDIMMごとのチャネルごとのTxDqDll制御
13	バイトごとのDIMMごとのチャネルごとのRxEnDll制御
14	ニブルごとDIMMごとチャネルごとのRxDqsLDll制御（x4およびx8のDIMMタイプで使用される）
15	ニブルごとDIMMごとチャネルごとのRxDqsUDll制御（x4のDIMMタイプで使用される）
16	チャネルごとのAddrCmdDll制御
17	チャネルごとのCsOdtDll制御
18	チャネルごとのCkeDll制御
19	DIMM対ごとのチャネルごとのMemClkDll制御

【0024】

この例では、Phyインタフェースの関連するフィールドのプログラミングは、構成バス80を介して、コマンドを発行するか、またはCSR内の個々のフィールドをプログラミングすることによって達成される。例えば、1つのPhyPSから別への変更は、DDRPhyインタフェース間接レジスタ空間に発行された単一のコマンドで達成され得る。PhyPSコンテキスト依存CSRのプログラミングは、適切なPhyPSコンテキストを設定し、次いで、正常な間接CSR書込みまたは読取りを行うことによって達成され得る。あるいは、直接CSR書込みまたは読取りは、PhyPSコンテキストを考慮することなく、任意のCSRに達し得る。

【0025】

Phyインタフェースは、Master__0x08[12]-PhyPS、Master__0x08[8]-PstateToAccessおよびMaster__0x18[8]-PhyPSMasterChannelを含む、一連のコマンドで制御され得る。用語体系[12]、[8]および同様のものは、コマンド内のビット位置を指す。Mas

er_0x08[12] コマンドは、現在のPhyインタフェースP状態(0または1)に対応する。このコマンドは、電力コンテキスト(例えば、CSRのどのセットが現在アクティブであるか)を制御する。Master_0x08[8] コマンドは、CSRアクセス中に、どのP状態の読取りまたは書込みを行うかを選択する。BIOSは、実際のP状態変更を行う必要なく、どのP状態を書き込むかを制御するために、この方法を使用し得る。3つ以上の電力コンテキストをサポートするために、追加のビットが使用され得ることを理解すべきである。Master_0x18[8] コマンドはマスターチャネルを選択する。この実施形態では、このビットによって指定されたチャネル(マスターチャネル)のみが、0x0B[PhyPS変更] コマンドの発行を許可される。他のチャネルから発行されたどの0x0B[PhyPS変更] コマンドも無視されるであろう。3つ以上のチャネルをサポートするために、追加のビットが追加され得ることを理解すべきである。この実施形態では、以下の0x0Bコマンドが定義される：

【表2】

番号	説明
1	PhyPSを変更する
2	PhySR(セルフリフレッシュ)に入って、出る
3	DDRレート変更(すなわち、DDRPLL周波数を変更する)
4	他のPhyインタフェースsubモードを(subコマンドで)更新する
5	~PwrOkの既定の設定

【0026】

完全な0x0Bデータパケットを以下の表3に示す。

【表 3】

ビット位置	説明
31	PhySRRequest (セルフリフレッシュ)
30	PhyPSRequest
29～27	予約済み
26	PhyPS=<所望のPhyインタフェース電力状態> PhyPS 0または1の低電力または高電力モードとの 関連(それぞれ低DDRレートまたは高DDRレート)は、 Phyの観点から見れば任意である。この例では、Phy PS1は低電力状態として、またPhyPS0は高電力状 態として定義される。
25～24	PhySR PLL制御
23	PhySR
22～18	予約済み
17～16	DdrRate
15～4	予約済み
3	DdrRateRequest
2～0	DdrRate

【0027】

この例では、電力コンテキストが単一ビット(すなわち、PhyPS[0]およびPhyPS[1] ビット位置26)で選択される。3つ以上のコンテキストをサポートするために、追加のビットが追加され得ることを理解すべきである。PhyPSRequestビット(ビット位置30)は、コマンドがコンテキスト変更を含むことを示すために使用される。電力コンテキストは、通常、以下のように変更され得る。コンテキスト変更要求を受信すると、各アクティブなチャンネルがセルフリフレッシュ(SR)モード(0x0B[PhySR=1])にされる。この「セーフモード」では、メモリデバイスは、必要に応じて内部のリフレッシュサイクルを生成するために、オンチップタイマーを使用する。選択したSRモードに応じて、外部クロックもこの間、停止され得る。SRモードに入るのは、異なる時に起こり得るが、全てのチャンネルは、コンテキスト変更の前にSRモードでなければならない。電力コンテキスト変更が開始される(0x0B[PhyPS=X]、[PhyPSRequest=1])。アクティブなチャンネルの各々は、SRモードから正常モード(0x0B[PhySR=0])に切り替えられる。

【0028】

コンテキスト変更コマンドを受信すると、Phyインタフェースは、PhyPSコンテキストを変更し、DDR PLL乗算器およびドライバが更新され、そして、PLLが再ロックされる。これが全て完了すると、CfgDoneが設定される。その後、SRモードから出ると、DLLが再ロックする。PhyPS変更が生じている間、メモリコントローラは、Phyインタフェースへの全ての入力に対する制御(具体的には、CKE、MemReset、ReadPending、WritePending、他の3状態制御の全て)を維持する。

【0029】

意図は、PhyPSを可能な限り迅速に変更することにあるので、PhyPS変更はP

10

20

30

40

50

LLを電源の入ったままにすることが一般に予期される。従って、PhyPSの変更は、直ちにPLL再ロック（および再ロックの待機）をトリガーし、SRから出た後にDLL再ロックが生じる。しかし、PLL（VCOまたは調整器のいずれか）の電源を落としているSRモード中にあり、かつ、このSRモードにある間にPhyPSを変更することは可能である。この場合、Phyインタフェースはまだ、PhyPS変更の直後にPLL再ロック時間を待機するが、後にSRから出ると、Phyインタフェースは、PLL再ロック（PLLが電源を入れ戻す過程にあるので）およびDLL再ロックの両方を待機するであろう。

【0030】

Phyインタフェースセルフリフレッシュ（PhySR）に入って出る0x0Bコマンドは、チャンネル0または1のいずれかで発行され得；1つのDRAMチャンネルをSRモードにするが、他はそうではないことを可能にする。Phy内のPClkグローバルグリッドの電源を完全に落とすために、チャンネル0および1の両方がPhySRにある必要がある。PhySRを変更する0x0Bを、以下の表4に示す：

【表4】

ビット位置	説明
31	PhySRRequest（セルフリフレッシュ）
30	PhyPSRequest=0
29～27	該当なし
26	PhyPS=<所望のPhyインタフェース電力状態>=X
25～24	PhySR PLL制御 {調整器PD, VCO PD}
23	PhySRに入るにはPhySR=1、出るには0
22～0	該当なし

【0031】

0x0B[23=PhySR]とともに0x0B[31=PhySRRequest]を設定すると、Phyインタフェースがセルフリフレッシュに入るか、または出るようにする。0x0B[25:24]は、SRにある間、PLLの動作を制御する。SRモードに入るのは非常に迅速であり、0x0Bコマンドの受信からCfgDoneがアサートされる時間まで～300nsかかる。メモリコントローラが可能な限り迅速に前進できるようにするため、メモリコントローラは、CfgDoneが1から0へ遷移するのをモニタし得る。これは、Phyインタフェースがメモリコントローラからの入力を遮断して、すべてのCKEを低下させ、MemResetを適切にドライブし、残りの全てのDDRバスピンを3状態にすることを示す。この時点で、メモリコントローラが狂うのは安全である。CfgDoneがアサートされるまでVddrレールをドロップして、必要なCSRトランザクションの全てが完了していることを示すことは安全ではない。

【0032】

PhySRから出ることは、はるかに潜在的であり、Phyインタフェースクロックグリッド（PClk）がオンに戻され、DLLが再ロックされることを必要とする。PLLは、SRの間、オンのままにされるか、またはオフにされ得る。この実施形態では、PhySR変更の実行中、0x0B[30=PhyPSRequest]は0でなければならない。0x0B[31=PhySRRequest]または0x0B[30=PhyPSRequest]のいずれかを設定すると、0x0B[22:0]を無効にし、それは、DDRレートフィールドを更新するために0x0B[3=DdrRateRequest]を設定することはできないことを意味する。PhyインタフェースPLL乗算器およびドライバを含むCSRをロードするために、DDRレートを第1に設定する。0x0B[P

h y P S 変更]を送信すると、D D R P L L 乗算器およびドライバをロードする。他の実施形態では、複数の高レベル 0 x 0 B コマンドを同時に実行することが可能であり得る。0 x 0 B [3 1 = P h y S R R e q u e s t] = 1 かつ 0 x 0 B [2 3 = P h y S R] が結果的に P h y S R 状態に対して変更を行わない場合、C f g D o n e をアサートする前に、D L L ロック時間がまだ従われる。

【 0 0 3 3 】

この実施形態では、D D R バス上の D R A M データレートに対する 0 x 0 B コマンドは、チャンネル 0 においてのみ B I O S によって設定される。チャンネル 1 における D R A M データレートを設定するために 0 x 0 B コマンドを送信しても、効果がない。他の実施形態は、各チャンネル上で独自の D D R データレートをサポートし得ることを理解すべきである。0 x 0 B D D R レートコマンドを以下の表 5 に示す：

【表 5】

ビット位置	説明
3 1	P h y S R R e q u e s t = 0
3 0	P h y P S R e q u e s t = 0
2 9 ~ 2 7	予約済み
2 6	P h y P S = < 所望の P h y インタフェース電力状態 > = X
2 5 ~ 2 4	P h y S R P l l 制御 = X
2 3	P h y S R = X
2 2 ~ 1 8	予約済み
1 7 ~ 1 6	D d r R a t e [4 : 3]
1 5 ~ 4	予約済み
3	D d r R a t e R e q u e s t = 1
2 ~ 0	D d r R a t e [2 : 0] = 所望のレートに設定

【 0 0 3 4 】

この 0 x 0 B [D d r R a t e] コマンドを介して実際の D D R レートおよび P L L 周波数を変更することは、レガシー B I O S のために含まれている。P h y インタフェース P 状態を理解する B I O S は、代わりに、直接 C S R M a s t e r _ 0 x 0 0 [D d r R a t e] および M a s t e r _ 0 x 4 0 [D d r R a t e] を通じて P h y インタフェース P 状態の両方に対して D D R レートをプログラムすべきである。これらの（直接 C S R を通じた）レート変更は、以降の 0 x 0 B [P h y P S 変更] の後にのみ影響を及ぼすであろう。

【 0 0 3 5 】

この実施形態では、D D R レートに対する更新を実行する場合、0 x 0 B [3 1 = P h y S R R e q u e s t] および 0 x 0 B [3 0 = P h y P S R e q u e s t] は両方とも 0 でなければならない。実施形態は、複数の高レベル 0 x 0 B コマンドを同時にサポートし得ることを理解すべきである。

【 0 0 3 6 】

コールドリセット、ウォームリセット後の P h y P S 状態、または高度構造及び電力制御インタフェース (A C P I) 電力状態 S 3 (一般に、R A M に対するスタンバイ (S t a n b y) 、スリープ (S l e e p) 、またはサスペンド (S u s p e n d) と呼ばれる) を以下の表 6 に示す。全ての状態は、ウォームリセットを通して持続する。それ故、以下のウォームリセットの項目では、値は、ウォームリセットの前のままである。

10

20

30

40

【表 6】

チャンネル実装	イベント	両方のPhyPSがサポートされる場合の規定値	1つだけのPhyPSがサポートされる場合の規定値
チャンネル0のみ	コールドリセット	PhyPS=0	PhyPS=0
		PStateToAccess=0	PStateToAccess=0
		PhyPSMasterChannel=0	PhyPSMasterChannel=0
		PhySR=0 (すべてのチャンネル)	PhySR=0 (すべてのチャンネル)
	ウォームリセット	PhyPS=0または1	PhyPS=0
		PStateToAccess=0または1	PStateToAccess=0
		PhyPSMasterChannel=0	PhyPSMasterChannel=0
		PhySR=0または1 (チャンネル0)	PhySR=0または1 (チャンネル0)
		PhySR=1 (チャンネル1)	PhySR=1 (チャンネル1)
	S3	コールドリセットと同じ	コールドリセットと同じ
チャンネル1のみ	コールドリセット	PhyPS=0	PhyPS=0
		PStateToAccess=0	PStateToAccess=0
		PhyPSMasterChannel=0	PhyPSMasterChannel=0
		PhySR=0 (すべてのチャンネル)	PhySR=0 (すべてのチャンネル)
	ウォームリセット	PhyPS=0または1	PhyPS=0
		PStateToAccess=0または1	PStateToAccess=0
		PhyPSMasterChannel=0	PhyPSMasterChannel=0
		PhySR=1 (チャンネル0)	PhySR=1 (チャンネル0)
		PhySR=0または1 (チャンネル0)	PhySR=0または1 (チャンネル0)
	S3	コールドリセットと同じ	コールドリセットと同じ
両方のチャンネル	コールドリセット	PhyPS=0	PhyPS=0
		PStateToAccess=0	PStateToAccess=0
		PhyPSMasterChannel=0	PhyPSMasterChannel=0
		PhySR=0 (すべてのチャンネル)	PhySR=0 (すべてのチャンネル)
	ウォームリセット	PhyPS=0または1	PhyPS=0
		PStateToAccess=0または1	PStateToAccess=0
		PhyPSMasterChannel=0	PhyPSMasterChannel=0
		PhySR=0または1 (チャンネル0)	PhySR=0または1 (チャンネル0)
		PhySR=1 (チャンネル1)	PhySR=1 (チャンネル1)
	S3	コールドリセットと同じ	コールドリセットと同じ

【0037】

典型的なコンピュータ電力供給が最初に通電される場合、様々な電圧が安定するのに時間がかかる。電圧が安定する前に、コンピュータを起動しようとするのが許可された場合、予測できない結果が生じ得る。コンピュータが時期尚早に起動するのを防ぐため、電力供給が使える状態であれば、電力供給はPwrOK信号を出力する。この信号が送信されるまで、マザーボードはコンピュータを起動するのを拒否する。

【0038】

コールドブート(～PwrOKがアサートされ、同様にResetがアサートされる)時には、PhyPSが自動的に0に設定され、PhyインタフェースはPhySRではないであろう(全てのDLLの電源が入れられる)。ウォームブート(PwrOKは継続的にアサートされ、Resetのみがアサートされる)時には、PhyPSおよびPhySR状態が、ウォームリセットの直前の状態によって決定される。ウォームリセットからの復帰は、BIOSが適切なPhyPSおよびPhySR状態を設定することを必要とする(時間最適化ソリューションは、これらの状態の直接CSR読取りを実行し、次いで必要

であれば、書込みを実行することである)。B I O Sは、ウォームリセットの後、P h y インタフェースが、S Rモードにないことを確実にすべきである。

【 0 0 3 9 】

D D R 3は、書込み平準化(W L)およびR x E nの曖昧でない収束を有するために、2つのパスのトレーニング手順を必要とするので、より高出力のP状態に対するトレーニングの過程の一部として低周波数P状態に対してトレーニングすることが可能である。その手順は、B I O Sは既に、ウォームリセットまたはS 3からの退出とは対照的に、これはコールドリセット(従って、トレーニングを必要とする)である判断しているとも仮定する。

【 0 0 4 0 】

メモリトレーニングの第1のパスは、以下の表7に示すような曖昧でないトレーニングに対して必要な初期のD D R レートを使用する。以下の表の全てで設定されるステップの特定の順番は、本開示の範囲から逸脱することなく、変更し得ることを理解すべきである。

【表 7】

	説明
1	メモリを備えたチャネルに対して以下の手順を実行する。メモリのないどのチャネルもS Rにし、これらのチャネルに対して他に何もしない。
2	関連するメモリコントローラレジスタ空間を構成する。
3	全てのアクティブなチャネルをS Rにする(随意)。
4	D D R レートを目標レートにプログラムする。
5	全てのアクティブなチャネルを正常モードに設定する(すなわち、チャネルをS Rから出す)。D L Lが再ロックされる。

【 0 0 4 1 】

この時点で、D D R P L LおよびD L Lが、使用するために適切に構成されている。P h y P SコンテキストはP h y P S 0である。メモリが存在する場合、両方のチャネルがトレーニングを開始する準備ができています。トレーニングは、表8に大まかに示すように、A M D G e n e r i c E n c a p s u l a t e d S o f t w a r e A r c h i t e c t u r e (A G E S A)ブートストラッププロトコルなどの、周知のプロトコルに従う：

【表 8】

	説明
6	全てのドライブ強度フィールドを設定する。
7	NB__T x C h n C l k E nを設定し、ハードウェアベースのフェンス (f e n c e) トレーニングを実行し、最終的なフェンス値を書き留めて設定する。
8	アドレス／バンク／R A S／C A S／W E／M e m C l k D l l 起動遅延を設定する。
9	C s & O d t D l l 起動遅延を設定する。
10	C k e D l l 起動遅延を設定する。
11	D R A M i n i tを実行する。
12	既定値定数を使用して、ハードウェアベースのW L トレーニングを実行する。
13	既定値定数を使用して、ハードウェアベースのR x E n トレーニングを実行する。
14	T x D q / R x D q s 書込み／読取りトレーニングを実行する。
15	M R L トレーニングがここで起こり得るか、または後で起こり得る。N C L Kは、このD D R レートに対する最終的な目標速度で実行する必要がある。

10

20

30

【0042】

この時点で、曖昧でないトレーニングに対する初期D D R レートのトレーニングおよびプログラミングが完了する。この初期トレーニングプロトコル中に確立されたこのD D R レートおよび他のパラメータが、第1の電力コンテキスト（例えば、より低速度 P h y P S [1]）として使用され得る。P h y P S [1]は、別のD D R レートに基づき設定され得る（すなわち、この周波数に対してステップ4～15を繰り返す）ことを理解すべきである。次の段階は、より高い（P h y P S [0]）D D R レートに対してトレーニングする。前述したパスは、書込み平準化およびR x E n ハードウェアのトレーニングアルゴリズムにおけるエイリアシングを解決した。書込み平準化およびR x E n トレーニング値は、M e m C l k データレート比P h y P S 0 / P h y P S 1によってスケールされ、P h y P S [0]におけるトレーニングの第2の段階に対するシードとして使用される。手順は、表9に示すように開始される。

40

【表 9】

	説明
16	全てのアクティブなチャンネルをSRにする（随意）。
17	DDRレートを所望のPhyPS[0] DDRレートにプログラミングする。
18	全てのアクティブなチャンネルを正常モードに設定する（すなわち、チャンネルをSRから出す）。DLLが再ロックされる。

10

【0043】

この時点で、DDR PLLおよびDLLが、使用するために適切に構成される。PhyPSコンテキストはまだPhyPS[0]であり、PhyインタフェースはPhyPS[0] DDRレートで実行する。両方のチャンネルは、メモリが存在する場合、表8に大まかに示すように、トレーニングの次の段階を開始する準備ができています：

【表 10】

	説明
19	全てのドライブ強度フィールドを設定する。
20	NB_TxClnClkEnを設定し、ハードウェアベースのフェンストレーニングを実行し、最終的なフェンス値を書き留めて設定する。
21	アドレス／バンク／RAS／CAS／WE／MemClk DLL起動遅延を設定する。
22	Cs&Od t Dll起動遅延を設定する。
23	Ck e Dll起動遅延を設定する。
24	DRAM i n i tを実行する。
25	既定値定数を使用して、ハードウェアベースのWLトレーニングを実行する。
26	既定値定数を使用して、ハードウェアベースのRxEnトレーニングを実行する。
27	TxDq/RxDqs書き込み／読取りトレーニングを実行する。
28	MRLトレーニングがここで起こり得るか、または後で起こり得る。NCLKは、このDDRレートに対する最終的な目標速度で実行する必要がある。

20

30

40

【0044】

これで、電力コンテキスト PhyPS[0]の第2の段階のトレーニングおよびプログラミングが完了する。この時点で、両方のPhyPS空間に対するレートがトレーニングされ、トレーニングされた値が、トレーニングの一部としてPhyインタフェースPhyPS[0] CSRに対して既に入力されている。Phyインタフェースは、現在、

50

PhyPS[0] コンテキスト内である。PhyPS[1] トレーニング値はトレーニングされているが、PhyPS[1] CSRにまだ書き込まれていない。PhyPS[0] CSRは、表11に大まかに示すように更新される：

【表11】

	説明
29	1をMaster_0x08[PSStateToAccess] に書き込む—PhyPS[1]
30	格納されたPhyPS1 トレーニング値をPhyPS1 CSRに書き込む
31	0をMaster_0x08[PSStateToAccess] に書き込む—PhyPS[0]

10

【0045】

1つのPhyPSのみが必要とされる場合には、BIOSがどのPhyPSコンテキストを使用すべきかを選択し得る。歴史的な意味を維持するため、BIOSはPhyPS[0] に対してPhyインタフェースを構成し得る。

20

【0046】

S3からの復帰は、いかなるDRAMトレーニングも伴わず、不揮発性状態（通常、サウスブリッジにおける）からのトレーニング値の復元のみである。S3からの復帰は、一般に、（～PwrOkのために）PhyPSコンテキストがPhyPS0であり、Master_0x08[PSStateToAccess] が0であること、両方のチャンネルがPhySRから出ていること、およびマスターチャンネルがチャンネル0である（たとえ、メモリがチャンネル0上に存在しなくても）ことを保証する。S3からの復帰に対する手順を大まかに表12に示す：

【表 1 2】

	説明	
1	全てのアクティブなチャンネルをSRにする。	
2	チャンネル0上にメモリがない場合、アクティブなDCTのみによってチャンネル1上で発行されているPhyPS変更が影響を与えることを確実にするため、Master_0x18 [PhyPSMasterChannel] (3.7を参照)を設定する。メモがチャンネル1上に存在しない場合には、Master_18 [PhyPSMasterChannel] はチャンネル0の既定値のままにされるべきであるので、動作は必要とされないことに留意する。	10
3	S3中に発行されている~PwrOkのためにPhyインタフェースがPhyPS0内にあると、PhyPS0トレーニング値および導出値の全てを復元する。	20
4	DDRレートをPhyPS0レートで更新する。	
5	1をマスターチャンネル内のMaster_0x08 [PS tateToAccess] に書き込む。これは、全てのPCIアクセスがここでPhyPS=1で向けられることを意味する。	
6	全てのPhyPS1トレーニング値および導出値を復元する。	30
7	DDRレートをPhyPS1レートで更新する。ここで、PhyPS CSRの両方のセットが復元され、準備ができています。	
8	0x0B [PhyPSの所望のPhyPSへの変更] をマスターチャンネル内で発行する。これは、DDR PLLを所望のPhyPSレートに更新する。CfgDoneがアサートされるのを待機する。	40
9	全てのアクティブなチャンネルをSRから出す。未実装のチャンネルはSRに残す。DLLが再ロックされる。PhyインタフェースはここでいずれかのPhyPSで動作する準備ができ、現在は要求されたPhyPS内である。	

【0047】

ウォームリセットの復帰は、S3からの復帰とほぼ同一である。S3からの復帰は、~PwrOkで、PhyPSコンテキストをPhyPS0に設定し、PhyPS変更を伝達するためにチャンネル0をマスターチャンネルとして指定し、両方のチャンネルでPhyインタフェースをPhySRから出す。対照的に、ウォームリセットは、PhyPSおよびPh

10

20

30

40

50

y S Rをマスターチャンネルと同様に未知の状態のままにしておく。ウォームリセットの復帰には構造上の穴が存在することにさらに留意すべきである。システがコールドブートし、メモリトレーニングを完了し、全てのトレーニング値を不揮発性メモリに格納し得る前にウォームリセット復帰が発行されると、復帰は失敗するであろう。この問題を回避するため、B I O Sは、トレーニング値が計算され、うまく格納されているか否かを示すために（コールドリセット時にリセットされるが、ウォームリセットを通じて持続する）フラグを使用すべきであり、ウォームリセット中に、B I O Sがこのフラグが設定されているのを見ると、トレーニングされた状態を復元することによって復帰し得る。トレーニングされた状態は、M a s t e r _ 0 x 1 8 [P h y P S M a s t e r C h a n n e l] も含み、同様に、どの未実装のチャンネルもS Rモードのままにされる。このフラグが設定されていなければ、B I O SはP h yを（再）トレーニングする必要がある。以下の開示では、このフラグが設定されていて、トレーニング値が格納されていると仮定する。従って、ウォームリセット復帰はいかなるD R A Mトレーニングも伴わず、（通常、サウスブリッジ内の）不揮発性状態からトレーニング値を復元するだけである。ウォームリセット復帰を実行するための手順を大まかに表 1 3 に示す：

【表 1 3】

	説明
1	全てのアクティブなチャンネルをS Rにする。
2	B I O Sが、どのチャンネルがP h y P S変更を受け入れるマスターチャンネルであるかを決定する。B I O SによるM a s t e r _ 0 x 0 8 [P S t a t e T o A c c e s s] または0 x 0 B [P h y P S] コマンドへの以後のアクセスは、マスターチャンネルへ向けられるべきである。
3	P h y P S [0] を選択する（M a s t e r _ 0 x 0 8 [P S t a t e T o A c c e s s] = 0）
4	全てのP h y P S 0のトレーニング値および導出値を復元する。
5	DDRレートをP h y P S [0] レートに更新する。
6	P h y P S [1] を選択する（M a s t e r _ 0 x 0 8 [P S t a t e T o A c c e s s] = 1）
7	全てのP h y P S 1のトレーニング値および導出値を復元する。
8	DDRレートをP h y P S 1レートに更新する。
9	0 x 0 B [P h y P Sの所望のP h y P Sへの変更] をマスターチャンネル内で発行する。これは、DDR P L Lを所望のP h y P Sレートに更新する。C f g D o n eがアサートするのを待機する。
1 0	全てのアクティブなチャンネルをS Rから出す。これによりD L Lを再ロックする。

【 0 0 4 8 】

表 1 4 は、各電力コンテキストに対して重複されている全てのP h y P S C S Rのリストを示す。

【表 14】

	説明
1	PhyPS0のためのMaster_0x00 [Config0]
2	PhyPS1のためのMaster_0x40 [Config0]
3	PhyPS0のためのMaster_0x01 [Config1]
4	PhyPS1のためのMaster_0x41 [Config1]
5	PhyPS0のためのMaster_0x02 [AddrCmdCke]
6	PhyPS1のためのMaster_0x42 [AddrCmdCke]
7	Master_0x03 [PllDiv, PllMult, PllDiv2Bypass, PllUpdate] は現在、PhyPS0 PLL制御用である
8	Master_0x43 [PllDiv, PllMult, PllDiv2Bypass, PllUpdate] は現在、PhyPS1 PLL制御用である
9	8:0. DByte_0x {00, 04, 08} はPhyPS0 DSおよびPODt制御である
10	8:0. DByte_0x {40, 44, 48} はPhyPS1 DSおよびPODt制御である
11	8:0. DByte_0x0CはPhyPS0 Dll&Pre Configである
12	8:0. DByte_0x0CはPhyPS1 Dll&Pre Configである
13	8:0. DByte_0x {10, 11} はPhyPS0 Dll電力管理である
14	8:0. DByte_0x {50, 51} はPhyPS1 Dll電力管理である
15	8:0. DByte_0x13はPhyPS0 Dll DByte Configである
16	8:0. DByte_0x53はPhyPS1 Dll DByte Configである
17	PhyPS0のための8:0. DByte_0x {14, 15, 16, 17} [RxDQS初期/後期選択]
18	PhyPS1のための8:0. DByte_0x {54, 55, 56, 57} [RxDQS初期/後期選択]
19	PhyPS0のための8:0. DByte_0x {18, 19, 1A, 1B} [RxDQS初期終止]
20	PhyPS1のための20. 8:0. DByte_0x {58, 59, 5A, 5B} [RxDQS初期終止]
21	PhyPS0のための8:0. DByte_0x1E [バイアス制御]
22	PhyPS1のための8:0. DByte_0x5E [バイアス制御]
23	PhyPS0のための8:0. DByte_0x1F [受信器制御]
24	PhyPS1のための8:0. DByte_0x5F [受信器制御]
25	PhyPS0のためのRxDqs後期遅延 {0, 1, 2, 3} に対する8:0. DByte_0x {20, 21, 22, 23}
26	PhyPS1のためのRxDqs後期遅延 {0, 1, 2, 3} に対する8:0. DByte_0x {60, 61, 62, 63}
27	PhyPS0のためのDllTxDqsに対する8:0. DByte_0x {24, 25}
28	PhyPS1のためのDllTxDqsに対する8:0. DByte_0x {64, 65}
29	PhyPS0のためのDllRxEnに対する8:0. DByte_0x {26, 27}
30	PhyPS1のためのDllRxEnに対する8:0. DByte_0x {66, 67}
31	PhyPS0のためのDllTxDqに対する8:0. DByte_0x {28, 29}
32	PhyPS1のためのDllTxDqに対する8:0. DByte_0x {68, 69}
33	PhyPS0のためのDllTxUDQSに対する8:0. DByte_0x {2A, 2B}
34	PhyPS1のためのDllTxUDQSに対する8:0. DByte_0x {6A, 6B}
35	PhyPS0のためのDllTxUDqに対する8:0. DByte_0x {2C, 2D}
36	PhyPS1のためのDllTxUDqに対する8:0. DByte_0x {6C, 6D}
37	3:0. MemCk_0x00 [TX Config/DS] はPhyPS0 DSである
38	3:0. MemCk_0x40 [TX Config/DS] はPhyPS1 DSである
39	3:0. MemCk_0x1E [バイアス制御] はPhyPS0 DSである
40	3:0. MemCk_0x5E [バイアス制御] はPhyPS1 DSである
41	PhyPS0のための3:0. MemCk_0x20 [Dll遅延/コンフィグ]
42	PhyPS1のための3:0. MemCk_0x60 [Dll遅延/コンフィグ]
43	0:1. CsOdt_0x {00, 04, 08} [OdtCfg, CsCfg0145, CsCfg2367] はPhyPS0 DSである
44	0:1. CsOdt_0x {40, 44, 48} [OdtCfg, CsCfg0145, CsCfg2367] はPhyPS1 DSである
45	PhyPS0のための0:1. CsOdt_0x1E [バイアス制御]
46	PhyPS1のための0:1. CsOdt_0x5E [バイアス制御]
47	PhyPS0のための0:1. CsOdt_0x {20, 21} [Dll遅延/コンフィグ]
48	PhyPS1のための0:1. CsOdt_0x {60, 61} [Dll遅延/コンフィグ]
49	A2Byt_0x {00, 04, 08, 0C, 10} [A2CfgCke, A2Cfg0189, A2Cfg23AB, A2Cfg45CD, A2Cfg67EF] はPhyPS0 DSである
50	A2Byt_0x {40, 44, 48, 4C, 50} [A2CfgCke, A2Cfg0189, A2Cfg23AB, A2Cfg45CD, A2Cfg67EF] はPhyPS1 DSである
51	PhyPS0のためのA2Byt_0x1E [バイアス制御]
52	PhyPS1のためのA2Byt_0x5E [バイアス制御]
53	PhyPS0のためのA2Byt_0x {20, 21} [Dll遅延/コンフィグ]
54	PhyPS1のためのA2Byt_0x {60, 61} [Dll遅延/コンフィグ]
55	PhyPS0のためのD3Cmp_0x02 [ブレイクパスルーレート]

本明細書の開示に基づいて、多数の変形形態が可能であることを理解すべきである。例えば、複数の電力コンテキストが他のメモリ位置（例えば、ノースブリッジ14またはサウスブリッジ16内）に格納できる。このシナリオでは、標準的なPhyインタフェースは、コンテキスト依存CSRの専用セットを必要とせずに、電力コンテキストを切り替え得る。かかるシナリオは、CSRの複数のセットに対する必要性を取り除き得るが、コンテキスト変更に対する待機時間を増加させる。

【0050】

特徴および要素は特定の組合せで前述されているが、各特徴または要素は、他の特徴および要素なしで単独で、または他の特徴および要素の有無に関わらず様々な組合せで使用され得る。本明細書で提供する方法または流れ図は、汎用コンピュータまたはプロセッサによる実行のために、コンピュータ可読記憶媒体内に組み込まれた、コンピュータプログラム、ソフトウェア、またはファームウェアで実装され得る。コンピュータ可読記憶媒体の例には、読取り専用メモリ（ROM）、ランダムアクセスメモリ（RAM）、レジスタ、キャッシュメモリ、半導体メモリデバイス、内蔵されたハードディスクおよび取り外し可能ディスクなどの磁気媒体、光磁気媒体、ならびにCD-ROMディスク、およびデジタル多用途ディスク（DVD）などの光媒体を含む。

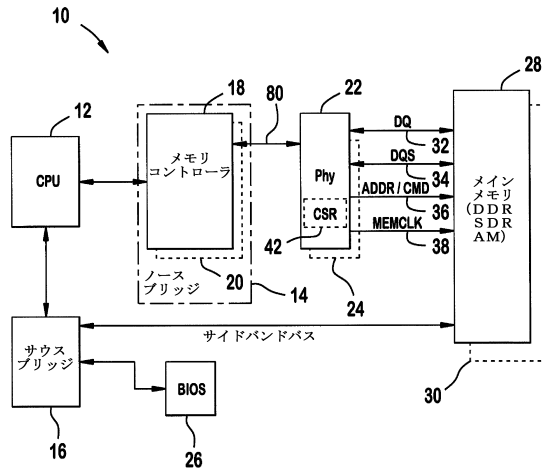
【0051】

適切なプロセッサには、例として、汎用プロセッサ、専用プロセッサ、従来型のプロセッサ、デジタル信号プロセッサ（DSP）、複数のマイクロプロセッサ、DSPコアと関連した1つまたは複数のマイクロプロセッサ、コントローラ、マイクロコントローラ、特定用途向け集積回路（ASIC）、フィールドプログラマブルゲートアレイ（FPGA）回路、任意の他のタイプの集積回路（IC）、および/または状態マシンを含む。かかるプロセッサは、処理されたハードウェア記述言語（HDL）命令（コンピュータ可読媒体上に格納可能なかかる命令）の結果を用いて、製造プロセスを構成することにより製造され得る。かかる処理の結果は、本発明の態様を実施するプロセッサを製造するために、その後、半導体製造プロセスで使用されるマスクワークであり得る。

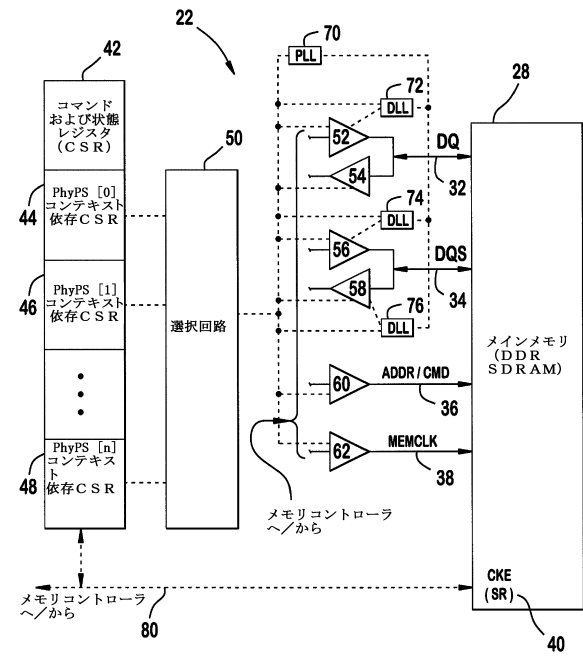
10

20

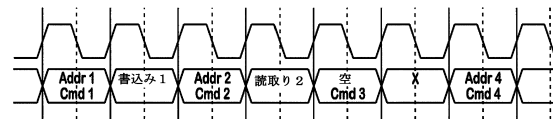
【図 1】



【図 2】



【図 3】



【図 4】

アドレス	値	機能
[15:13]	000	チップレットタイプ: D3Dby te (9インスタンス)
	001	チップレットタイプ: D3C l k (4インスタンス)
	010	チップレットタイプ: D3C mp (3インスタンス)
	011	チップレットタイプ: D3MemV re f (2インスタンス)
	100	チップレットタイプ: D3CsO d t (1インスタンス)
	101	チップレットタイプ: D3ABy te l (1インスタンス)
	110	チップレットタイプ: D3ABy te (1インスタンス)
[12]	0/1	1の場合、アドレスビット[5, 1, 0]に一致するチップレット内の全てのレジスタを更新する: 0の場合、全てのアドレスビット[5: 0]を復号する
[11:8]	1011-0000	チップレットタイプのインスタンス数。最大12のインスタンスが許可される
	1100	[7: 6]に一致するすべてのODT補償レジスタへのブロードキャスト
	1101	[7: 6]に一致するすべてのドライバ補償レジスタへのブロードキャスト
	1110	[7: 6]に一致するすべてのブレイク補償レジスタへのブロードキャスト
[7:6]	11-00	タイプフィールド[15:13]に一致するすべてのチップレットへのブロードキャスト
	11-00	補償情報をブロードキャストする([11: 8])場合に、情報源を識別する最大4つのコード。各チップレットは、それに関連するD3CMPブロックを識別する2ビットのメタルストラップコードを有する。補償ブロードキャストが生じる場合、[7: 6]は、データが供給されたD3CMPを識別する
[5:0]	1x_xxxx	DLLレジスタ
	0x_xx00	ドライバ強度レジスタ
	0x_xx01	ドライバ補償レジスタ
	0x_xx10	ブレイク補償レジスタ
	0x_xx11	ODT補償レジスタ (D3DBYTEのみ)

フロントページの続き

(74)代理人 100162156

弁理士 村雨 圭介

(72)発明者 ショーン サールズ

アメリカ合衆国 7 8 7 3 7 テキサス州、オースティン、トワイライト メサ ドライブ 6 9
1 2

(72)発明者 ニコラス ティー . ハンフリーズ

アメリカ合衆国 7 8 7 5 7 テキサス州、オースティン、ピエモント アベニュー 1 7 1 7

(72)発明者 ブライアン ダブリュ . アミック

アメリカ合衆国 0 1 7 3 0 マサチューセッツ州、ベッドフォード、ヒルクレスト ロード 4

(72)発明者 リチャード ダブリュ . リーブス

アメリカ合衆国 0 1 5 8 1 マサチューセッツ州、ウェストボロー、ベルクナップ ストリート
9

(72)発明者 ハンウー チョウ

アメリカ合衆国 0 1 7 2 0 マサチューセッツ州、アクトン、フォークナー ヒル ロード 3
8

(72)発明者 ロナルド エル . ペティジョン

アメリカ合衆国 0 1 7 4 2 マサチューセッツ州、コンコード、ミニステリアル ドライブ 1
4 3

審査官 滝谷 亮一

(56)参考文献 特開 2 0 0 7 - 2 4 9 7 3 8 (J P , A)

特開 2 0 1 0 - 1 6 0 7 2 4 (J P , A)

特開 2 0 0 7 - 2 4 1 7 9 9 (J P , A)

特表 2 0 0 4 - 5 2 8 6 1 3 (J P , A)

米国特許出願公開第 2 0 0 3 / 0 2 3 7 0 5 6 (U S , A 1)

米国特許第 0 6 6 1 8 7 9 1 (U S , B 1)

(58)調査した分野(Int.Cl. , D B 名)

G 0 6 F 1 2 / 0 0