

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2018-502362  
(P2018-502362A)

(43) 公表日 平成30年1月25日(2018.1.25)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G06F 13/38 (2006.01)</b>	G06F 13/38 340A	5B061
<b>G06F 13/36 (2006.01)</b>	G06F 13/36 520Z	5B077
	G06F 13/36 510	

審査請求 未請求 予備審査請求 未請求 (全 65 頁)

(21) 出願番号 特願2017-523883 (P2017-523883)  
 (86) (22) 出願日 平成27年11月2日 (2015.11.2)  
 (85) 翻訳文提出日 平成29年6月30日 (2017.6.30)  
 (86) 国際出願番号 PCT/US2015/058688  
 (87) 国際公開番号 W02016/070197  
 (87) 国際公開日 平成28年5月6日 (2016.5.6)  
 (31) 優先権主張番号 14/530, 203  
 (32) 優先日 平成26年10月31日 (2014.10.31)  
 (33) 優先権主張国 米国 (US)

(71) 出願人 390020248  
 日本テキサス・インスツルメンツ株式会社  
 東京都新宿区西新宿六丁目24番1号  
 (71) 出願人 507107291  
 テキサス インスツルメンツ インコーポ  
 レイテッド  
 アメリカ合衆国 テキサス州 75265  
 -5474 ダラス メール ステーショ  
 ン 3999 ピーオーボックス 655  
 474  
 (74) 上記1名の代理人 100098497  
 弁理士 片寄 恭三

最終頁に続く

(54) 【発明の名称】 ノンブロッキング高性能トランザクションクレジットシステムを備えるマルチコアバスアーキテクチャ

(57) 【要約】

説明する例のバス通信プロトコルでは、マスターデバイスがバスクレジットをストアする(2402)。マスターデバイス(2410)は、充分な数及びタイプのバスクレジットを持つ場合のみ、バストランザクション(2414、2415、2415)を送信し得る。送信の際、マスターデバイスはストアされたバスクレジットの数を低減する。バスクレジットは、バストランザクションを受け取るスレーブデバイス上のリソースに対応する。スレーブデバイス(2420)は、適切なクレジットを伴う場合、バストランザクション(2424、2425、2426)を受け取らねばならない。スレーブデバイスは、トランザクションをサービスし、クレジットリターンを送信する。マスターデバイスは、対応する数及びタイプのクレジットを、ストアされた量に加算する。スレーブデバイスは別のバストランザクションをアクセプトする準備ができ、マスターデバイスはバストランザクションを再び開始可能とされる。多くのタイプの相互作用において、バスエージェントはプロセスの状態に応じてマスターとスレーブの両方として機能し得る。

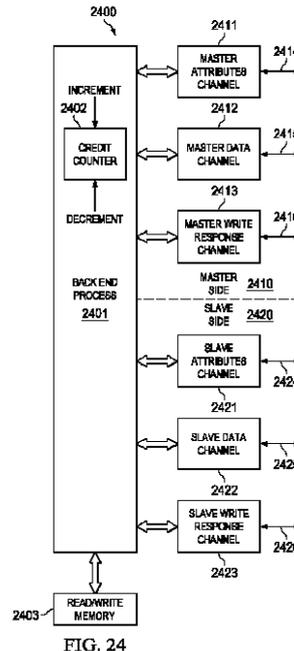


FIG. 24

**【特許請求の範囲】****【請求項 1】**

第 1 のデバイスと第 2 のデバイスとの間のバス通信の方法であって、  
前記第 1 のデバイスにおいて複数のバスクレジットをストアすること、  
前記第 1 のデバイスが、バストランザクションを前記第 2 のデバイスにバス上で送信すること、  
前記第 1 のデバイスがバストランザクションを前記第 2 のデバイスに前記バス上で送信すると、前記第 1 のデバイスが、  
前記ストアされたバスクレジットの数を、前記バストランザクションに対応する量だけ減少させること、及び  
対応するバスクレジット信号を前記第 2 のデバイスに前記バス上で送信することであって、  
前記ストアされたバスクレジットの数が前記バストランザクションに対応する前記量より少ない場合、前記第 1 のデバイスがバストランザクションを送信しない可能性があること、及び  
前記第 1 のデバイスが、前記バスを介して前記第 2 のデバイスからクレジットリターンを受け取ると、前記ストアされたバスクレジットの数を前記クレジットリターンに対応する量だけ増加させること、  
を含む、方法。

10

**【請求項 2】**

20

請求項 1 に記載の方法であって、  
前記バストランザクションがコマンドであり、  
前記バストランザクションを前記バス上で送信することが、前記コマンドに対応する信号を前記バス上で送信することを含む、  
方法。

**【請求項 3】**

請求項 1 に記載の方法であって、  
前記バストランザクションが読み出しであり、  
前記バストランザクションを前記バス上で送信することが、読み出しアドレスの範囲を示す少なくとも 1 つの信号を前記バス上で送信することを含む、  
方法。

30

**【請求項 4】**

請求項 3 に記載の方法であって、  
読み出しアドレスの範囲を示す少なくとも 1 つの信号を送信することが、  
初期読み出しアドレスを示す第 1 の信号を送信すること、及び  
読み出し長を示す第 2 の信号を送信すること、  
を含む、方法。

**【請求項 5】**

請求項 1 に記載の方法であって、  
前記バストランザクションが書き込みであり、前記バストランザクションを前記バス上で送信することが、  
書き込みアドレスの範囲を示す少なくとも 1 つの信号を送信すること、及び  
書き込みデータに対応する信号を前記バス上で送信すること、  
を含む、方法。

40

**【請求項 6】**

請求項 5 に記載の方法であって、書き込みアドレスの範囲を示す少なくとも 1 つの信号を送信することが、  
初期書き込みアドレスを示す第 1 の信号を送信すること、及び  
書き込み長を示す第 2 の信号を送信すること、  
を含む、方法。

50

## 【請求項 7】

請求項 1 に記載の方法であって、前記第 1 のデバイスが第 1 のトランザクション識別を前記バスを介して前記第 2 のデバイスに送信するステップを更に含む、方法。

## 【請求項 8】

請求項 7 に記載の方法であって、前記第 2 のデバイスが、トランザクション応答を前記バスを介して前記第 1 のデバイスに送信することを更に含み、前記トランザクション応答が、前記第 1 のトランザクション識別に対応する第 2 のトランザクション識別を含む、方法。

## 【請求項 9】

請求項 8 に記載の方法であって、  
前記バストランザクションが読み出しであり、  
前記トランザクション応答が前記読み出しに対応するデータを含む、  
方法。

10

## 【請求項 10】

第 1 のデバイスと第 2 のデバイスとの間のバス通信の方法であって、  
バスを介して前記第 2 のデバイスからバストランザクション及びバスクレジット信号を受け取ること、  
前記バスクレジット信号が前記バストランザクションに対応する場合、前記第 1 のデバイスにおいて前記バストランザクションをアクセプトすること、  
前記第 1 のデバイスにおいて前記バストランザクションをサービスすること、及び  
前記第 1 のデバイスにおいて前記バストランザクションをサービスすることが完了すると、前記第 1 のデバイスが前記バスを介して前記第 2 のデバイスにクレジットリターンを送信すること、  
を含む、方法。

20

## 【請求項 11】

請求項 10 に記載の方法であって、  
前記バストランザクションがコマンドであり、  
前記第 1 のデバイスにおいて前記バストランザクションをサービスすることが、前記コマンドを実行することを含む、  
方法。

30

## 【請求項 12】

請求項 10 に記載の方法であって、前記バストランザクションが読み出しであり、前記バストランザクションが読み出しアドレスの範囲を示す少なくとも 1 つの信号を含み、  
前記第 1 のデバイスにおいて前記バストランザクションをサービスすることが、  
読み出しアドレスの前記範囲において前記第 1 のデバイスからデータをリコールすることと、  
前記第 1 のデバイスから前記リコールされたデータを、前記バスを介して前記第 2 のデバイスに送信することと、  
を含む、  
方法。

40

## 【請求項 13】

請求項 12 に記載の方法であって、読み出しアドレスの範囲を示す前記少なくとも 1 つの信号が、初期読み出しアドレスを示す第 1 の信号、及び読み出し長を示す第 2 の信号を含む、方法。

## 【請求項 14】

請求項 10 に記載の方法であって、  
前記バストランザクションが書き込みであり、  
前記バストランザクションが、書き込みアドレスの範囲を示す少なくとも 1 つの信号、  
及び書き込みデータに対応する信号を含み、  
前記第 1 のデバイスにおいて前記バストランザクションをサービスすることが、前記書

50

き込みデータを書き込みアドレスの前記範囲に対応するアドレスにおいてストアすることを含む、

方法。

【請求項 15】

請求項 14 に記載の方法であって、書き込みアドレスの範囲を示す前記少なくとも 1 つの信号が、初期書き込みアドレスを示す第 1 の信号、及び書き込み長を示す第 2 の信号を含む、方法。

【請求項 16】

請求項 10 に記載の方法であって、前記バストランザクションが第 1 のトランザクション識別を含む、方法。

10

【請求項 17】

請求項 16 に記載の方法であって、

前記第 1 のデバイスにおいて前記バストランザクションをサービスすることが完了すると、前記第 1 のデバイスが、トランザクション応答を前記バスを介して前記第 2 のデバイスに送信することを更に含み、

前記トランザクション応答が、前記第 1 のトランザクション識別に対応する第 2 のトランザクション識別を含む、

方法。

【請求項 18】

第 1 のデバイスと第 2 のデバイスとの間のバス通信の方法であって、

20

前記第 1 のデバイスにおいて複数のバスクレジットをストアすること、

前記第 1 のデバイスが、バストランザクションを前記第 2 のデバイスにバス上で送信すること、

前記第 1 のデバイスがバストランザクションを前記第 2 のデバイスに前記バス上で送信すると、前記第 1 のデバイスが、

バスクレジットの前記ストアされた数を、前記バストランザクションに対応する量だけ減少させること、及び

対応するバスクレジット信号を、前記第 2 のデバイスに前記バスで送信することであって、

30

ストアされたバスクレジットの前記数が前記バストランザクションに対応する前記量より少ない場合、前記第 1 のデバイスがバストランザクションを送信しない可能性があること、

前記第 2 のデバイスにおいて、前記バストランザクション及び前記バスクレジット信号を、前記バスを介して前記第 1 のデバイスから受け取ること、

前記バスクレジット信号が前記バストランザクションに対応する場合、前記第 2 のデバイスにおいて前記バストランザクションをアクセプトすること、

前記第 2 のデバイスにおいて前記バストランザクションをサービスすること、

前記第 1 のデバイスにおいて前記バストランザクションをサービスすることを完了すると、前記第 2 のデバイスが、前記バスを介して前記第 1 のデバイスにクレジットリターンを送信すること、及び

40

前記第 1 のデバイスが前記バスを介して前記第 2 のデバイスからクレジットリターンを受け取ると、バスクレジットの前記ストアされた数を前記クレジットリターンに対応する量だけ増加させること、

を含む、方法。

【請求項 19】

請求項 18 に記載の方法であって、

前記バストランザクションがコマンドであり、

前記バストランザクションを前記バス上で送信することが、前記コマンドに対応する信号を前記バス上で送信することを含み、

前記第 2 のデバイスにおいて前記バストランザクションをサービスすることが、前記コ

50

マンドを実行することを含む、  
方法。

【請求項 20】

請求項 18 に記載の方法であって、  
前記バストランザクションが読み出しであり、  
前記バストランザクションを前記バス上で送信することが、読み出しアドレスの範囲を示す少なくとも 1 つの信号を前記バス上で送信することを含み、  
前記第 2 のデバイスにおいて前記バストランザクションをサービスすることが、  
前記読み出しアドレスの範囲において前記第 2 のデバイスからデータをリコールすることと、  
前記第 2 のデバイスから前記リコールされたデータを、前記バスを介して前記第 1 のデバイスに送信することと、  
を含む、  
方法。

10

【請求項 21】

請求項 20 に記載の方法であって、読み出しアドレスの範囲を示す少なくとも 1 つの信号を送信することが、初期読み出しアドレスを示す第 1 の信号を送信すること、及び読み出し長を示す第 2 の信号を送信することを含む、方法。

【請求項 22】

請求項 18 に記載の方法であって、  
前記バストランザクションが書き込みであり、  
前記バストランザクションを前記バス上で送信することが、書き込みアドレスの範囲を示す少なくとも 1 つの信号を送信すること、及び書き込みデータに対応する信号を前記バス上で送信することを含み、  
前記第 2 のデバイスにおいて前記バストランザクションをサービスすることが、書き込みアドレスの前記範囲に対応するアドレスにおいて前記書き込みデータをストアすることを含む、  
方法。

20

【請求項 23】

請求項 22 に記載の方法であって、書き込みアドレスの範囲を示す少なくとも 1 つの信号を送信することが、初期書き込みアドレスを示す第 1 の信号を送信すること、及び書き込み長を示す第 2 の信号を送信することを含む、方法。

30

【請求項 24】

請求項 18 に記載の方法であって、  
前記第 1 のデバイスが、記バスを介して第 1 のトランザクション識別を前記第 2 のデバイスに送信すること、及び  
前記第 1 のデバイスにおいて前記バストランザクションをサービスすることが完了すると、前記第 2 のデバイスが、前記バスを介して前記第 1 のデバイスにトランザクション応答を送信すること、  
を更に含み、  
前記トランザクション応答が、前記第 1 のトランザクション識別に対応する第 2 のトランザクション識別を含む、  
方法。

40

【請求項 25】

バス上で情報を交換するように動作可能なバスエージェントであって、  
中に複数のバスクレジットをストアするクレジットカウンタ、  
前記バスに及び前記クレジットカウンタに接続される属性チャネルであって、  
バストランザクションを前記バス上で送信すること、  
前記クレジットカウンタにストアされたバスクレジットの数を、前記バストランザクションに対応する量だけ減少させること、及び

50

対応するバスクレジット信号を前記バス上で送信すること、  
 によってバストランザクションを開始するように動作可能である、前記属性チャンネル、  
 を含み、

ストアされたバスクレジットの前記数が、前記バストランザクションに対応する前記量  
 より少ない場合、前記属性チャンネルが前記バストランザクションを送信しない可能性があ  
 り得、

前記属性チャンネルが更に、

前記バスからのクレジットリターンを受け取るように、及び

前記クレジットカウンタにストアされたバスクレジットの前記数を前記クレジットリ  
 ターンに対応する量だけ増加させるように、

動作可能である、

バスエージェント。

10

【請求項 26】

請求項 25 に記載のバスエージェントであって、前記バストランザクションが読み出し  
 であり、前記属性チャンネルが、読み出しアドレスの範囲を示す少なくとも 1 つの信号を送  
 信することによって、前記読み出しを前記バス上で送信するように動作可能である、バス  
 エージェント。

【請求項 27】

請求項 26 に記載のバスエージェントであって、

前記属性チャンネルが、

初期読み出しアドレスを示す第 1 の信号を送信すること、及び

読み出し長を示す第 2 の信号を送信すること、

によって読み出しアドレスの範囲を示す前記少なくとも 1 つの信号を送信するように動  
 作可能である、

バスエージェント。

20

【請求項 28】

請求項 25 に記載のバスエージェントであって、

前記バストランザクションが書き込みであり、

前記属性チャンネルが、書き込みアドレスの範囲を示す少なくとも 1 つの信号を送信す  
 ることによって、前記書き込みを前記バス上で送信するように動作可能であり、

書き込みデータに対応する信号を送信するように、書き込みの際に動作可能なデータチ  
 ャネルを更に含む、

バスエージェント。

30

【請求項 29】

請求項 28 に記載のバスエージェントであって、

前記属性チャンネルが、

初期書き込みアドレスを示す第 1 の信号を送信すること、及び

書き込み長を示す第 2 の信号を送信すること、

によって、書き込みアドレスの範囲を示す前記少なくとも 1 つの信号を送信するよう  
 に動作可能である、

バスエージェント。

40

【請求項 30】

請求項 25 に記載のバスエージェントであって、前記属性チャンネルが更に、トランザク  
 ション識別を送信することによって、前記バストランザクションを前記バス上で送信す  
 るように動作可能である、バスエージェント。

【請求項 31】

バス上で情報を交換するように動作可能なバスエージェントであって、

前記バスに接続される属性チャンネルであって、前記属性チャンネルが、

前記バスからバストランザクション及びバスクレジット信号を受け取るように、及び

前記バスクレジット信号が前記バストランザクションに対応する場合、前記バストラン

50

ザクシオンをアクセプトするように、

動作可能である、前記属性チャンネル、及び

前記属性チャンネルによってアクセプトされるバストランザクシオンをサービスするように動作可能である前記属性チャンネルに接続されるバックエンドプロセス、

を含み、

前記属性チャンネルが更に、前記バックエンドプロセスによって前記バストランザクシオンをサービスすることが完了すると、クレジットリターンを前記バス上で送信するように動作可能である、

バスエージェント。

【請求項 3 2】

10

請求項 3 1 に記載のバスエージェントであって、

前記バスプロセスに結合される読み出し / 書き込みメモリを更に含み、

前記バストランザクシオンが読み出しであり、

前記属性チャンネルが、読み出しアドレスの範囲を示す少なくとも 1 つの信号を受け取ることを含み、前記読み出しを前記バス上で受け取るように動作可能であり、

前記バックエンドプロセスが、前記読み出し / 書き込みメモリから、前記読み出しアドレスの範囲に対応するアドレスから、データをリコールすることによって、前記読み出しをサービスするように動作可能であり、

更に、前記バス及び前記読み出し / 書き込みメモリに接続されるデータチャンネルを含み、前記データチャンネルが、前記読み出し / 書き込みメモリからリコールされたデータを前記バス上で送信するように動作可能である、

20

バスエージェント。

【請求項 3 3】

請求項 3 2 に記載のバスエージェントであって、

前記属性チャンネルが、

初期読み出しアドレスを示す第 1 の信号を受け取ること、及び

読み出し長を示す第 2 の信号を受け取ること、

によって、読み出しアドレスの範囲を示す前記少なくとも 1 つの信号を受け取るように動作可能である、

バスエージェント。

30

【請求項 3 4】

請求項 3 1 に記載のバスエージェントであって、

前記バスプロセスに接続される読み出し / 書き込みメモリを更に含み、

前記バストランザクシオンが書き込みであり、

前記属性チャンネルが、書き込みアドレスの範囲を示す少なくとも 1 つの信号を受け取ることを含み、前記書き込みを前記バス上で受け取るように動作可能であり、

更に、前記バスに接続されるデータチャンネルを含み、前記データチャンネルが、前記バスから書き込みデータを受け取るように動作可能であり、

前記バックエンドプロセスが更に、前記読み出し / 書き込みメモリに結合され、更に、書き込みアドレスの前記範囲に対応するアドレスにおいて前記バスから前記読み出し / 書き込みメモリに書き込みデータストアすることによって、前記書き込みをサービスするように動作可能である、

40

バスエージェント。

【請求項 3 5】

請求項 3 4 に記載のバスエージェントであって、

前記属性チャンネルが、

初期書き込みアドレスを示す第 1 の信号を受け取ること、及び

書き込み長を示す第 2 の信号を受け取ること、

によって、書き込みアドレスの前記範囲を示す前記少なくとも 1 つの信号を受け取るように動作可能である、

50

バスエージェント。

【請求項 36】

請求項 31 に記載のバスエージェントであって、前記バストランザクションが第 1 のトランザクション識別を含む、バスエージェント。

【請求項 37】

請求項 36 に記載のバスエージェントであって、

前記バス及び前記読み出し / 書き込みメモリに接続されるデータチャネルを更に含み、前記データチャネルが、前記第 1 のデバイスにおける前記バストランザクションのサービスが完了すると、トランザクション応答を前記バス上で送信するように動作可能であり、

10

前記トランザクション応答が、前記第 1 のトランザクション識別に対応する第 2 のトランザクション識別を含む、

バスエージェント。

【請求項 38】

バス上で情報を交換するように動作可能なバスシステムであって、

第 1 のバスエージェント、及び

第 2 のバスエージェント、

を含み、

前記第 1 のバスエージェントが、中に複数のバスクレジットをストアするクレジットカウンタと、前記バスに及び前記クレジットカウンタに接続される第 1 の属性チャネルとを含み、

20

前記第 1 の属性チャネルが、

前記バス上でバストランザクションを送信すること、

前記クレジットカウンタにストアされたバスクレジットの前記数を前記バストランザクションに対応する量だけ減少させること、及び

対応するバスクレジット信号を前記バス上で送信すること、

によってバストランザクションを開始するように動作可能であり、

ストアされたバスクレジットの前記数が前記バストランザクションに対応する前記量より少ない場合、前記第 1 の属性チャネルが前記バストランザクションを送信しない可能性があり得、

30

前記第 1 の属性チャネルが、更に、

前記バスからクレジットリターンを受け取るように、及び前記クレジットカウンタにストアされたバスクレジットの前記数を前記クレジットリターンに対応する量だけ増加させるように動作可能であり、

前記第 2 のバスエージェントが、

前記バスに接続される第 2 の属性チャネルであって、前記バスからバストランザクション及びバスクレジット信号を受け取るように、及び前記バスクレジット信号が前記バストランザクションに対応する場合、前記バストランザクションをアクセプトするように動作可能である、前記第 2 の属性チャネルと、

前記第 2 の属性チャネルに接続され、前記第 2 の属性チャネルによってアクセプトされたバストランザクションをサービスするように動作可能であるバックエンドプロセスと

40

を含み、

前記第 2 の属性チャネルが、更に、前記バックエンドプロセスによる前記バストランザクションのサービスが完了すると、前記バス上でクレジットリターンを送信するように動作可能である、

バスシステム。

【請求項 39】

請求項 38 に記載のバスシステムであって、

前記バストランザクションが読み出しであり、前記第 1 の属性チャネルが、読み出しア

50

ドレスの範囲を示す少なくとも1つの信号を送信することによって、前記読み出しを前記バス上で送信するように動作可能であり、前記第2のエージェントが、更に、前記バスプロセスに結合される読み出し/書き込みメモリを含み、

前記第2の属性チャンネルが、読み出しアドレスの範囲を示す少なくとも1つの信号を含み、前記読み出しを前記バス上で受け取るように動作可能であり、前記バックエンドプロセスが、前記読み出し/書き込みメモリから、読み出しアドレスの前記範囲に対応するアドレスから、データをリコールすることによって、前記読み出しをサービスするように動作可能であり、

更に、前記バス及び前記読み出し/書き込みメモリに接続されるデータチャンネルを含み、前記データチャンネルが、前記読み出し/書き込みメモリからリコールされたデータを前記バス上で送信するように動作可能である、

10

バスシステム。

【請求項40】

請求項38に記載のバスシステムであって、

前記第1の属性チャンネルが、

初期読み出しアドレスを示す第1の信号を送信すること、及び

読み出し長を示す第2の信号を送信すること、

によって、読み出しアドレスの範囲を示す前記少なくとも1つの信号を送信するように動作可能であり、

前記第2の属性チャンネルが、

20

前記初期読み出しアドレスを示す前記第1の信号を受け取ること、及び

前記読み出し長を示す前記第2の信号を受け取ること、

によって、読み出しアドレスの範囲を示す前記少なくとも1つの信号を受け取るように動作可能である、

バスシステム。

【請求項41】

請求項38に記載のバスエージェントであって、

前記バストラザクションが書き込みであり、前記第1の属性チャンネルが、書き込みアドレスの範囲を示す少なくとも1つの信号を送信することによって、前記書き込みを前記バス上で送信するように動作可能であり、前記第1のバスエージェントが更に、書き込みの際、書き込みデータに対応する信号を送信するように動作可能である第1のデータチャンネルを含み、

30

前記第2のエージェントが更に、

前記バスプロセスに結合される読み出し/書き込みメモリと、

前記バスに接続されるデータチャンネルと、

を含み、前記データチャンネルが前記バスから書き込みデータを受け取るように動作可能であり、

前記第2の属性チャンネルが、書き込みアドレスの範囲を示す少なくとも1つの信号を受け取ることを含み、前記書き込みを前記バス上で受け取るように動作可能であり、及び

40

前記バックエンドプロセスが更に、前記読み出し/書き込みメモリに結合され、更に、前記バスからの書き込みデータを、書き込みアドレスの前記範囲に対応するアドレスにおいて、前記読み出し/書き込みメモリにストアすることによって、前記書き込みをサービスするように動作可能である、

バスエージェント。

【請求項42】

請求項41に記載のバスエージェントであって、前記第1の属性チャンネルが、

初期書き込みアドレスを示す第1の信号を送信すること、及び

書き込み長を示す第2の信号を送信すること、

によって、書き込みアドレスの範囲を示す前記少なくとも1つの信号を送信するように動作可能であり、

50

前記第 2 の属性チャンネルが更に、  
前記初期書き込みアドレスを示す前記第 1 の信号を受け取ること、及び  
前記書き込み長を示す前記第 2 の信号を受け取ること、  
によって、書き込みアドレスの前記範囲を示す前記少なくとも 1 つの信号を受け取るよ  
うに動作可能である、  
バスエージェント。

【請求項 4 3】

請求項 3 8 に記載のバスシステムであって、  
前記第 1 の属性チャンネルが更に、第 1 のトランザクション識別を送信することによって  
、前記バス上で前記バストランザクションを送信するように動作可能であり、  
前記第 2 のエージェントが更に、前記バス及び前記読み出し / 書き込みメモリに接続さ  
れるデータチャンネルを含み、  
前記データチャンネルが、前記バストランザクションをサービスすることが完了すると、  
トランザクション応答を前記バス上で送信するように動作可能であり、  
前記トランザクション応答が、前記第 1 のトランザクション識別に対応する第 2 のトラ  
ンザクション識別を含む、  
バスシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本願は、デジタルデータ処理に関し、特に複数の CPU コア間のデータバスに関する。

【背景技術】

【0002】

従来のバスプロトコルは、マスターとスレーブとの間でトランザクションを送信するた  
めに通信ハンドシェイクに依存する。典型的なハンドシェイクは下記の形式を取る。マス  
ターがバス上にトランザクションを置き、マスターがコマンド要求信号をアサートする。  
トランザクションは、トランザクションが受け取られたことを示すコマンド肯定応答信号  
をスレーブがアサートするまで、バス上に留まり、他のトランザクションが送信されるこ  
とを防ぐ。処理が完了した後、スレーブは、オプションとして、応答を別個のバス上に置  
き得、応答要求をアサートし得る。応答は、応答が受け取られたことを示す肯定応答信号  
をマスターがアサートするまで、バス上に留まり、他の応答が送信されることを防ぐ。

【0003】

この従来のバス通信ハンドシェイクは、次の要因によって性能損失があった。肯定応答  
信号を待って要求信号がアサートされるとき、コマンド又は応答等のトランザクションが  
複数のサイクルの間バス上に存在し得る。この待機は、他のトランザクションがバスを用  
いることを阻む。これにより、その時間期間の間に送信されるトランザクションの数が減  
少し、通信性能が低下する。このハンドシェイクは、トランザクションを送信するため  
に、トランザクション送信側からの要求信号、及びトランザクション受信側からの肯定応答  
信号の 2 方向通信を必要とする。これらの 2 方向信号は、典型的に異なるサイクルにあり  
、単一トランザクション処理の待ち時間を付加する。

【0004】

典型的な通信プロトコルは、読み出し、書き込み、及びコヒーレンストランザクション  
を別個の物理チャンネル上で送信する。コマンド及び応答もまた典型的に別個のチャンネル上  
で送信される。アドレス及びデータ幅が増加すると、その結果、物理設計中に配路されな  
ければならない物理的ワイヤの数が非常に膨大になる。その結果ワイヤが密集し、これは  
、面積の増加、電力消費の増大、及び設計スケジュールの長期化につながる。面積の増加  
が付加的なパイプラインステージの挿入に繋がる場合、この配線密集は性能低下も引き起  
こす恐れがある。

【発明の概要】

【0005】

10

20

30

40

50

説明される例のバス通信プロトコルにおいて、マスターデバイスが通信を開始し得る。各マスターデバイスは複数のバスクレジットをストアする。マスターデバイスは、バストランザクションに対応する数及びタイプのバスクレジットをストアしている場合にのみ、そのバストランザクションを送信し得る。マスターデバイスが十分なバスクレジットを持つ場合、マスターデバイスは、バストランザクション及びクレジット信号を送信し得る。送信の際、マスターデバイスは、ストアされたバスクレジットの数を、バストランザクションに対応する量、減少させる。

**【 0 0 0 6 】**

バスクレジットは、バストランザクションを受け取るスレーブデバイス上のリソースに対応する。マスターデバイスが適切なクレジットを持つ場合、スレーブデバイスは、バストランザクションを受け取る容量を有する。このように、スレーブデバイスは、適切なクレジットが付随している場合、バストランザクションを受け取らなければならない。

10

**【 0 0 0 7 】**

バストランザクションをアクセプトした後、スレーブデバイスはトランザクションをサービスする。そのようなサービスには、コマンドを実行すること、読み出しに 응답してデータをリコールすること、又は書き込みに 응답してデータをストアすることが含まれ得る。バストランザクションをサービスし、スレーブデバイスが別のトランザクションをアクセプトし得る状態を再開した後、スレーブデバイスは、クレジットリターンをバス上に送信する。マスターデバイスがクレジットリターンを受け取ると、マスターデバイスは、対応する数及びタイプのクレジットをストアされている量に加算する。クレジットリターンの送信に続いて、スレーブデバイスは、別のバストランザクションをアクセプトする準備が整う。クレジットリターンを受け取ると、マスターデバイスは、バストランザクションを開始することが再度可能になる。

20

**【 0 0 0 8 】**

多くのタイプの相互作用において、バスエージェントは、プロセスの状態に応じてマスターとスレーブの両方として機能し得る。通常読み出し動作では、第1のバスエージェントが、マスターデバイスとして機能する間に、読み出しを送信し、読み出しアドレスを指示する。第2のバスエージェントが、スレーブデバイスとして機能する間に、読み出しを受け取り、アクセプトする。第2のバスエージェントは、スレーブデバイスとして機能する間に、メモリにアクセスすること、及び特定されたデータをリコールすることによって、読み出しをサービスする。好ましい実施形態において、第2のバスエージェントは、独立したバストランザクションによって、リコールされたデータを第1のエージェントに送信する。この例において、第2のバスエージェントは、第1のバスエージェントへのデータ送信をサポートするための数とタイプのバスクレジットをストアしなければならない。第2のエージェントが適正なクレジットをストアする場合、第1のバスエージェントは、バストランザクションを受け取り、サービスする容量を有する。この例において、第2のバスエージェントは、十分なバスクレジットを持ち、読み出されたデータのデータ転送を開始する。第1のバスエージェントは、バストランザクション及び読み出されたデータをアクセプトする。第1のバスエージェントは、読み出されたデータをストアすることによって、バストランザクションをサービスする。バストランザクションをサービスし、異なるバストランザクションを受け取るためにリソースをクリアにすると、第1のバスエージェントは、対応するクレジットリターンを第2のバスエージェントに送信する。第2のバスエージェントは、その後、ストアされたクレジットを増加させ、第1のバスエージェントが再びバストランザクションを受け取ることが可能であることを確認する。

30

40

**【 0 0 0 9 】**

好ましい実施形態において、複数のそのようなバス交換がオーバーラップし得る。そのようなオーバーラップするバストランザクションを分離させておくために、第1のバスエージェントは、好ましくは、第1のトランザクション識別をトランザクションと共に送信する。第2のバスエージェントは、バストランザクション応答において第1のトランザクション識別に対応する第2のトランザクション識別を送信する。これによって、第1のバ

50

スエージェントは応答を区別することが可能になる。

【図面の簡単な説明】

【0010】

【図1】一実施形態に従ったシングルコアスカラープロセッサを図示する。

【0011】

【図2】別の実施形態に従ったデュアルコアスカラープロセッサを図示する。

【0012】

【図3】更なる実施形態に従ったシングルコアベクトルプロセッサを図示する。

【0013】

【図4】更なる実施形態に従ったデュアルコアベクトルプロセッサを図示する。

【0014】

【図5】CPUの一実施形態の構成を図示する。

【0015】

【図6】グローバルスカラーレジスタファイルを図示する。

【0016】

【図7】グローバルベクトルレジスタファイルを図示する。

【0017】

【図8】乗算及び相関機能ユニットによって共有されるローカルベクトルレジスタファイルを図示する。

【0018】

【図9】ロード/ストアユニットのローカルレジスタファイルを図示する。

【0019】

【図10】プレディケートレジスタファイルを図示する。

【0020】

【図11】好ましい実施形態に従った、中央処理装置のパイプラインフェーズを図示する。

【0021】

【図12】シングルフェッチパケットの16個の命令を図示する。

【0022】

【図13】例示の実施形態によって用いられる命令の命令コーディングの例を図示する。

【0023】

【図14】例示の実施形態に従った、SIMD演算のためのキャリー制御を図示する。

【0024】

【図15】本発明を適用し得るコンピュータクラスタ1500を図示する。

【0025】

【図16】シングルマルチコアバスアーキテクチャインタフェースのための各チャネルに対する信号方向を図示する。

【0026】

【図17】例示のペアにされたインタフェースポートロジックにおける各チャネルを図示する。

【0027】

【図18】別の例示のペアにされたインタフェースポートロジックにおける各チャネルを図示する。

【0028】

【図19】dedc信号のエンコーディングのためのロジックを図示する。

【0029】

【図20】トランザクション属性チャネルのためのシンプルなクレジットハンドシェイクを図示する。

【0030】

【図21】同じチャネル上のインターリーブされたクレジットハンドシェイクを示す。

10

20

30

40

50

【 0 0 3 1 】

【 図 2 2 】 クレジット非書き込みシーケンスにおけるステップを図示する。

【 0 0 3 2 】

【 図 2 3 】 クレジットされた ( credited ) 書き込みコマンドシーケンスの一般的な形式を図示する。

【 0 0 3 3 】

【 図 2 4 】 例示の実施形態において用いられるようなエージェントのブロック図を示す。

【 発明を実施するための形態 】

【 0 0 3 4 】

図 1 は、一実施形態に従ったシングルコアスカラープロセッサを図示する。シングルコアプロセッサ 1 0 0 はスカラー中央処理装置 ( CPU ) 1 1 0 を含み、スカラー CPU 1 1 0 は、別個のレベル 1 命令キャッシュ ( L 1 I ) 1 1 1 及びレベル 1 データキャッシュ ( L 1 D ) 1 1 2 に結合される。中央処理装置コア 1 1 0 は、既知のように構成され得、典型的に、レジスタファイル、整数算術論理ユニット、整数積算器、及びプログラムフロー制御ユニットを含み得る。シングルコアプロセッサ 1 0 0 は、命令とデータの両方を持つレベル 2 組み合わせ命令 / データキャッシュ ( L 2 ) 1 1 3 を含む。好ましい実施形態において、スカラー中央処理装置 ( CPU ) 1 1 0、レベル 1 命令キャッシュ ( L 1 I ) 1 1 1、レベル 1 データキャッシュ ( L 1 D ) 1 1 2、及びレベル 2 組み合わせ命令 / データキャッシュ ( L 2 ) 1 1 3 は、単一の集積回路上に形成される。

10

【 0 0 3 5 】

好ましい実施形態において、この単一の集積回路はまた、電力制御回路 1 2 1 等の補助回路、エミュレーション / トレース回路 1 2 2、D S T ( design for test ) P B I S T ( programmable built-in self-test ) 回路 1 2 3、及びクロッキング回路 1 2 4 を含む。メモリコントローラ 1 3 1 が、CPU 1 1 0 の外部として、単一集積回路 1 0 0 上に集積され得る。

20

【 0 0 3 6 】

CPU 1 1 0 は、定義されたデータに対してデータ処理演算を実施するように、プログラム制御下で動作する。CPU 1 1 0 を制御するプログラムは、デコーディング及び実行の前にフェッチされなければならない複数の命令で構成される。シングルコアプロセッサ 1 0 0 は複数のキャッシュメモリを含む。図 1 は、第 1 のレベルキャッシュのペアを図示する。レベル 1 命令キャッシュ ( L 1 I ) 1 1 1 は、CPU 1 1 0 によって用いられる命令をストアする。CPU 1 1 0 は、最初に、レベル 1 命令キャッシュ 1 2 1 から任意の命令にアクセスすることを試みる。レベル 1 データキャッシュ ( L 1 D ) 1 1 2 は、CPU 1 1 0 によって用いられるデータをストアする。CPU 1 1 0 は、最初に、レベル 1 データキャッシュ 1 1 2 から任意の必要とされるデータにアクセスすることを試みる。2 つのレベル 1 キャッシュ ( L 1 I 1 1 1、及び L 1 D 1 1 2 ) は、レベル 2 統合キャッシュ ( L 2 ) 1 1 3 によってバックアップされている。レベル 1 命令キャッシュ 1 1 1 又はレベル 1 データキャッシュ 1 1 2 に対するキャッシュミスがあった場合、要求された命令又はデータが、レベル 2 統合キャッシュ 1 1 3 から探される。要求された命令又はデータがレベル 2 統合キャッシュ 1 1 3 にストアされている場合は、その命令又はデータは、中央処理装置コア 1 1 0 に供給するために、要求しているレベル 1 キャッシュに供給される。当業者には周知のように、要求された命令又はデータは、使用を迅速にするために、要求しているキャッシュと CPU 1 1 0 の両方に同時に供給され得る。

30

40

【 0 0 3 7 】

レベル 2 統合キャッシュ 1 1 3 は、更に、メモリコントローラ 1 3 1 を介して、より高いレベルのメモリシステムに結合される。メモリコントローラ 1 3 1 は、外部メモリ ( 図 1 に図示されない ) にアクセスすることによって、レベル 2 統合キャッシュ 1 1 3 におけるキャッシュミスを取り扱う。メモリコントローラ 1 3 1 は、キャッシュアビリティ判定、エラー検出及び訂正、アドレス翻訳等など、全てのメモリ中心の機能を取り扱う。シングルコアプロセッサ 1 0 0 は、マルチプロセッサシステムの一部であり得る。その場合、メ

50

メモリコントローラ 131 は、プロセッサ間のデータ転送を取り扱い、プロセッサ間のキャッシュコヒーレンスを維持する。

【0038】

図2は、別の実施形態に従ったデュアルコアプロセッサを図示する。デュアルコアプロセッサ200は、別個のレベル1命令キャッシュ(L1I)211及びレベル1データキャッシュ(L1D)212に結合される第1のCPU210、及び別個のレベル1命令キャッシュ(L1I)221及びレベル1データキャッシュ(L1D)212に結合される第2のCPU220を含む。中央処理装置210及び220は、好ましくは、図1に図示されるCPU110と同様に構成される。デュアルコアプロセッサ200は、4個のレベル1キャッシュ(L1I 211、L1D 212、L1I 221、及びL1D 222)全てをサポートする、単一の共有されたレベル2組み合わせ命令/データキャッシュ(L2)231を含む。好ましい実施形態において、CPU210、レベル1命令キャッシュ(L1I)211、レベル1データキャッシュ(L1D)212、CPU220、レベル1命令キャッシュ(L1I)221、レベル1データキャッシュ(L1D)222、及びレベル2組み合わせ命令/データキャッシュ(L2)231が、単一の集積回路上に形成される。この単一の集積回路は、好ましくは、電力制御回路245等の補助回路、エミュレーション/トレース回路116、DST (design for test) PBIST (programmable built-in self-test) 回路117、及びクロッキング回路118を含む。この単一の集積回路はまたメモリコントローラ251を含み得る。

10

【0039】

図3及び図4は、それぞれ図1及び図2に示されるものと類似のシングルコアプロセッサ及びデュアルコアプロセッサを図示する。図3及び図4は、ベクトル中央処理装置が示されている点で、図1及び図2とは異なる。下記に更に詳細を説明するように、シングルコアベクトルプロセッサ300はベクトルCPU310を含む。デュアルコアベクトルプロセッサ400は、2つのベクトルCPU410及び420を含む。ベクトルCPU310、410、及び420は、対応するスカラーCPU110、210、及び220に比較して、一層幅広いデータパスオペレーショナルユニット、及び一層幅広いデータレジスタを含む。

20

【0040】

ベクトルCPU310、410、及び420は更に、ストリーミングエンジン313(図3)、及びストリーミングエンジン413及び423(図5)を含むという点で、対応するスカラーCPU110、210、及び220とは異なる。ストリーミングエンジン313、413、及び423は類似である。ストリーミングエンジン313は、データをレベル2統合キャッシュ313(L2)からベクトルCPU310に転送する。ストリーミングエンジン413は、データをレベル2統合キャッシュ431からベクトルCPU410に転送する。ストリーミングエンジン423は、データをレベル2統合キャッシュ431からベクトルCPU420に転送する。好ましい実施形態に従って、各ストリーミングエンジン313、413、及び423は、最大2つのデータストリームを管理する。

30

【0041】

各ストリーミングエンジン313、413、及び423は、一定の制約された状況においてデータを転送する。ストリームは特定のタイプのエレメントのシーケンスで構成される。ストリーム上で動作するプログラムは、データを順次読み出し、各エレメント上で順に動作する。ストリームは全て、下記の基本的な特性を有する。ストリームデータは、適切に定義された、時間における開始と終了を有する。ストリームデータは、ストリームにわたって固定のエレメントサイズ及びタイプを有する。ストリームデータは、エレメントの固定のシーケンスを有する。このように、プログラムはストリーム内をランダムに探すことはできない。ストリームデータは、アクティブの間は読み出し専用である。プログラムは、ストリームから読み出している間は、同時にストリームに書き込みできない。一旦、ストリームが開始されると、ストリーミングエンジンは、アドレスを計算し、定義されたデータタイプをレベル2統合キャッシュからフェッチし、ゼロ拡張、符号拡張等のデー

40

50

タイプマニピュレーション、マトリックス転位等のデータエレメントソーティング/スワッピングを実施し、データをCPU内のプログラムされた実行ユニットに直接的に送達する。ストリーミングエンジンは、このように、正常に動作するデータ上でのリアルタイムのデジタルフィルタリング演算のために有用である。ストリーミングエンジンは、対応するCPUからこれらのメモリフェッチタスクを解放して、他の処理機能を可能にする。

#### 【0042】

ストリーミングエンジンは下記利点を提供する。多次元メモリアクセスを可能にする。機能ユニットに対して利用可能な帯域幅を増大させる。ストリームバッファがL1Dキャッシュ及びL2キャッシュをバイパスし得るので、キャッシュミスストール(詰まり)の数を最小化する。ループにおいて維持する必要のあるスカラー演算の数を低減する。アドレスポインタを管理する。他の計算のためのアドレス生成命令スロット及び、Dユニットを自動的に解放するアドレス生成を取り扱う。

10

#### 【0043】

図5は、CPUの一実施形態の構成を図示する。特に注釈がない場合は、本明細書は、スカラーCPU及びベクトルCPUの両方に適用される。CPUは、複数の実行ユニット、乗算ユニット511(.M)、相関ユニット512(.C)、算術ユニット513(.L)、算術ユニット514(.S)、ロード/ストアユニット515(.D)、分岐ユニット516(.B)、及びプレディケーションユニット517(.P)を含む。これらの実行ユニットの動作及び関係を以下に詳細に説明する。

#### 【0044】

乗算ユニット511は、主として乗算を実施する。乗算ユニット511は、最大2個のダブルベクトルオペランドをアクセプトし、最大1個のダブルベクトル結果を生成する。乗算ユニット511は、8ビットから64ビットの乗算演算の精度を備える種々の整数乗算演算、種々の正規及び複素ドット積演算、及び種々の浮動小数点乗算演算、ビットワイズ論理演算、ムーブ、並びに、加算及び減算を行うように構成可能な命令である。図5に図示されるように、乗算ユニット511は、4個の同時16ビット×16ビットの乗算のためのハードウェアを含む。乗算ユニット511は、下記に説明する様式で、グローバルスカラーレジスタファイル521、グローバルベクトルレジスタファイル522、及び共有の.M及びC.ローカルレジスタ523ファイルにアクセスし得る。フォワーディングマルチプレクサ530は、グローバルスカラーレジスタファイル521、グローバルベクトルレジスタファイル522、対応するストリーミングエンジン、及び乗算ユニット511の間のデータ転送を調停する。

20

30

#### 【0045】

相関ユニット512(.C)は、最大2つのダブルベクトルオペランドをアクセプトし、最大1つのダブルベクトル結果を生成する。相関ユニット512は、これらの主要演算をサポートする。WCDMA「Rake」及び「Search」命令のサポートにおいて、相関ユニット512は、クロックサイクル毎に、最大512個の2ビットPN×8ビットI/Q複素乗算を実施する。相関ユニット512は、クロックサイクル毎に、最大512個のSAD(Sum-of-Absolute-Difference)を実施する、8ビット及び16ビットのSAD計算を実施する。相関ユニット512は、水平加算及び水平最小/最大命令を実施する。相関ユニット512は、ベクトルパーミュート(permutates)命令を実施する。相関ユニット512は、8個の256ビット幅の制御レジスタを含む。これらの制御レジスタは、或る相関ユニット命令の演算を制御するように用いられる。相関ユニット512は、下記に説明する様式で、グローバルスカラーレジスタファイル521、グローバルベクトルレジスタファイル522、及び共有.M及びC.ローカルレジスタファイル523にアクセスし得る。フォワーディングマルチプレクサ530は、グローバルスカラーレジスタファイル521、グローバルベクトルレジスタファイル522、対応するストリーミングエンジン、及び相関ユニット512の間のデータ転送を調停する。

40

#### 【0046】

CPU500は、算術ユニット513(.L)及び算術ユニット514(.S)の2つ

50

の算術ユニットを含む。算術ユニット 5 1 3 及び算術ユニット 5 1 4 の各々は、最大 2 つのベクトルオペランドをアクセプトし、1 つのベクトル結果を生成する。コンピュータユニットは、これらの主要演算をサポートする。算術ユニット 5 1 3 及び算術ユニット 5 1 4 は、8 ビットから 6 4 ビットまでわたる精度で種々の S I M D (single-instruction-multiple-data) 固定小数点算術演算を実施する。算術ユニット 5 1 3 及び算術ユニット 5 1 4 は、結果をプレディケートレジスタファイル 5 2 6 (下記に詳細を説明する) に直接書き込む、種々の、ベクトル比較及び最小 / 最大命令を実施する。これらの比較には、 $A = B$ 、 $A > B$ 、 $A \leq B$ 、 $A < B$ 、及び  $A \neq B$  が含まれる。比較が真であると、プレディケートレジスタ内の対応するビット位置に 1 ビットがストアされる。比較が偽であると、プレディケートレジスタ内の対応するビット位置に 0 がストアされる。ベクトル比較命令は、バイト (8 ビット) データを前提とし、従って、3 2 個のシングルビット結果を生成する。算術ユニット 5 1 3 及び算術ユニット 5 1 4 は、下記に説明されるように、指定されたプレディケートレジスタを用いて種々のベクトル演算を実施する。算術ユニット 5 1 3 及び算術ユニット 5 1 4 は、半精度 (1 6 ビット)、単精度 (3 2 ビット) から倍精度 (6 4 ビット) までわたる精度で、種々の S I M D 浮動小数点算術演算を実施する。算術ユニット 5 1 3 及び算術ユニット 5 1 4 は、種々のアルゴリズム及び関数を迅速化するための特殊な命令を実施する。算術ユニット 5 1 3 及び算術ユニット 5 1 4 は、グローバルスカラーレジスタファイル 5 2 1、グローバルベクトルレジスタファイル 5 2 2、共有 . L 及び . S ローカルレジスタファイル 5 2 4、及びプレディケートレジスタファイル 5 2 6 にアクセスし得る。フォワーディングマルチプレクサ 5 3 0 は、グローバルスカラーレジスタファイル 5 2 1、グローバルベクトルレジスタファイル 5 2 2、対応するストリーミングエンジン、及び算術ユニット 5 1 3、5 1 4 の間のデータ転送を調停する。

#### 【 0 0 4 7 】

ロード / ストアユニット 5 1 5 ( . D ) は、主としてアドレス計算に用いられる。ロード / ストアユニット 5 1 5 は、最大 6 4 ビットのスカラーオペランドをアクセプトするように拡張され、最大 6 4 ビットのスカラー結果を生成する。ロード / ストアユニット 5 1 5 は、他のユニット上のワークロードを低減するように、スワッピング、ロード及びストアデータ上でのパック、アンパック等のデータマニピュレーションを実施するための付加的ハードウェアを含む。ロード / ストアユニット 5 1 5 は、各クロックサイクルに 1 つのロード又はストア要求を、4 4 ビットの物理アドレスと共に、レベル 1 データキャッシュ ( L 1 D ) に送り出し得る。ロード又はストアデータ幅は、3 2 ビット、6 4 ビット、2 5 6 ビット、又は 5 1 2 ビットであり得る。ロード / ストアユニット 5 1 5 は、6 4 ビット S I M D 算術演算、6 4 ビットビットワイズ論理演算、及びスカラー及びベクトルロード及びストアデータマニピュレーション、の主要演算をサポートする。ロード / ストアユニット 5 1 5 は、好ましくは、4 8 ビット仮想アドレスから 4 4 ビット物理アドレスへのアドレス翻訳を実施するためのマイクロ T L B (table look-aside buffer) ブロックを含む。ロード / ストアユニット 5 1 5 は、後述する様式で、グローバルスカラーレジスタファイル 5 2 1、グローバルベクトルレジスタファイル 5 2 2、及び . D ローカルレジスタファイル 5 2 5 にアクセスし得る。フォワーディングマルチプレクサ 5 3 0 は、グローバルスカラーレジスタファイル 5 2 1、グローバルベクトルレジスタファイル 5 2 2、対応するストリーミングエンジン、及びロード / ストアユニット 5 1 5 の間のデータ転送を調停する。

#### 【 0 0 4 8 】

分岐ユニット 5 1 6 ( . B ) は、分岐アドレスを計算し、分岐予測を実施し、予測の結果に応じて制御フローを変更する。

#### 【 0 0 4 9 】

プレディケーションユニット 5 1 7 ( . P ) は、ベクトルプレディケーションレジスタ上で基本演算を実施する小型制御ユニットである。プレディケーションユニット 5 1 7 は、ベクトルプレディケーションレジスタ 5 2 6 への直接アクセスを有する。プレディケーションユニット 5 1 7 は、AND、ANDN、OR、XOR、NOR、BITR、NEG

、SET、BITCNT（ビットカウント）、RMBD（一番右のビット検出）、BIT Decimate and Expand等のプレディケーションレジスタ上で異なるビット演算を実施する。

#### 【0050】

図6は、グローバルスカラーレジスタファイル521を図示する。16個の独立した64ビット幅のスカラーレジスタがある。グローバルスカラーレジスタファイル521の各レジスタは、32ビットスカラーデータ（レジスタA0～A15 601と称する）又は64ビットのスカラーデータ（レジスタEA0～EA15 611と称する）として読み出され得る。しかしながら、書き込みは常に64ビットであり、必要に応じ、最大64ビットになるようにゼロ拡張される。全ての機能ユニットの全てのスカラー命令が、グローバルスカラーレジスタファイル521に対して読み出し又は書き込みし得る。命令タイプはデータサイズを決定する。グローバルスカラーレジスタファイル521は、サイズが8ビットから64ビットまでわたるデータタイプをサポートする。また、ベクトル命令が、ベクトルの上位192ビットデータが無視されて、64ビットのグローバルスカラーレジスタ521に書き込み得る。ベクトル命令はまた、グローバルスカラーレジスタファイル511から64ビットデータを読み出し得る。この場合、オペランドは、入力ベクトルを形成するために、上位192ビットにおいてゼロ拡張される。

10

#### 【0051】

図7は、グローバルベクトルレジスタファイル522を図示する。16個の独立した256ビット幅のベクトルレジスタがある。グローバルベクトルレジスタファイル522の各レジスタは、32ビットスカラーデータ（レジスタX0～X15 701と称する）、64ビットのスカラーデータ（レジスタEX0～EX15 711と称する）、256ビットベクトルデータ（レジスタVX0～VX15 721と称する）、又は512ビットのダブルベクトルデータ（DVX0～DVX12と称するが、図示せず）として読み出され得る。この実施形態において、乗算ユニット511及び相関ユニット512のみがダブルベクトル命令を実行し得る。全ての機能ユニットの全てのベクトル命令が、グローバルベクトルレジスタファイル522に対して読み出し又は書き込みし得る。また、任意の機能ユニットの任意のスカラー命令が、読み出し又は書き込みのために、グローバルベクトルレジスタファイル522レジスタの下位32又は64ビットにアクセスし得る。命令タイプはデータサイズを決定する。

20

30

#### 【0052】

図8は、ローカルベクトルレジスタファイル523を図示する。16個の独立した256ビット幅のベクトルレジスタがある。ローカルベクトルレジスタファイル523の各レジスタは、32ビットスカラーデータ（レジスタM0～M15 701と称する）、64ビットのスカラーデータ（レジスタEM0～EM15 711と称する）、256ビットベクトルデータ（レジスタVM0～VM15 721と称する）、又は512ビットダブルベクトルデータ（DVM0～DVM7と称するが、図示せず）として読み出され得る。この実施形態において、乗算ユニット511及び相関ユニット512のみがダブルベクトル命令を実行し得る。全ての機能ユニットの全てのベクトル命令が、ローカルベクトルレジスタファイル523に対して書き込みし得る。乗算ユニット511及び相関ユニット512の命令のみが、ローカルベクトルレジスタファイル523から読み出しし得る。命令タイプはデータサイズを決定する。

40

#### 【0053】

乗算ユニット511は、ダブルベクトル（512ビットデータ）上で動作し得る。乗算ユニット511は、グローバルベクトルレジスタファイル521及びローカルベクトルレジスタファイル523から、ダブルベクトルデータを読み出し、グローバルベクトルレジスタファイル521及びローカルベクトルレジスタファイル523にダブルベクトルデータを書き込み得る。レジスタ指定DVXx及びDVMxは、下記のように、グローバルベクトルレジスタファイル521及びローカルベクトルレジスタファイル523にマッピングされる。

50

【表 1】

Table 1

Instruction Designation 命令指定	Register Accessed アクセスされるレジスタ
DVX0	VX1:VX0
DVX1	VX3:VX2
DVX2	VX5:VX4
DVX3	VX7:VX6
DVX4	VX9:VX8
DVX5	VX11:VX10
DVX6	VX13:VX12
DVX7	VX15:VX14
DVM0	VM1:VM0
DVM1	VM3:VM2
DVM2	VM5:VM4
DVM3	VM7:VM6
DVM4	VM9:VM8
DVM5	VM11:VM10
DVM6	VM13:VM12
DVM7	VM15:VM14

10

20

各ダブルベクトル指定は、グローバルベクトルレジスタ 5 2 2 か又はローカルベクトルレジスタ 5 2 3 における、対応する近隣ベクトルレジスタのペアにマッピングする。指定 DVX0 ~ DVX7 は、グローバルベクトルレジスタ 5 2 2 にマッピングする。指定 DVM0 ~ DVM7 は、ローカルベクトルレジスタ 5 2 3 にマッピングする。

## 【0054】

30

ローカルベクトルレジスタファイル 5 2 4 は、ローカルベクトルレジスタファイル 5 2 3 に類似する。16 個の独立した 256 ビット幅のベクトルレジスタがある。ローカルベクトルレジスタファイル 5 2 4 の各レジスタは、32 ビットスカラーデータ（レジスタ L0 ~ L15 701 と称する）、64 ビットのスカラーデータ（レジスタ EL0 ~ EL15 711 と称する）、又は 256 ビットベクトルデータ（レジスタ VL0 ~ VL15 721 と称する）として読み出され得る。全ての機能ユニットの全てのベクトル命令が、ローカルベクトルレジスタファイル 5 2 4 に対して書き込みし得る。算術ユニット 5 1 3 及び算術ユニット 5 1 4 の命令のみが、ローカルベクトルレジスタファイル 5 2 4 から読み出しし得る。

## 【0055】

40

図 9 は、ローカルレジスタファイル 5 2 5 を図示する。16 個の独立した 64 ビット幅のレジスタがある。ローカルレジスタファイル 5 2 5 の各レジスタは、32 ビットスカラーデータ（レジスタ D0 ~ D15 701 と称する）、又は 64 ビットのスカラーデータ（レジスタ ED0 ~ ED15 711 と称する）として読み出され得る。全ての機能ユニットの全てのスカラー及びベクトル命令が、ローカルレジスタファイル 5 2 5 に対して書き込みし得る。ロード/ストアユニット 5 1 5 の命令のみが、ローカルレジスタファイル 5 2 5 から読み出しし得る。また、任意のベクトル命令が、結果のベクトルの上位 192 ビットデータが無視されて、ローカルレジスタファイル 5 2 5 に 64 ビットデータを書き込むことができる。任意のベクトル命令はまた、64 ビットローカルレジスタファイル 5 2 5 レジスタから 64 ビットデータを読み出すことができる。リターンデータは、入力ベ

50

クトルを形成するために、上位 192 ビットにおいてゼロ拡張される。ローカルレジスタファイル 525 のレジスタは、ロード/ストアユニット 515 の 64 ビット算術論理命令のためのストアデータとして又はソースとしてではなく、ロード/ストア命令におけるアドレスとしてのみ用いられ得る。

#### 【0056】

図 10 は、プレディケートレジスタファイル 517 を図示する。プレディケートレジスタファイル 517 に 16 個のレジスタ 32 ビットレジスタがある。プレディケートレジスタファイル 517 は、何れかの算術によって実行されるベクトル比較演算の結果を含み、ベクトル選択命令及びベクトルプレディケートストア命令によって用いられる。また、特殊命令のスマールサブセットが、プレディケートレジスタから直接的に読み出し、演算を実施し、プレディケートレジスタへ直接的にライトバックし得る。また、グローバルレジスタファイル (521 及び 522) とプレディケートレジスタファイル 517 との間で値を転送できる命令がある。プレディケートレジスタファイル 517 とローカルレジスタファイル (523、524、及び 525) との間の転送はサポートされていない。プレディケーションレジスタ (P0 ~ P15 と称する) の各ビットは、ベクトルデータのバイトを制御する。ベクトルが 256 ビットであるので、プレディケートレジスタの幅は、 $256 / 8 = 32$  ビットに等しい。プレディケートレジスタファイルは、ベクトル比較の結果をストアするために、ベクトル比較演算によって書き込まれ得る。

#### 【0057】

CPU 110、210、220、310、410、又は 420 等の CPU は、命令パイプライン上で動作し得る。この命令パイプラインは、サイクル毎に、命令を 7 個の実行ユニット (乗算ユニット 511、相関ユニット 512、算術ユニット 513、算術ユニット 514、ロード/ストアユニット 515、分岐ユニット 516、及びプレディケーションユニット 517) に提供するように、最大 9 個の並列 32 ビットスロットをディスパッチし得る。命令は、更に後述するように、バックされた固定長のフェッチされた命令である。全ての命令は、フェッチ及びデコードのために同数のパイプラインフェーズを必要とするが、様々な数の実行フェーズを必要とする。

#### 【0058】

図 11 は、プログラムフェッチフェーズ 1110、ディスパッチ及びデコードフェーズ 1110、及び実行フェーズ 1130 のパイプラインフェーズを図示する。プログラムフェッチフェーズ 1110 は、全ての命令に対して 3 つのステージを含む。ディスパッチ及びデコードフェーズは、全ての命令に対して 3 つのステージを含む。実行フェーズ 1130 は、命令に応じて 1 から 4 個のステージを含む。

#### 【0059】

フェッチフェーズ 1110 は、プログラムアドレス生成ステージ 1111 (PG)、プログラムアクセスステージ 1112 (PA)、及びプログラムレシーブステージ 1113 (PR) を含む。プログラムアドレス生成ステージ 1111 (PG) の間、プログラムアドレスが CPU において生成され、読み出し要求がレベル 1 命令キャッシュ L1I のためのメモリコントローラに送信される。プログラムアクセスステージ 1112 (PA) の間、レベル 1 命令キャッシュ L1I は、要求を処理し、そのメモリ内のデータにアクセスし、CPU 境界にフェッチパケットを送信する。プログラムレシーブステージ 1113 (PR) の間、CPU はフェッチパケットを登録する。

#### 【0060】

命令は、常に、一度にフェッチされた 16 語である。図 12 は、このフェッチパケットを図示する。図 12 は、シングルフェッチパケットの 16 個の命令 1201 ~ 1216 を図示する。フェッチパケットは、512 ビット (16 ワード) 境界上で整合される。個々の命令の実行は、各命令における p ビットによって部分的に制御される。p ビットは、好ましくは、命令のビット 0 である。p ビットは、命令が別の命令と並列に実行するか否かを決定する。p ビットは、下位から上位アドレスへスキャンされる。命令の p ビットが 1 の場合、次に続く命令が、その命令 I と並列に (同じサイクルにおいて) 実行される。命令

10

20

30

40

50

の p ビットが 0 である場合、次に続く命令は、その命令の後のサイクルで実行される。並列に実行する全ての命令が、実行パケットを構成する。実行パケットは、最大 9 個の命令を含み得る。実行パケットにおける各命令は、異なる機能ユニットを使用しなければならない。実行パケットは、最大 9 個の 32 ビット幅スロットを含み得る。スロットは、自己自足の (self-contained) 命令であり得るか、又は直前の命令によって特定された一定フィールドを拡張し得る。スロットは、同じフェッチパケット内の命令に適用するように、条件付きコードとして用いられ得る。フェッチパケットは、最大 2 個の一定拡張スロット、及び 1 つの条件コード拡張スロットを含み得る。

【 0 0 6 1 】

最大 11 個の個別の命令スロットがあるが、スケジューリング制約によって並列スロット最大数である 9 個に制限される。最大 9 個のスロットは、下記のように共有される。

【 0 0 6 2 】

乗算ユニット 5 1 1、相関ユニット 5 1 2、算術ユニット 5 1 3、算術ユニット 5 1 4、ロード/ストアユニット 5 1 5、プレディケートユニット 5 1 7 と共有される分岐ユニット 5 1 6、第 1 の一定拡張、第 2 の一定拡張、及び条件コード拡張と共有されるユニットレス命令。実行パケットにおける最後の命令は、0 に等しい p ビットを有する。

【 0 0 6 3 】

CPU 及びレベル 1 命令キャッシュ L1I パイプラインは、互いから結合解除され得る。レベル 1 命令キャッシュ L1I からのフェッチパケットリターンは、レベル 1 命令キャッシュ L1I においてヒットがあるか否か等の外部状況に応じて、異なる数のクロックサイクルを取り得る。従って、プログラムアクセスステージ 1 1 1 2 (PA) は、他のステージにおけるように 1 クロックではなく、幾つかのクロックサイクルを取り得る。

【 0 0 6 4 】

ディスパッチ及びデコードフェーズ 1 1 1 0 は、適切な実行ユニットステージ 1 1 2 1 (DS)、命令プリデコードステージ 1 1 2 2 (D1)、及び命令デコード、オペランド読み出しステージ 1 2 2 2 (D2) に対する命令ディスパッチを含む。適切な実行ユニットステージ 1 1 2 1 (DS) に対する命令ディスパッチの間、フェッチパケットは、実行パケットに分けられ、適切な機能ユニットにアサインされる。命令プリデコードステージ 1 1 2 2 (D1) の間、ソースレジスタ、宛先レジスタ、及び関連するパスは、機能ユニットにおける命令の実行のためにデコードされる。命令デコードの間、オペランド読み出しステージ 1 2 2 2 (D2) では、レジスタファイルからのオペランド読み出しと共に、より詳細なユニットデコードが行われる。

【 0 0 6 5 】

実行フェーズ 1 1 3 0 は、実行ステージ 1 1 3 1 ~ 1 1 3 5 (E1 ~ E5) を含む。異なるタイプの命令は、それらの実行を完了させるために異なる数のこれらのステージを必要とする。パイプラインのこれらのステージは、CPU サイクル境界でのデバイス状態の理解において重要な役割を果たす。

【 0 0 6 6 】

実行 1 ステージ 1 1 3 1 (E1) の間、命令に対する条件が評価され、オペランドが動作される。図 1 1 に図示されるように、実行 1 ステージ 1 1 3 1 は、ストリームバッファ 1 1 4 1 から、及び概略的に 1 1 4 2 として示されるレジスタファイルの 1 つから、オペランドを受け取り得る。ロード及びストア命令では、アドレス生成が実施され、アドレス変更がレジスタファイルに書き込まれる。分岐命令では、PG フェーズにおける分岐フェッチパケットが影響される。図 1 1 に図示されるように、ロード及びストア命令は、本明細書では概略的にメモリ 1 1 5 1 として示されるメモリにアクセスする。単一サイクル命令では、宛先レジスタファイルに結果が書き込まれる。これは、命令に対する任意の条件が真として評価されると仮定している。条件が偽として評価される場合、命令は、如何なる結果も書き込むことはなく、又は、実行 1 ステージ 1 1 3 1 の後に任意のパイプライン演算を有する。

【 0 0 6 7 】

10

20

30

40

50

実行2ステージ1132(E2)の間、ロード命令がアドレスをメモリに送る。ストア命令がアドレス及びデータをメモリに送る。結果を飽和する単一サイクル命令は、飽和が起これると、制御状態レジスタ(CSR)にSATビットを設定する。2サイクル命令では、結果が宛先レジスタファイルに書き込まれる。

【0068】

実行3ステージ1133(E3)の間、データメモリアクセスが行われる。結果を飽和する任意の乗算命令は、飽和が起これると、制御状態レジスタ(CSR)にSATビットを設定する。3サイクル命令では、結果が宛先レジスタファイルに書き込まれる。

【0069】

実行4ステージ1134(E4)の間、ロード命令が、データをCPU境界に運ぶ。4サイクル命令では、結果が宛先レジスタファイル書き込まれる。

10

【0070】

実行5ステージ1135(E5)の間、ロード命令がデータをレジスタに書き込む。これは、メモリ1151から実行5ステージ1135への入力と共に図11に概略的に図示される。

【0071】

図13は、例示の実施形態によって用いられる命令の命令コーディングの例を図示する。各命令は、32ビットで構成され、個別に制御可能な機能ユニット(乗算ユニット511、相関ユニット512、算術ユニット513、算術ユニット514、ロード/ストアユニット515)の1つの演算を制御する。ビットフィールドは、次のように定義される。クレグフィールド及びzビットは、条件付き命令において用いられるオプションのフィールドである。これらのビットは、条件付き命令が、プレディケートレジスタ及び条件を識別するために用いられる。zビット(ビット28)は、プレディケートレジスタにおいて、プレディケーションがゼロに基づくか又は非ゼロに基づくかを示す。z=1の場合、テストはゼロに等しいことに対するものである。z=0の場合、テストは非ゼロに対するものである。クレグ=0及びz=0のケースは、常に真として扱われ、無条件の命令実行を可能にする。クレグフィールド及びzフィールドは、命令において表2に示されるようにエンコードされる。

20

【表2】

Table 2

30

Conditional Register 条件付きレジスタ	creg クレグ			z
	31	30	29	28
Unconditional 無条件	0	0	0	0
Reserved 予備	0	0	0	1
A0	0	0	1	z
A1	0	1	0	z
A2	0	1	1	z
A3	1	0	0	z
A4	1	0	1	z
A5	1	1	0	z
Reserved 予備	1	1	x	x

40

なお、zビットの列における「z」は、上述のゼロ/非ゼロ比較選択を指し、「x」は、ドントケア(don't care)状態を指す。このコーディングは、16グローバルスカラーレジスタのサブセットのみをプレディケートレジスタとして特定し得る。この選択は、命令コーディングにおいてビットを保持するために成される。なお、無条件の命令は、これらのオプションのビットを有さない。無条件の命令では、これらのビット(28~31)

50

は、好ましくは付加的オペコードビットとして用いられる。しかしながら、必要に応じ、実行パッケージが、同じ実行パッケージにある命令のための4ビットクレグ/zフィールドを含む一意の32ビット条件コード拡張スロットを含み得る。表3は、そのような条件コード拡張スロットのコーディングを示す。

【表3】

Table 3

Bits ビット	Functional Unit 機能ユニット
3:0	.L
7:4	.S
11:5	.D
15:12	.M
19:16	.C
23:20	.B
28:24	Reserved 予備
31:29	Reserved 予備

10

このように、条件コード拡張スロットは、同じ実行パッケージにおいてクレグ/zビットが特殊な機能ユニットにアサインされるのと同じ様式でデコードされたビットを特定する。

20

【0072】

特殊ベクトルプレディケート命令は、ベクトル演算を制御するために、指定されたプレディケートレジスタを用いる。この実施形態において、全てのこれらのベクトルプレディケート命令は、バイト(8ビット)データ上で動作する。プレディケートレジスタの各ビットは、データの対応するバイト上でSIMD演算が成されるか否かを制御する。プレディケートユニット517の演算は、複数のベクトル比較に基づく様々な複素ベクトルSIMD演算を可能にする。例えば、2つの比較を用いてレンジ判定を行うことができる。候補ベクトルが、第1のデータレジスタ内にパックされた最小限のレンジを有する第1のベクトル参照と比較される。候補ベクトルの第2の比較が、第2のデータレジスタ内にパックされた最大限のレンジを有する第2の参照ベクトルと比較される。2つの結果のプレディケートレジスタの論理的組み合わせが、候補のベクトルの各データ部分がレンジ内かレンジ外かを判定するためのベクトル条件付き演算を可能にし得る。

30

【0073】

dstフィールドは、命令結果の宛先として、対応するレジスタファイルにおけるレジスタを特定する。

【0074】

src2フィールドは、第2のソースオペランドとして、対応するレジスタファイルにおけるレジスタを特定する。

40

【0075】

src1/cstフィールドは、命令オペコードフィールド(ビット2~12、及び付加的に、無条件の命令のためにビット28~31)に応じて幾つかの意味を有する。第1の意味は、対応するレジスタファイルのレジスタを第1のオペランドとして特定する。第2の意味は即値定数である。命令タイプに応じて、これは、符号なし整数として扱われ、特定されたデータ長にゼロ拡張されるか、又は、符号付き整数として扱われ、特定されたデータ長に符号拡張される。

【0076】

オペコードフィールド(全ての命令に対してビット2~12、及び付加的に、無条件の命令に対してビット28~31)は、命令のタイプを特定し、適切な命令オプションを指

50

定する。これは、機能ユニット及び実施される演算の指定を含む。後述する命令オプションを除き、オペコードの詳細な説明は本記載の範囲に限定されない。

【0077】

pビット(ビット0)は実行パケットをマークする。pビットは、後に続く命令と並列に実行するか否かを決定する。pビットは、下位アドレスから上位アドレスへスキャンされる。現在の命令に対してp=1の場合、次の命令が現在の命令と並列に実行する。現在の命令に対してp=0の場合、次の命令は現在の命令の後のサイクルにおいて実行する。並列に実行する全ての命令は、実行パケットを構成する。実行パケットは、最大8個の命令を含み得る。実行パケットにおける各命令は、異なる機能ユニットを用いなければならない。

10

【0078】

関連ユニット512、及び算術ユニット513及び514は、しばしば、SIMD(single instruction multiple data)モードで動作する。このSIMDモードでは、2つのオペランドからパックされたデータに同じ命令が適用される。各オペランドは、所定のスロットに配置される複数のデータエレメントを持つ。SIMD演算は、データ境界においてキャリー制御によって可能にされる。そのようなキャリー制御は、種々のデータ幅上での演算を可能にする。

【0079】

図14はキャリー制御を図示する。ANDゲート1401が、オペランド幅算術論理ユニット内のビットNのキャリー出力を受け取る(算術ユニット513及び514に対して256ビット、関連ユニット512に対して512ビット)。ANDゲート1401はまた、後述されるキャリー制御信号を受け取る。ANDゲート1401の出力は、オペランド幅算術論理ユニットのビットN+1のキャリー入力に供給される。ANDゲート1401等のANDゲートは、あり得るデータ境界における全ビットペア間に配置される。例えば、8ビットデータに対して、そのようなANDゲートは、ビット7と8、ビット15と16、ビット23と24の間等となる。そのようなANDゲートの各々は、対応するキャリー制御信号を受け取る。データサイズが最小である場合、各キャリー制御信号は0であり、隣接するビット間のキャリー送信を効果的にブロックする。選択されたデータサイズが両方の算術論理ユニットセクションを必要とする場合、対応するキャリー制御信号は1である。下記の表4は、8ビット、16ビット、32ビット、64ビット、又は128ビットのセクションに分けられ得る、算術ユニット513及び514において用いられるような256ビット幅オペランドのケースに対するキャリー制御信号の例を図示する。最上位ビットのキャリー出力の制御は必要ないので、31個のキャリー制御信号のみが必要とされる。

20

30

【表4】

Table 4

Data Size データサイズ	Carry Control Signals キャリー制御信号
8ビット	-000 0000 0000 0000 0000 0000 0000 0000
16ビット	-101 0101 0101 0101 0101 0101 0101 0101
32ビット	-111 0111 0111 0111 0111 0111 0111 0111
64ビット	-111 1111 0111 1111 0111 1111 0111 1111
128ビット	-111 1111 1111 1111 0111 1111 1111 1111
256ビット	-111 1111 1111 1111 1111 1111 1111 1111

40

当分野では、2の整数乗( $2^N$ )であるデータサイズ上で演算することが典型的である。しかしながら、このキャリー制御技術は、2の整数乗に限定されない。当業者であれば、どのようにしてこの技術を他のデータサイズ及び他のオペランド幅に適用するか理解するであろう。

50

## 【0080】

図15は、本発明を適用し得るコンピュータクラスタ1500を図示する。コンピュータクラスタ1500は、好ましくは、単一の集積回路において具現化される。コンピュータクラスタ1500は、6個のデュアルコアユニット1510、1520、1530、1540、1550、及び1560を含む。各デュアルコアユニット1510、1520、1530、1540、1550、及び1560は、図2において200で図示されるようなスカラーデュアルコアユニット、又は、図4において図示される400等のベクトルデュアルコアユニットであり得る。典型的なデュアルコア1510は、第1のCPU0 1511、第2のCPU1 1512を含み、その各々が、別個のL1I及びL1Dキャッシュ、共有レベル2(L2)キャッシュ1513、電力制御回路1514及びその他のサポート回路1514を含む。6個のデュアルコアユニット1510、1520、1530、1540、1550、及び1560は、マルチコア共有メモリコントローラ1570に接続される。マルチコア共有メモリコントローラ1570は、MSMCコア1571、システムオンチップ(SOC)インタフェース1572、電力回路1573、及び種々の標示されていないサポート回路を含む。マルチコア共有メモリコントローラ1570は、コンピュータクラスタ1500とシステムメモリとの間のインタフェースとして働き、レベル3(L3)キャッシュ、外部インタフェース等を含み得る。マルチコア共有メモリコントローラ1570は、コンピュータクラスタ1500の種々のパーツをL3SRAM/キャッシュと相互接続する。標示されていないサポート回路は、SOC毎に必要なL3SRAM、クロック、DFT(design for test)、電力管理、及びSOCモジュールと

10

20

## 【0081】

デュアルコア1510、1520、1530、1540、1550、及び1560は、マルチコアバスアーキテクチャ(MBA)を介して、MSMC 1570と相互接続される。MBAプロトコルは、従来の通信ハンドシェイクをクレジットシステムで置き換える。MBAクレジットシステムは、スレーブの、利用可能なクレジットの数をトラッキングすることによってトランザクションをアクセプトする能力を、マスターが正確にトラッキングするためのメカニズムを提供する。

## 【0082】

マスターからスレーブに送られる各トランザクションは、マスターが少なくとも1つのクレジットを使うことを必要とする。クレジットは、マスターに、特定のタイプのトランザクションを送る権利を与える有限のリソースである。一般的に、クレジットは、トランザクションが、スレーブによって受け取られると常駐することになるスレーブにおける物理ストレージ位置を表す。

30

## 【0083】

マスターは、所有しているクレジットの総数を正確にトラッキングすることが必要である。各トランザクションに対して、少なくとも1つのクレジットがマスターによって使われ(spend)、その結果、利用可能なクレジットの数が減少する。マスターが所与のクレジットタイプに対して、全ての利用可能なクレジットを使った場合、マスターは、そのクレジットタイプを必要とするトランザクションを送ることはできない。マスターが、トランザクションによって必要とされるより少ないクレジットを保有することもあり得る。特殊なタイプの利用可能なクレジットは、スレーブにおいて利用可能なデータバッファの数をトラッキングし得る。マスターは、必要とされるクレジットのタイプ及び数が不足しているトランザクションを開始することはできない。エージェントが、書き込み量を、保証されているスレーブ容量に対応する利用可能なクレジットタイプ及び数に限定するように、トランザクションを限定し得る。

40

## 【0084】

スレーブは、クレジットタイプに対応する物理リソースがフリーにされると、マスターにクレジットをリターンバックしなければならない。このように、マスターはクレジットを使い、スレーブがクレジットをリターンする。システムにおけるクレジットの総数は固

50

定され、それが閉鎖システムである。クレジットタイプに対するトランザクションタイプのマッピングは、実装依存であり、MBAプロトコルによって特定されない。

【0085】

MBAプロトコルにおいてつくられるクレジットシステムは、マスターとスレーブとの間の交渉を変化させて、より高い性能、より高い稼働インタフェースをつくる。従来のハンドシェイクは、トランザクションをアクセプト及び処理する前に、スレーブがトランザクションを肯定応答することを必要とするので、そのようなトランザクションが、肯定応答を待っている間、バスをブロックし得る。従来の通信ハンドシェイクにおいて、マスターは、スレーブがトランザクションをアクセプトすることが可能か否かを知らないし、知る必要がない。マスターは、トランザクションをバス上に置き、要求をアサートする。スレーブは、肯定応答をアサートすることによって準備が整う任意の時点で、コマンドをアクセプトし得る。

10

【0086】

クレジットシステムは、スレーブのトランザクションをアクセプトする能力をマスターがトラッキングすることを可能にし、そのため、ハンドシェイクから肯定応答が取り除かれる。マスターが各クレジットタイプに対して利用可能なクレジットの数をトラッキングするので、マスターは、スレーブによってアクセプトされることを保証されているトランザクションのみを送信する。

【0087】

クレジットは、スレーブにおける物理リソースを表す有限のリソースである。クレジットは、マスターによって各トランザクションに対して使われる。クレジットは、スレーブが、クレジットタイプに関連する物理リソースをフリーにするために十分にトランザクションを処理すると、スレーブからマスターにリターンされなければならない。クレジットのリターンは、スレーブにおいて進行中のトランザクション処理の後、成され、このアクションをクリティカルバスから取り除く。これとは逆に、従来のハンドシェイクは、トランザクションを処理するために、クリティカルバスにスレーブ肯定応答を置く。

20

【0088】

MBAクレジットシステムには、従来の通信ハンドシェイクに比べて2つの重要な利点がある。第1の利点は、トランザクションが、正確に1つのサイクルの間、通信チャンネルを占有することである。これは、単一のトランザクションが、スレーブからの肯定応答を待っている間、チャンネルをブロックすることを防止する。この結果、特に、同じ物理チャンネルが複数の仮想又は物理マスターによって共有されているとき、スルーブットが一層高い通信チャンネルとなる。第2の利点は、トランザクションを送信するための要件としての肯定応答信号を除去することによって、マスターからスレーブへトランザクションを送信するために必要とされる通信ハンドシェイクが短縮されることである。これはトランザクションをマスターからスレーブへ通信することの全体のレイテンシを削減し、その結果、より高性能な通信となる。

30

【0089】

マルチコアバスアーキテクチャ(MBA)は、マルチプロセッサプラットフォームにおけるデバイス間の内部通信のために設計された、高性能なポイントツーポイント単一方向バスプロトコルである。MBAプロトコルは下記の特性を有する。MBAプロトコルは、ポイントツーポイントである。正確に2つのエージェントが別個のトランザクションによって通信する。一方のエージェントがマスターとして機能し、他方のエージェントがスレーブとして機能する。単一のマスターと複数のスレーブとの間のブロードキャストはサポートされていない。MBAプロトコルは単一方向性である。コマンドトランザクションが、マスターによってのみ開始され得る。スレーブは、コマンドトランザクションを開始しない可能性があり得る。応答トランザクションは、第2の、ミラーリングされたインタフェース上で開始される。MBAプロトコルはノンブロッキングである。マスターによって発せられたトランザクションは、単一サイクルにおいてスレーブによってアクセプトされなければならない。バス上のストールしているトランザクションに対する対策はない。

40

50

M B A プロトコルはクレジットベースである。各トランザクションが、関連するクレジット及びクレジットタイプを有する。マスターは、トランザクションを開始し得る前に、適切なクレジットを有さなければならない。適切なクレジットタイプのクレジットの所有によって、スレーブがそのトランザクションをアクセプトすることが保証される。

【 0 0 9 0 】

プロトコルにおけるデバイス及びそれらの挙動を正確に識別するために、本明細書を通して、次の用語が用いられる。

【 0 0 9 1 】

エージェント：本明細書で定義される M B A プロトコルを用いて通信する単一デバイス又はロジックコントローラ。

10

【 0 0 9 2 】

エージェントペア：M B A プロトコルを用いて通信する 2 つのエージェント。

【 0 0 9 3 】

トランザクション：マスターからスレーブへのデータの転送を伴う可能性もあり、伴わない可能性もある、2 つのエージェントであるマスターとスレーブとの間の読み出し、書き込み、又はメッセージ通信。

【 0 0 9 4 】

マスター：トランザクションを開始するエージェント。

【 0 0 9 5 】

スレーブ：マスターによって開始されるトランザクションを受け取るエージェント。

20

【 0 0 9 6 】

コアサイド：両方のエージェントが、(例えば、キャッシュ階層において)ハードウェアキャッシュを有し得る場合の、トランザクションにおいて C P U コアに近い方のエージェント。

【 0 0 9 7 】

ファーサイド：両方のエージェントが、例えばキャッシュ階層において、ハードウェアキャッシュを有する場合の、トランザクションにおいて C P U コアから遠い方のエージェント。

【 0 0 9 8 】

データフェーズ：トランザクションデータチャネル ( T D C ) 上で単一サイクルにおいて転送されるデータの単一のビット。

30

【 0 0 9 9 】

データ転送：単一 `did [ 1 1 : 0 ]` に対するデータフェーズの完全な集合。最終データフェーズは、`dlast` 信号のアサートによって示される。

【 0 1 0 0 】

データウィンドウ：`ddata [ C F G : 0 ]` 信号の幅によって決定されるようなデータ転送の最大サイズ。デフォルトの幅は 5 1 2 ビットである。

【 0 1 0 1 】

コマンドトランザクション：2 つのエージェントの間でシーケンスを始めるトランザクションのタイプ。コマンドトランザクションが、一意の `cid [ 1 1 : 0 ]` によって区別され、データの転送を伴う可能性もあり、伴わない可能性もある。

40

【 0 1 0 2 】

応答トランザクション：2 つのエージェントの間でシーケンスを完了するトランザクションのタイプ。応答トランザクションは、コマンド I D ( `cid [ 1 1 : 0 ]` ) によって、前に開始されたコマンドトランザクションとマッチングされ、データの転送を伴う可能性もあり、伴わない可能性もある。

【 0 1 0 3 】

書き込み応答：書き込みコマンドトランザクションのためのシーケンスを完了する書き込み応答チャンネル ( W R C ) 上の専用応答。

【 0 1 0 4 】

50

トランザクションシーケンス：ペアにされたインタフェーストポロジーを用いる2つのエージェント間の完全なプロトコルレベル通信。シーケンスは、一方のエージェントによって開始されるコマンドトランザクションで始まり、他方のエージェントからの応答トランザクションか又は書き込み応答のいずれかによって完了する。

【0105】

クレジット：マスターに、単一トランザクションを発する権利を与える単一原子単位。アクセプトの肯定応答を必要とせずに、スレーブが単一サイクルにおいてアクセプトすることを保証する。クレジットはクレジットタイプによって分類される。クレジットは、マスターによって保有され、使われる。クレジットは、十分な処理が成された後にスレーブによってリターンされ、それによって、マスターは同一クレジットタイプの新しいトランザクションを開始し得る。

10

【0106】

クレジットタイプ：使われる又はリターンされるクレジットの種類を識別するためにマスターによって用いられるクレジットの、総称的に定義されるが特定のみに実装される分類。クレジットタイプは、典型的に、スレーブエージェントにおける特定の物理リソースに関連する。マスターに対して利用可能なリソースの個々に割り当て可能なスロットの数は、そのクレジットタイプに対する利用可能なクレジットの数に正確に等しい。マスター及びスレーブは、各総称的なクレジットタイプの厳密な定義において合意を有さなければならない。

20

【0107】

クレジットカウント：使われるべき、マスターにとって利用可能な所与のクレジットタイプのクレジットの現在の数。

【0108】

クレジットスPEND：トランザクション属性チャンネル(TAC)上の有効なトランザクションを識別するvalid信号のアサート、及びccredit[CFG:0]信号上の1つ又は複数の有効なクレジットタイプのアサートを介して、TAC上でトランザクションを開始するアクション。マスターは、クレジットスPENDが許可される前に、各関連するクレジットタイプに対して少なくとも1つのクレジットを有さなければならない。マスターは、各クレジットスPENDに対して関連するクレジットタイプを1つ減らさなければならない。クレジットスPENDは、暗示的なコマンドレディを備えるコマンド要求と等価である。下記に説明するように、複数の異なるタイプのクレジットがあり得る。クレジットスPENDは、トランザクションに対応するクレジットタイプのものでなければならない。また、クレジットスPENDは、トランザクションをサポートするために十分な数のクレジットを含まなければならない。このように、データ転送の場合において、マスターによって費やされるクレジットの数は、データ長、及び受け取り側のスレーブのバッファ容量に対応し得る。

30

【0109】

クレジットリターン：前のクレジットスPENDに回答して、スレーブからマスターにクレジットをリターンするアクション。マスターは、クレジットリターンに回答して、関連するクレジットタイプを増加させる。

40

【0110】

クレジットリミット：リセットの際、マスターに対して利用可能な特定のクレジットタイプに対するクレジットの最大数。クレジットリミットは、第1のクレジットリターンを受け取る前にマスターによって開始され得る個々のクレジットスPENDアクションの最大数を表す。

【0111】

チャンネル：トランザクションを記述すること又はデータを転送すること等の、共通の目的を集合的に果たすインタフェース上の信号の集合。

【0112】

クレジットされた(credited)チャンネル：クレジットを用いるチャンネル。これには、ク

50

レジットスペンド及びクレジットリターンの両方のためのシグナリングが含まれる。

【0113】

キャッシュビリティドメイン：それに対して割り当てが可能であることをトランザクションが指示し得る、潜在的にキャッシュ階層の異なるレベルにあるキャッシュの集合である。

【0114】

シェアビリティドメイン：潜在的にキャッシュ階層の異なるレベルにあるキャッシュの集合である。これを備えると、コヒーレントなトランザクションが、コヒーレンスプロトコルの規則に従うことを期待される。

【0115】

MBAプロトコルは、好ましくは、エージェントがMESIハードウェアキャッシュコヒーレンスプロトコルを実装するためのシグナリングサポートを提供する。サポートされたトランザクションタイプ及び応答状態信号は、ACEとして知られるARM AXIコヒーレンス拡張に基づく。コヒーレンスプロトコルは、データを変更する前にキャッシュライン精度で排他的所有権を執行することによって、同じシェアビリティドメインにおけるキャッシュがコヒーレントなままであるように規定する。コヒーレンスプロトコルの完全な取り扱いとは本明細書の範囲外であるが、ここでは、MBAプロトコルにおいて用いられるコヒーレンスシグナリングのための背景を提供するために概要を示す。

【0116】

標準MESIキャッシュコヒーレンスプロトコルは、キャッシュラインのための4つの状態、即ち、変更、排他的、共有、及び無効、を提供する。

【0117】

変更：変更された状態を有するキャッシュラインであり、メインメモリに対して変更されている。このキャッシュラインは、同じシェアビリティドメインにおいて、他のどのキャッシュにも存在しない。

【0118】

排他的：排他的状態を有するキャッシュラインであり、メインメモリに対して変更されていないが、このキャッシュラインは、同じシェアビリティドメインにおいて、他のどのキャッシュにも存在しない。

【0119】

共有：共有状態を有するキャッシュラインであり、メインメモリに対して変更されていない。このキャッシュラインは、同じシェアビリティドメインにおいて、他のキャッシュに存在し得る。

【0120】

無効：無効状態を有するキャッシュラインであり、キャッシュに存在しない。そのようなキャッシュラインにストアされた如何なるデータも無効であり、アクセスされるべきではない。

【0121】

プロトコルにおけるキャッシュコヒーレンスサポートは、トランザクション間に複数の依存関係を導入することを必要とする。例えば、コヒーレントReadCleanトランザクションの結果、1つ又は複数のCleanSharedスヌープトランザクションが生成し得、このCleanSharedスヌープトランザクションは、ReadCleanトランザクションの適法な完了の前に完了しなければならない。このケースでは、ReadCleanは、完了するためにCleanSharedに依存する。ACE仕様は、AXIプロトコルに対するコヒーレンス拡張に関連する多くのそのような依存関係を詳述している。

【0122】

依存関係は、一つのトランザクションが別のトランザクションが完了するまで、ブロックしなければならない場合のような、ブロッキングシナリオをつくる。また、コヒーレントトランザクションの順も、ブロッキングシナリオをつくる。例えば、オーバーラップす

10

20

30

40

50

るアドレスに対する2つのコピーレント書き込みトランザクションは、何らかの方式で順序付けされなければならない。一方の書き込みトランザクションが他方によってブロックされる。

【0123】

デッドロックは、解消できない相互依存関係が存在する状態である。これは、依存性グラフにおけるループと考えることもできる。システムデッドロックを防止するために、ブロッキング規則が確立されなければならない。MBAプロトコルにおける全てのトランザクションが、ブロッキングトランザクションか又はノンブロッキングトランザクションのいずれかとして分類され得る。

【0124】

ブロッキングトランザクションは、下記の全てにより特徴付けられる。ブロッキングトランザクションは、1つ又は複数のスヌープコマンドトランザクションを生成することを許可されている。ブロッキングトランザクションは、完了するために、スヌープコマンドトランザクションの完了に依存することを許可されている。ブロッキングトランザクションは、完了するために、ノンブロッキングトランザクションの完了に依存することを許可されている。ブロッキングトランザクションは、完了するために、ブロッキングトランザクションの完了に依存することを許可されている。

【0125】

ノンブロッキングトランザクションは、下記の全てにより特徴付けられる。ノンブロッキングトランザクションは、スヌープコマンドトランザクションを生成することを許可されていない。ノンブロッキングトランザクションは、スヌープコマンドトランザクションの完了に依存することを許可されている。ノンブロッキングトランザクションは、ブロッキングトランザクションの完了に依存することを許可されていない。ノンブロッキングトランザクションは、ノンブロッキングトランザクションの完了に依存することを許可されている。

【0126】

後述の表6は、MBAプロトコルにおける各トランザクションを、ブロッキング又はノンブロッキングとして識別する。

【0127】

単一のMBAインタフェースが、3つの別個のチャンネルに分割され、それらは、マスターからスレーブへの単一方向データフローを備える完全なトランザクションのために共に動作する。3つのMBAインタフェースチャンネルは、要約すると次のようになる。トランザクション属性チャンネル(TAC)は、マスターからスレーブへの単一のトランザクションのための全ての属性を特定するクレジットされたチャンネルである。トランザクションデータチャンネル(TDC)は、マスターからスレーブへのデータ転送に用いられるクレジットされない(non-credited)チャンネルである。各TDCは、TAC上で前に又は同時に開始されたトランザクションにマッチングするIDを含む。書き込み応答チャンネル(WRC)は、スレーブに書き込み応答を送るために、マスターによって用いられるクレジットされたチャンネルである。各WRCは、ペアにされたインタフェースポートにおける反対側のTAC上で前に開始されたトランザクションにマッチングするIDを含む。

【0128】

図16は、単一のMBAインタフェースのための各チャンネルに対する信号方向を図示する。図16は、マスターエージェント1610及びスレーブエージェント1620を含む。データフロー1611によって示されるように、マスターエージェント1610は、スレーブエージェント1620にシグナリングする。マスターエージェント1610は、TAC 1612、TDC 1613、及びWRC 1614を介して、信号をスレーブ1620に送信する。

【0129】

図17は、ペアにされたインタフェースポート(PIT)における各チャンネルを図示する。完全なトランザクションシーケンスを実施するために、エージェントペアは、ミ

10

20

30

40

50

ラーリングされたペア構成の2つの別個のMBAインタフェースを用いて通信しなければならない。これは対インタフェースと称される。対インタフェースは、2つのエージェント間で双方向のデータ転送をサポートする。図17は、コアサイドエージェント1710及びファーサイドエージェント1720を図示する。コアサイドエージェント1710は、CPUコアに近い方のエージェントである。ファーサイドエージェント1720は、CPUコアから遠い方のエージェントである。対インタフェーストポロジにおいて、一方のエージェントは常にコアサイドエージェントであり、他方のエージェントは常にファーサイドエージェントである。

#### 【0130】

第1のインタフェースは、データフロー1711によって示されるように、ファーサイドエージェント1720に対するコマンド及び応答トランザクションの両方を開始するために、コアサイドエージェント1710によって用いられる。これらのトランザクションでは、コアサイドエージェント1710はマスターであり、ファーサイドエージェント1720はスレーブである。コアサイドエージェント1710は、TAC 1712、TDC 1713、及びWRC 1714を介して、信号をファーサイドエージェント1720に送信する。

10

#### 【0131】

第2の、ミラーリングされたインタフェースは、データフロー1721によって示されるように、コアサイドエージェント1710に対するコマンド及び応答トランザクションの両方を開始するために、ファーサイドエージェント1720によって用いられる。ファーサイドエージェント1720は、TAC 1722、TDC 1723、及びWRC 1724を介して、信号をコアサイドエージェント1710に送信する。

20

#### 【0132】

図18は、別の例示の対インタフェーストポロジ(PIT)における各チャネルを図示する。図18は、コアサイドエージェント1810及びファーサイドエージェント1820を図示する。下記の図18は、対インタフェーストポロジを示す。この例では、読み出しシーケンスが示される。コアサイドエージェント1810は、マスターとして機能しながら、インタフェース上で読み出し演算を開始する。そのような読み出し演算は、コアサイドエージェント1810からファーサイドエージェント1820へのデータ転送を必要としない。このように、このシーケンスに対して、トランザクションデータチャンネル(TDC)がコアサイドエージェント1810によって用いられない。データフロー1811に従って、コアサイドエージェント1810は、マスターとして、TAC 1712を介して、信号をファーサイドエージェント1820に送信する。トランザクションシーケンスを完了するために、ファーサイドエージェント1820は、マスターとして(データフロー1821に従って)TAC 1822を介して読み出し応答トランザクションを開始し、読み出されたデータをTDC 1823上に供給する。

30

#### 【0133】

対インタフェーストポロジは、各エージェントが、コマンド及び応答トランザクションの両方を開始することを許可し、必要に応じ他のエージェントにデータを転送する。データフローは、マスターからスレーブへの単一の方向に制約される。この理由により、各エージェントは、1つのインタフェース上でマスターとして機能し、第2のミラーリングされたインタフェース上でスレーブとして機能する。

40

#### 【0134】

トランザクション属性チャンネル(TAC)は、単一のクレジットされたトランザクションを開始するために必要な全ての属性を特定するクレジットされたチャンネルである。トランザクション属性を形成する全てのTAC信号は、正確に1サイクルの間にマスターによってアサートされ、同じサイクルにおいてスレーブによってアクセプトされなければならない。TAC上でのトランザクションの開始は、クレジットの使用を必要とする。クレジットの使用は、スレーブが、トランザクションをアクセプトするために内部物理リソース内にスペースを有することを保証する。結果として、スレーブは、TAC上でアサートさ

50

れたあらゆる新しいトランザクションをアクセプトすることを要求される。

【0135】

マスターは、利用可能なクレジットによってのみ制限されるが、サイクル毎に新たなトランザクションを開始し得る。トランザクションは、データの転送を必要とする可能性も、必要としない可能性もある。TAC上で開始されたトランザクションがデータの転送を必要とする場合、トランザクションデータチャンネル(TDC)が用いられる。

【0136】

c v a l i dのアサートと共に、マスターによって使われる全てのクレジットは、c c r e d i t [ C F G : 0 ]信号によって決定されるクレジットタイプを有する。トランザクションがハードウェアリソースから成功裏に除去されると、スレーブは、c r e t u r n [ C F G : 0 ]信号を用いて、同じタイプのクレジットをマスターにリターンバックしなければならない。各インタフェースに対して全てのTAC信号が必要とされるわけではない。表5は、好ましい実施形態に従った、トランザクション属性チャンネル信号を列挙する。

【表 5】

信号	ビット サイズ	説明	必要/ 任意	ドライバ
cvalid	1	有効なトランザクションの存在を示す	必要	マスター
ccredit	CFG	クレジットスPENDビットベクトル。トランザクションによって使われるクレジットの種類を示す。	必要	マスター
cid	12	トランザクションID	必要	マスター
ctype	6	トランザクションタイプ	必要	マスター
cmstid	12	マスターID	任意	マスター
cdata	1	トランザクションが、関連するデータを有するか否かを示す	任意	マスター
cpriority	3	トランザクションの優先レベル	任意	マスター
csband	CFG	応答においてミラーリングされないユーザー定義サイドバンド信号	任意	マスター
cmsband	CFG	応答においてミラーリングされるユーザー定義サイドバンド信号	任意	マスター
caddress	48	トランザクションアドレス	任意	マスター
cmemtype	2	トランザクションのメモリタイプ (MMU)	任意	マスター
cpable	1	プリフェッチ可能であることを示す (MMU)	任意	マスター
ccinner	2	内部キャッシュアビリティドメインに対するキャッシュ割り当てポリシー (MMU)	任意	マスター
ccouter	2	外部キャッシュアビリティドメインに対するキャッシュ割り当てポリシー (MMU)	任意	マスター
csdomain	1	シェアアビリティドメインを示す (MMU)	任意	マスター
cbytecnt	7	バイトでのトランザクションサイズ	任意	マスター
cemudbg	1	トランザクションエミュレーションレベル	任意	マスター
csecure	1	マスターセキュアレベル	任意	マスター
cpri	1	マスター特権レベル	任意	マスター
ccoh	2	コヒーレンス状態情報	任意	マスター
cdirty	1	コヒーレンスダーティライン状態	任意	マスター
cstatus	2	応答成功/失敗状態	任意	マスター
ctrace	4	トランザクショントレース情報	任意	マスター
creturn	CFG	クレジットリターンベクトル、各クレジットタイプに対して1ビット	必要	スレーブ

## 【 0 1 3 7 】

信号 cvalid、ccredit、cid、ctype、cmstid、cdata、cpriority、及びcsbandは、全てのトランザクションにおいて用いられ得る。信号caddress、cmemtype、cpable、ccinner、ccouter、csdomain、cbytecnt、cemudbg、及びcsecure

10

20

30

40

50

e は、コマンドトランザクションにおいてのみ用いられ得る。信号 `ccoh`、`cdirty`、`cstatus`、及び `ctrace` は、応答トランザクションにおいてのみ用いられ得る。信号 `creturn` は、クレジットリターントランザクションにのみ用いられ得る。

【0138】

下記は、トランザクション属性チャンネル上の上述された信号の説明である。表5に記載されるように、幾つかの信号は必要として指定され、幾つかは任意として指定される。チャンネルがインタフェース上に存在する場合、必要とされる信号が存在しなければならない。構成にわたって全チャンネルが除去される場合、必要/任意の指定は適用可能ではない。

【0139】

`valid` 信号は、TAC上の有効なトランザクションの存在を識別する。0はトランザクションがないことを示す。1は有効なトランザクションがバス上にあることを示す。

【0140】

`ccredit[CFG:0]` 信号は、このトランザクションによってクレジットが使われていることを識別するビットベクトルである。利用可能なクレジットタイプの数、及び従って、この信号の幅は、構成オプションである。好ましい実施形態において、MBAプロトコルは、ジェネリック整数クレジットタイプを定義する。クレジットタイプは、ゼロから漸増的に番号が付される符号なしの整数である。`creturn[CFG:0]` 信号の幅は、`ccredit` の幅にマッチングしなければならない。トランザクション属性チャンネル(TAC)上のトランザクションの存在を示すように `valid` がアサートされるとき、`ccredit` は、アサートされた少なくとも1つのビットと共に非ゼロ値を1にキャリーする必要がある。複数のクレジットが、単一のトランザクションと共に使われ得る。これは `ccredit` ビットベクトルにおいて複数のビットをアサートすることによって達成される。マスター及びスレーブによるクレジットタイプの解釈及び物理リソースへのマッピングは、実装固有である。

【0141】

`cadress[47:0]` 信号は、現在のトランザクションに対する開始アドレスを識別する。アドレスは、仮想アドレス、物理アドレス、又は他の任意の中間アドレス表現であり得る。アドレスは、整合及び非整合トランザクションの両方に対するトランザクションのための開始バイトを特定する。この信号は、マスターがコマンドトランザクションを開始するインタフェース上で必要とされる。

【0142】

`cmstid[11:0]` 信号は、このトランザクションを開始するシステムマスターを一意的に識別する。`cmstid` 信号は、正確なセキュリティ保護チェックを可能にするために、現在のマスターの外で発するトランザクションに対して正確に伝搬されるべきである。

【0143】

`cid[11:0]` 信号は、現在のトランザクションを一意的に識別する。コマンドトランザクションが `cid` を確立する。トランザクションデータチャンネル(TDC)上の対応するデータ転送は、`did[11:0]` 信号を介して `cid` 信号をミラーリングする。後続の応答トランザクション及び書き込み応答は、`cid` 及び `wid[11:0]` 信号を介して `cid` をミラーリングする。

【0144】

`ctype[5:0]` 信号は、現在のトランザクションのタイプを識別する。表6は、6ビット `ctype` 信号の好ましいデコーディングを列挙する。トランザクションタイプは、トランザクションのカテゴリを識別する3つの最上位ビット、及びカテゴリ内の特定のタイプを識別する3つの最下位ビットを用いてエンコードされる。

10

20

30

40

【表 6】

cytype	説明	カテゴリ	ブロッキング/ノン ブロッキング
000 000	ReadNoSnoop	読み出し	ノンブロッキング
000 001	ReadOnce	読み出し	ブロッキング
000 010	ReadClean	読み出し	ブロッキング
000 011	ReadUnique	読み出し	ブロッキング
001 000	WriteNoSnoop	書き込み	ノンブロッキング
001 001	WriteUnique	書き込み	ブロッキング
001 010	WriteLineUnique	書き込み	ブロッキング
001 011	WriteClean	書き込み	ノンブロッキング
001 100	WriteBack	書き込み	ノンブロッキング
010 000	SnoopReadOnce	スヌープ	ノンブロッキング
010 001	SnoopReadShared	スヌープ	ノンブロッキング
010 010	SnoopCleanShared	スヌープ	ノンブロッキング
010 011	SnoopCleanInvalid	スヌープ	ノンブロッキング
001 100	SnoopMakeInvalid	スヌープ	ノンブロッキング
100 000	DVMOperation	メッセージ	ノンブロッキング
100 001	DVMSync	メッセージ	ノンブロッキング
100 010	DVMComplete	メッセージ	ノンブロッキング
101 000	CacheWarm	メッセージ	ブロッキング
101 001	CacheWarmNotify	メッセージ	ブロッキング
101 010	予備		
101 011	Evict	メッセージ	ノンブロッキング
110 000	CleanSharedPOU	メンテナンス	ブロッキング
110 001	CleanInvalidPOU	メンテナンス	ブロッキング
110 010	MakeInvalidPOU	メンテナンス	ブロッキング
111 000	Read Response	応答	ノンブロッキング
111 001	Snoop Response	応答	ノンブロッキング
111 010	Message Response	応答	ノンブロッキング
111 011	Maintenance Response	応答	ノンブロッキング

10

20

30

40

他のあり得る全てのコーディングは予備であり、好ましい実施形態において用いられていない。

50

## 【 0 1 4 5 】

c m e m t y p e [ 1 : 0 ] 信号は、仮想メモリ翻訳によって決定されるように、所与のトランザクションアドレスに対するメモリタイプを識別する。3つのサポートされたメモリタイプは、異なる挙動上の意味を有する。00のコーディングは、デバイスメモリタイプを示す。01のコーディングは、通常のライトバックキャッシュメモリタイプを示す。10のコーディングは、通常のライトスルーキャッシュメモリタイプを示す。11のコーディングは予備であり、好ましい実施形態において用いられていない。

## 【 0 1 4 6 】

c p a b l e 信号は、仮想メモリ翻訳によって決定されるように、トランザクションアドレスがプリフェッチ可能か否かを識別する。プリフェッチ可能とマークされたメモリ領域は、ハードウェアプリフェッチストリームの開始に貢献し得る。0は、プリフェッチ不可能を示す。1はプリフェッチ可能を示す。

10

## 【 0 1 4 7 】

c c i n n e r [ 1 : 0 ] 信号タイプは、仮想メモリ翻訳によって決定されるように、トランザクションアドレスに対する内部キャッシュアビリティドメインに対応するキャッシュ割り当て属性を識別する。

## 【 0 1 4 8 】

c c o u n t e r [ 1 : 0 ] 信号は、仮想メモリ翻訳によって決定されるように、トランザクションアドレスに対する外部キャッシュアビリティドメインに対応するキャッシュ割り当て属性を識別する。

20

## 【 0 1 4 9 】

キャッシュ階層（レベル1、レベル2、レベル3キャッシュ等）は、内部ドメインと外部ドメインの、2つの相互に排他的なキャッシュアビリティドメインに分けられる。キャッシュ割り当てポリシーは、各ドメインに対して別個に確立され、仮想メモリ翻訳によってストアされるメモリ属性において記録される。内部キャッシュアビリティドメインは、CPUに最も近いキャッシュ、典型的にレベル1及びレベル2キャッシュ、を含む。外部キャッシュアビリティドメインは、内部ドメインにおいてキャッシュを含まず、最終レベルキャッシュ等の内部ドメインを超えるキャッシュを含む。内部及び外部キャッシュドメインの解釈は実装依存であり、幾つかの実装は両ドメインを同一として取り扱うことを選択し得る。

30

## 【 0 1 5 0 】

表7は、同一である c c i n n e r 及び c c o u n t e r のコーディングを列挙する。

## 【表7】

ccinner, ccounter コーディ ング	割り当てポリシー	説明
00	割り当てない	ドメインにおけるキャッシュは、割り当ててはならない
01	書き込み割り当て	書き込み割り当て。読み出しは割り当ててはならない。
10	読み出し割り当て	読み出し割り当て。書き込みは割り当ててはならない。
11	読み出し／書き込み割り当て	読み出し及び書き込み割り当て

40

## 【 0 1 5 1 】

c s d o m a i n 信号は、仮想メモリ翻訳によって決定されるように、トランザクショ

50

ンに対する、シェアビリティドメイン、内部、又は外部を識別する。シェアビリティドメインは、このトランザクションに対してコヒーレントに維持されなければならないキャッシュを識別する。

【0152】

内部及び外部の2つのシェアビリティドメインがある。2つのドメインは、相互排他的ではない。外部ドメインは、内部ドメインにおいてキャッシュを含む。内部共有ドメインは、CPUに最も近いキャッシュ、典型的にレベル1及びレベル2キャッシュ、を含む。外部共有ドメインは、最終レベルキャッシュ等の内部ドメインを超えるものに加えて、内部ドメインにおけるキャッシュを含む。内部及び外部共有ドメインの解釈は実装依存であり、幾つかの実装は、両ドメインを同一として取り扱うことを選択し得る。0は内部共有ドメインを示す。1は外部共有ドメインを示す。c s d o m a i nは、シェアブルトランザクションに対してのみ有効である。c s d o m a i n信号は、R e a d N o S n o o p及びW r i t e N o S n o o pトランザクションタイプに対して無視される。

10

【0153】

c c o h [ 1 : 0 ]信号は、読み出し応答又はスヌープ応答トランザクションに対するコヒーレンス状態情報を識別する。コヒーレンス状態情報は、開始側マスターに、キャッシュラインの最終コヒーレント状態を通知するために用いられる。読み出し応答トランザクションでは、c c o hは、読み出しが完了したときに、割り当てキャッシュにおけるラインのコヒーレント状態を示す。スヌープ応答トランザクションでは、c c o hは、スヌープが完了したときに、スヌープされたキャッシュにおけるラインのコヒーレント状態を示す。表8は、c c o hの有効なエンコーディングを列挙する。

20

【表8】

ccoh コーディング	読み出し応答	スヌープ応答
00	非コヒーレント又は非割り当て 応答	スヌープされたキャッシュラインの最終状態が無効である
01	共有状態においてキャッシュラインがリターンされる	スヌープされたキャッシュラインの最終状態が共有されている
10	一意状態においてキャッシュラインがリターンされる	スヌープされたキャッシュラインの最終状態が一意である
11	予備	

30

【0154】

c d i r t y信号は、この応答トランザクションに関連するキャッシュラインデータ転送が、メインメモリに対してダーティか又はクリーンかを識別する。c d i r t y信号は、スヌープ側のマスターに、スヌープ応答データがメインメモリにライトバックされる必要があるか否かを示すために用いられる。c d i r t y信号は、オプションで、S h a r e d D i r t y M E S I状態を実装するために、読み出し演算と共に用いられ得る。c d i r t y信号は、応答トランザクションにのみ適用し、コマンドトランザクションに対する意味を搬送せず、コマンドトランザクションに対して、0に関連付けられなければならない。0のc d i r t y信号は、キャッシュラインデータ転送がメインメモリに対して、変更されていないことを示す。1のc d i r t y信号は、キャッシュラインデータ転送がメインメモリとは異なることを示す。これは、このキャッシュラインが、今後どこかの時点で、メインメモリにライトバックされなければならないことを示す。

40

【0155】

c d a t a信号は、このトランザクションが、関連するデータ転送を伴うか否かを識別する。読み出し演算は、読み出し演算がデータ転送を含まないので、この信号をアサートすることはない。後続の読み出し応答トランザクションは、c d a t a信号をアサートし

50

得る。0の`cdata`信号は、トランザクションが転送データを含まないことを示す。1の`cdata2`信号は、トランザクションが関連データを含むことを示す。従って、書き込みコマンドトランザクションは、`cdata = 1`をアサートすることが必要とされる。読み出し応答トランザクションは、`cstatus[1:0]`において示される状態に関係なく、`cdata = 1`をアサートすることが必要とされる。

【0156】

`cbytecnt[6:0]`信号は、トランザクションの総サイズを識別する。最大の適法サイズは128バイトである。全てゼロの`cbytecnt`は、128バイトトランザクションを示す。`cbytecnt`のこのコーディングは、スヌープ及びビクティムトランザクションに対して必要とされる。そうでない場合、`cbytecnt`信号の数は、データ転送バイトの数に等しい。ゼロサイズのトランザクションは、`cbytecnt`信号によって指定することができない。しかしながら、書き込みトランザクションは、如何なるバイトも変更しない書き込みトランザクションを実施するために、TDC上の`dbyten[CFG:0]`信号をゼロ設定し得る。

10

【0157】

`cstatus[1:0]`信号は、応答トランザクションに対するトランザクション完了状態を識別する。この信号は、応答トランザクションに対してのみ有効であり、コマンドトランザクションに対して無視される。表9は、有効なエンコーディングを列挙する。

【表9】

cstatus コーディング	説明	注記
00	成功	
01	アドレスエラーに起因する失敗	無効アドレス、MMRのSRAM
10	EDCエラーに起因する失敗	修正不能なデータエラー
11	保護エラーに起因する失敗	セキュリティ/ファイアウォール違反、MMR保護エラー

20

30

【0158】

`ctrace[3:0]`信号は、読み出し応答トランザクションに対するトレース情報を識別する。この情報は、エミュレーション/デバッグアーキテクチャによって用いられる。トレース情報は、トランザクションがサービスされたメモリ階層におけるレベルを記録する。トレース情報は、トランザクションの成功/失敗状態とは関係なく記録される。成功/失敗状態は、`cstatus[1:0]`信号においてキャプチャされる。信号エンコーディングは、エミュレーション/デバッグアーキテクチャ仕様によって指定されたものにマッチングするように定義される。表10は、有効なエンコーディングを列挙する。

【表 1 0】

ctrace	エンドポイント	デバッグニーモニック
0000	CPUレジスタ	OT
0001	L1P及びL1Dキャッシュ	CB
0010	L1Pキャッシュ	CP
0011	L1Dキャッシュ	CD
0100	L2キャッシュ	C2
0101	予備	
0110	予備	
0111	L2メモリ	M2
1000	L3キャッシュ	C3
1001	共有L2 SRAM	SL2
1010	長距離ポート	LD
1011	MSMC外	XT
1100	予備	
1101	予備	
1110	予備	
1111	予備	

10

20

## 【0159】

cemudbg信号は、現在のトランザクションを、エミュレーショントランザクションか又は非エミュレーショントランザクションとして識別する。0のcemudbg信号は、通常非エミュレーショントランザクションを示す。1のcemudbg信号は、エミュレーショントランザクションを示す。

30

## 【0160】

csecure信号は、現在のトランザクションのセキュリティレベルを識別する。0のcsecure信号は、非セキュアトランザクションを示す。1のcsecure信号は、セキュアトランザクションを示す。

## 【0161】

cpriv[1:0]信号は、現在のトランザクションの特権レベルを識別する。表11は、有効なエンコーディングを列挙する。

【表 1 1】

cpriv エンコーディング	説明
00	ユーザー
01	スーパーバイザー
10	ハイパーバイザー
11	予備

40

## 【0162】

cpriority[2:0]信号は、現在のトランザクションの優先レベルを識別する。この信号は、トランザクションを仲裁するためにスレーブによって用いられる。下位の数値は、より高い優先度を示す。従って、000のcpriority[2:0]信号は、最高優先度を示し、111のcpriority[2:0]信号は、最低優先度を示す。

50

## 【0163】

`csband` [CFG: 0] 信号は、応答上でミラーリングされないオプションのユーザー定義サイドバンド信号である。この信号はオプションであり、その解釈は実装固有である。`csband` 信号は、トランザクション属性チャンネル (TAC) によってのみ用いられ、トランザクションシーケンス演算の一部としてミラーバックされる必要がない。

## 【0164】

`cmsband` [CFG: 0] 信号は、応答上でミラーリングされなければならないオプションのユーザー定義サイドバンド信号である。この信号はオプションであり、その解釈は実装固有である。コマンドトランザクションのための `cmsband` 信号は、トランザクションシーケンス演算の一部として応答トランザクションがアサートされる時、同じ値を用いてミラーリングされなければならない。

10

## 【0165】

`creturn` [CFG: 0] 信号は、それに対してこのサイクルで単一のクレジットがリターンされる、クレジットタイプを識別する。利用可能なクレジットタイプの数、従って、この信号の幅は、構成オプションである。MBA プロトコルは、ジェネリック整数クレジットタイプを定義する。マスター及びスレーブによるクレジットタイプの解釈は、実装固有である。`creturn` 信号幅は、候補となり得るクレジットタイプの数に正確にマッチングする。`creturn` 信号は、インタフェース上で使用されている各クレジットタイプに対して1ビットを備えるビットベクトルとして解釈される。`creturn` 信号の各ビットは、各サイクルでスレーブによって独立してアサートされ得る。`creturn` 信号における任意のビット上の1の値は、そのタイプの単一のクレジットのスレーブからマスターへのリターンを示す。或るサイクルで `creturn` 信号における複数のビットがアサートされる場合、信号においてアサートされた各ビットに対して、1つのクレジットがスレーブからマスターにリターンされる。インタフェース上の各クレジットタイプに対して、スレーブからマスターに、各サイクルで最大1つのクレジットがリターンされ得る。

20

## 【0166】

トランザクションデータチャンネル (TDC) は、トランザクション属性チャンネル (TAC) 上で開始されるトランザクションに対して、マスターからスレーブへのデータ転送を実施するために用いられる、クレジットされないチャンネルである。TDC は、TAC とは独立してマスターされる。これらの2つのチャンネルは、TAC に存在する `cid` [11: 0] 信号、及び TDC に存在する `did` [11: 0] 信号を介してリンクされる。データ転送は、1つ又は複数のデータフェーズに分割される。TDC 上の信号は、各データフェーズに対して正確に1つのサイクルの間、マスターによってアサートされ、同じサイクルにおいてスレーブによってアクセプトされなければならない。

30

## 【0167】

TDC 上のデータ転送内の個々のデータフェーズは、厳密に線形順に成される必要があるが、連続したサイクルで成される (例えば、バーストする) 必要はない。データフェーズの間の空サイクルが、TDC 上で許可される。単一のトランザクションの最終データフェーズは、`dlast` 信号のアサートによって示される。TDC の全体的な利用を改善するために、データ転送内のデータフェーズが厳密に線形順に生じる限り、マスターは異なるトランザクションからのデータフェーズをインターリーブし得る。データウィンドウのサイズ、及び従って `ddata` 信号は、64、128、256、512 ビット幅に構成可能である。デフォルトは512ビットである。各データフェーズは、`dbyten` [CFG: 0] 信号によって指示されるように、データウィンドウの全体か又は一部のいずれかを転送し得る。各インタフェースに対して、全ての TDC 信号が必要とされるわけではない。表 12 は、トランザクションデータチャンネルの信号を列挙する。

40

【表 1 2】

信号	サイズビット	説明	必要／任意
dvalid	1	有効なデータの存在を示す	必要
did	12	現在のデータ転送に対するトランザクションID	必要
dtype	1	データ転送がコマンド又は応答に対応するかを示す	必要
dlast	1	現在のデータ転送の最終データフェーズを示す	必要
ddata	CFG (512)	現在のフェーズのデータ	必要
dword	CFG (1)	現在のフェーズの総データ転送におけるデータウィンドウを示す	必要
dmstid	12	現在のデータフェーズのマスターID	任意
dbyten	CFG (64)	現在のサイクルにおけるデータウィンドウ全体に対する有効なデータを示す	任意
dparity	CFG (16)	データウィンドウ全体に対して32ビットデータ毎にパリティベクトルが生じる	任意
dedc	CFG (20)	256ビット精度でコンピュータされた、EDCハミング及びパリティデータフィールド。複数の10ビットフィールド、256-bit データ毎に10ビット	任意
dsband	CFG	ユーザー定義サイドバンド信号	任意
dstatus	2	データフェーズ毎の応答成功／失敗	任意

10

20

## 【0168】

表12のサイズの列では、CFGは、この信号が構成可能であることを示す。括弧内の数字は、ビットでのデフォルトサイズである。dword、dparity、及びdedc信号のサイズは、ddata信号のサイズから導き出される。dword信号の幅は、構成されたデータウィンドウサイズ、及び128バイト(1024ビット)の最大データ転送の両方に基づいて決定される。最大データ転送サイズは、データウィンドウのサイズに関係なく、128バイトで固定される。

30

## 【0169】

dvalid信号は、トランザクション属性チャンネル(TAC)上の同じサイクルか又は前のサイクルのいずれかにおいて開始されたトランザクションに対する、TDC上の有効なデータフェーズの存在を識別する。0のdvalid信号は、現サイクルにデータがないことを示す。1のdvalid信号は、現サイクルが有効データを含むことを示す。

## 【0170】

did[11:0]信号は、現在のデータ転送に対応するTACトランザクションを一意的に識別する。異なるデータ転送に対するデータフェーズがインターリーブされ得るが、did信号はデータ転送における全てのデータフェーズに対して同じでなければならない。

40

## 【0171】

dtype信号は、データ転送が、コマンドに対応するか又は応答トランザクションに対応するかを識別する。この情報は、同じマスターがdmstid[11:0]及びdid[11:0]の両方を共有する、コマンド及び応答データ転送を一意的に区別するために必要である。0のdtype信号は、現トランザクションがコマンドトランザクションであることを示す。1のdtype信号は、現トランザクションが応答トランザクションであることを示す。

50

## 【0172】

d l a s t 信号は、現在のデータ転送が、現在のバストランザクションの最後のデータであるか否かを識別する。現在のバストランザクションのデータは、データバスの利用可能なライン上で送信するには大き過ぎる可能性がある。この場合、データは、複数のサイクルにおいて転送される。0のd l a s t 信号は、現トランザクションが最後のデータ転送ではないことを示す。1のd l a s t 信号は、現トランザクションが最後のデータ転送であることを示す。

## 【0173】

d s t a t u s [ 1 : 0 ] は、現在のデータ転送に対するデータ状態を識別する。この信号は、応答トランザクションに対してのみ適用可能であり、書き込みコマンドトランザクションに対して無視される。この信号は、データ転送における各データフェーズに対して別個にアサートされる。この信号は、データ転送における各データフェーズに対して同じである必要はない。これにより、各データフェーズに対して別個の状態をリターンするために、細分化された読み出しリターンが可能となる。d s t a t u s 信号は、表9に列挙されたc s t a t u s 信号と同じコーディングを有する。

10

## 【0174】

d b y t e n [ C F G : 0 ] 信号は、現在のデータフェーズに対するd d a t a [ C F G : 0 ] ウィンドウにおける有効バイトレーンを識別する。d d a t a [ C F G : 0 ] バス上の全ての有効バイトは、d b y t e n ビットベクトルの対応するビットにおいて1によって識別される。全てゼロのd b y t e n は、現在のデータフェーズにおいて有効データが転送されていないことを示す。d b y t e n 信号は、書き込みコマンドトランザクションに対する書き込みb y t e n か、又は読み出しリターン/スヌープリターントランザクションに対する読み出しb y t e n のいずれかとして働き得、d v a l i d がアサートされた全てのサイクルに対して有効である必要がある。d b y t e n 信号が、連続する1の間にゼロを有する、ギャッピー (gappy) データフェーズが、書き込みコマンドトランザクションに対してサポートされる。これは、キャッシュトランザクションに対する、書き込みバッファフラッシュ及び部分的ライン書き込みを意図している。d b y t e n 信号はオプションである。d b y t e n 信号が存在しない場合、全てのバイトレーンは、データ転送における各データフェーズに対して有効であると仮定される。

20

## 【0175】

d d a t a [ C F G : 0 ] 信号は、現在のデータフェーズに対するデータを識別する。データバイトは、アドレスによって固定バイトレーンに整合させられ、全てのデータが、d b y t e n [ C F G : 0 ] ビットベクトルによってクオリファイされる。非整合データ又はギャッピーデータ (例えば、書き込みマージデータ) は、d b y t e n [ C F G : 0 ] ビットベクトルにおける必要なビットを設定することによって示される。

30

## 【0176】

d w o r d [ C F G : 0 ] 信号は、現在のデータフェーズが適用するデータ転送全体におけるデータウィンドウを識別する。データウィンドウのサイズより大きいデータ転送は、複数のデータフェーズを必要とし得る。d w o r d 信号は、現在のd d a t a [ C F G : 0 ] 及びd b y t e n [ C F G : 0 ] 信号が適用するデータウィンドウを識別するために用いられる。データ転送は線形順に成される必要がある。従って、d w o r d 信号は、データ転送を通して単調増加される必要がある。d w o r d 信号は、データ転送における継続的なデータフェーズにおいて値を減少させることができない。マスターからの細分化された読み出しリターンが、より幅の狭いダウンストリームデータバスに遭遇すると、その結果、d w o r d は、データウィンドウの境界に達するまで、複数のデータフェーズに対して一貫した値を持つ。d w o r d 信号は、現在のデータフェーズにおけるどのデータバイトがデータ転送全体におけるどのデータウィンドウに対応するのかを一意的に区別することができないので、データウィンドウにわたるデータフェーズは許可されない。

40

## 【0177】

d m s t i d [ 1 1 : 0 ] 信号は、このデータ転送が所属するトランザクションを開始

50

したマスターを一意的に識別する。d m s t i d は、トランザクション属性チャンネル ( T A C ) 上の元のトランザクションの c m s t i d にマッチングしなければならない。

【 0 1 7 8 】

d l a s t 信号は、所与の d i d [ 1 1 : 0 ] に対するデータ転送の最終データフェーズを識別する。

【 0 1 7 9 】

d p a r i t y [ C F G : 0 ] 信号は、現在の d d a t a [ C F G : 0 ] ウィンドウに対するパリティを識別する。パリティは、32ビットサブライン毎に、d d a t a の X O R リダクションとしてコンピュータされる。表 1 3 は、最大 5 1 2 ビットのデータウィンドウに対する d p a r i t y 信号のエンコーディングを列挙する。

10

【表 1 3】

dparity サブフィールド	注記
dparity[15]	ddata[511:480]のXORリダクション
dparity[14]	ddata[479:448]のXORリダクション
dparity[13]	ddata[447:416]のXORリダクション
dparity[12]	ddata[415:384]のXORリダクション
dparity[11]	ddata[383:352]のXORリダクション
dparity[10]	ddata[351:320]のXORリダクション
dparity[9]	ddata[319:288]のXORリダクション
dparity[8]	ddata[287:256]のXORリダクション
dparity[7]	ddata[255:224]のXORリダクション
dparity[6]	ddata[223:192]のXORリダクション
dparity[5]	ddata[191:160]のXORリダクション
dparity[4]	ddata[159:128]のXORリダクション
dparity[3]	ddata[127:96]のXORリダクション
dparity[2]	ddata[95:64]のXORリダクション
dparity[1]	ddata[63:32]のXORリダクション
dparity[0]	ddata[31:0]のXORリダクション

20

30

【 0 1 8 0 】

細分化されたデータ転送は、複数のデータフェーズを介して、全データウィンドウを送り得る。細分化されたデータ転送における各データフェーズについて、d b y t e n [ C F G : 0 ] は、転送されている総データウィンドウのサブセットを特定し得る。各データフェーズは、全データウィンドウ上で d p a r i t y を計算しなければならず、ゼロにされた d b y t e n ビットに対応するデータバイトをゼロ設定する。これにより、スレーブが、各データフェーズからの中間 d p a r i t y 値の継続的な X O R を実施することによって、全データウィンドウに対する正しい d p a r i t y 値を再構成することが可能になる。

40

【 0 1 8 1 】

d e d c [ C F G : 0 ] 信号は、現在のデータフェーズに対するエラー検出 / 訂正ハミング及びパリティ情報を識別する。d e d c 信号は、それぞれ 2 5 6 ビットデータ精度で計算された、2つの 1 0 ビットフィールドで構成される。各 1 0 ビットフィールドは、9 ビットハミングシンドローム及び単一の全パリティビットで構成される。表 1 4 は、d e d c 信号のエンコーディングを列挙する。

【表 1 4】

dedc サブフィールド	説明	注記
dedc[19]	上位全パリティ	{ddata[511:256], dedc[18:10]}のXORリダクション
dedc[18:10]	上位ハミング	ddata[511:256]からコンピュータされる
dedc[9]	下位全パリティ	{ddata[255:0], dedc[8:0]}のXORリダクション
dedc[8:0]	下位ハミング	ddata[255:0]からコンピュータされる

10

dedc 信号は、データウィンドウが 256 ビットより小さく構成されるインタフェース上では用いることができない。

## 【0182】

図 19 は、dedc 信号のエンコーディングのためのロジックを図示する。データ 1910 は、データ [511:256] 及びデータ [255:0] の 2 つの部分に分けられる。ハミング生成器 1931 は、既知の技術に従って、データ 1910 のデータ [511:256] を受け取り、ハミングデータの 9 ビットを形成する。この 9 ビットは、データ 1920 の dedc [18:10] となる。パリティ XOR リダクション 1932 は、データビットデータ [511:256]、及びハミング生成器 1931 からのハミングデータの 9 ビットの組み合わせ排他的 OR を形成する。この単一の XOR ビットは、データ 1920 の dedc [19] になる。ハミング生成器 1933 は、既知の技術に従って、データ 1910 のデータ [255:0] を受け取り、ハミングデータの 9 ビットを形成する。この 9 ビットは、データ 1920 の dedc [8:0] となる。パリティ XOR リダクション 1934 は、データビットデータ [255:0]、及びハミング生成器 1933 からのハミングデータの 9 ビットの組み合わせ排他的 OR を形成する。この単一の XOR ビットは、データ 1920 の dedc [9] となる。

20

## 【0183】

細分化されたデータ転送は、複数のデータフェーズを介して、全データウィンドウを転送し得る。細分化されたデータ転送における各データフェーズに対して、dbyten [CFG:0] は、転送されている総データウィンドウのサブセットを特定し得る。各データフェーズは、全データウィンドウ上で dedc を計算しなければならず、ゼロにされた dbyten ビットに対応するデータバイトをゼロ設定する。これによって、スレーブが、各データフェーズから中間の dedc 値の継続的な XOR を実施することによって、全データウィンドウに対する正しい dedc 値を再構成することが可能になる。

30

## 【0184】

dsband [CFG:0] 信号は、ユーザー定義サイドバンド信号である。この信号はオプションであり、その解釈は実装固有である。dsband 信号は、トランザクションデータチャネル (TDC) によってのみ用いられる。

## 【0185】

書き込み応答チャネル (WRC) は、ペアにされたインタフェーストポロジーに対する反対側 M B A インタフェース上の、前に開始された書き込みコマンドトランザクションに対して、スレーブに書き込み状態をリターンするためにマスターによって用いられるクレジットされたチャネルである。WRC 上の書き込み応答の開始は、クレジットの使用を必要とする。WRC 信号は、正確に 1 サイクルの間マスターによってアサートされ、それらがアサートされたサイクルにおいて、スレーブによってアクセプトされなければならない。各書き込み状態は、元の書き込みコマンドトランザクションの cid [11:0] にマッチングする wid [11:0] 信号によって一意的に識別される。書き込み応答が成功裏に処理されると、wreturn 信号を用いて、スレーブによって書き込み応答クレジットがマスターにリターンされなければならない。書き込み応答に対する必要がない場合、チャネル全体が除外され得るが、チャネルが存在する場合、WRC 上の全ての信号が必

40

50

要とされる。

【0186】

表15は、書き込み応答チャンネル信号を列挙する。

【表15】

信号	サイズ	使用	説明	ドライバ
wvalid	1	書き込み応答	書き込み応答の存在を示す	マスター
wid	12	書き込み応答	専用の書き込み応答インタフェースに対するループバックID	マスター
wstatus	2	書き込み応答	専用の書き込み応答インタフェースに対する書き込み状態	マスター
wnstid	12	書き込み応答	書き込み応答に対するループバックマスターID	マスター
wreturn	1	クレジットリターン	クレジットリターンストローブ	スレーブ

10

【0187】

wmstid[11:0]信号は、WRC上の唯一のオプションの信号である。インタフェースが書き込みコマンドトランザクションをサポートするか否かに応じて、書き込み応答チャンネル全体が存在するように又は存在しないように構成され得る。

20

【0188】

wvalid信号は、有効な書き込み応答の存在を示す。0は書き込み応答が存在しないことを示す。1は現在のサイクルに対する有効な書き込み応答を示す。

【0189】

wid[11:0]信号は、これがそれに対する書き込み応答である、元のTACトランザクションを一意的に識別する。widは、元の書き込みコマンドトランザクションのcidにマッチングしなければならない。

【0190】

wstatus[1:0]信号は、書き込みトランザクションの状態を識別する。wstatus信号は、表9に列挙されたcstatusと同じコーディングを有する。

30

【0191】

wmstid[11:0]信号は、これがそれに対する応答である、書き込みコマンドトランザクションを發したシステムマスターを識別する。wmstidは、書き込みコマンドトランザクションの元のcmstidにマッチングしなければならない。

【0192】

wreturn信号は、書き込み応答クレジットのリターンを示す。0は書き込み応答クレジットリターンが存在しないことを示す。1は現在のサイクルに対する有効な書き込み応答クレジットリターンを示す。

40

【0193】

MBAプロトコルは、複数のトランザクションタイプを定義する。トランザクションは、コマンドと応答の2つの基本的なカテゴリにグループ分けされる。両方のトランザクションカテゴリが、プロトコルによってサポートされる全てのトランザクションの説明と共に以下に記載される。

【0194】

コマンドトランザクションが、データの転送を伴う可能性もあり、伴わない可能性もある、2つのエージェント間のトランザクションシーケンスを開始する。シーケンスは、次の2つの方式の1つにおいて完了する。1つは、反対側MBAインタフェースの書き込み応答チャンネル(WRC)上で他方のエージェントによって開始される書き込み応答であり

50

、もう1つは、反対側MBAインタフェースのトランザクション属性チャンネル(TAC)上で他方のエージェントによって開始される応答トランザクション、及びそのトランザクションデータチャンネル(TDC)上での関連するデータ転送である。全てのコマンドトランザクションは、一意のcid[11:0]を特定しなければならない。cidは、同じインタフェース上の同じマスターによって開始される如何なるアクティブシーケンスにおいても使用されない可能性がある。書き込みコマンドトランザクションは、マスターからスレーブへのデータの転送を必要とする。書き込みコマンドシーケンス以外の全てのシーケンスは、反対側MBAインタフェース上で他方のエージェントによって開始される応答トランザクションによって完了する。

【0195】

コマンドトランザクションは、デマンド、スヌープ、メッセージ、及びメンテナンスの4個のカテゴリに分類される。表16は、デマンドカテゴリにおけるコマンドトランザクションタイプを列挙する。表16における全てのコマンドは、ACE等価トランザクションに対応する。マスターの列は、各トランザクションタイプに対して予期される開始側を示す。シーケンス完了の列は、コマンドトランザクションによって開始されたシーケンスを完了させるための必要とされる応答を示す。

【表16】

トランザクションタイプ	説明	マスター	シーケンス完了
ReadNoSnoop	非共有メモリ領域への読み出し	コアサイド、 ファーサイド	読み出し応答 (TAC)
ReadOnce	キャッシュに割り当てない読み出し	コアサイド、 ファーサイド	読み出し応答 (TAC)
ReadClean	共有状態において割り当てる読み出し	コアサイド	読み出し応答 (TAC)
ReadUnique	一意状態において割り当てる読み出し	コアサイド	読み出し応答 (TAC)
WriteNoSnoop	非共有メモリ領域への書き込み	コアサイド、 ファーサイド	書き込み応答 (WRC)
WriteUnique	部分的ライン書き込み	コアサイド、 ファーサイド	書き込み応答 (WRC)
WriteLineUnique	全ライン書き込み	コアサイド、 ファーサイド	書き込み応答 (WRC)
WriteClean	次のレベルのメモリまでのビクティム書き込みフラッシュ。ローカルキャッシュが保持され、クリーンに変わる。	コアサイド	書き込み応答 (WRC)
WriteBack	次のレベルのメモリまでのビクティム書き込みフラッシュ。ローカルキャッシュコピーが削除される。	コアサイド	書き込み応答 (WRC)

【0196】

ReadNoSnoopコマンドは、非コヒーレント読み出し演算である。この読み出

10

20

30

40

50

し演算は、コヒーレンスプロトコルに参加せず、スヌープトランザクションを生成しない。ReadNoSnoopトランザクションは、`cmemtype[1:0]`、`ccinner[1:0]`、及び`ccouter[1:0]`信号の値に応じて、キャッシュ割り当てとなる可能性とならない可能性がある。

【0197】

ReadOnceコマンドは、開始側マスターにおけるキャッシュ内に割り当てないコヒーレント読み出し演算である。

【0198】

ReadCleanコマンドは、開始側マスターにおけるキャッシュ内に割り当てるコヒーレント読み出し演算である。開始側マスターは、共有状態においてラインを要求するが、ファーサイド構成要素は排他的状態においてアクセスをgrantし得る。

10

【0199】

ReadUniqueコマンドは、開始側マスターにおけるキャッシュ内に割り当てるコヒーレント読み出し演算である。開始側マスターは、排他的状態においてラインを要求し、ラインは排他的状態においてgrantされなければならない。

【0200】

WriteNoSnoopコマンドは、非コヒーレント書き込み演算である。この書き込み演算は、コヒーレンスプロトコルに参加せず、スヌープトランザクションを生成しない。

【0201】

WriteUniqueコマンドは、全キャッシュラインより小さい可能性のあるサイズを有するコヒーレント書き込み演算である。トランザクションアドレスに対するキャッシュラインは、開始側マスターにおけるキャッシュに存在してはならない。

20

【0202】

WriteLineUniqueコマンドは、全キャッシュラインに等しくなければならないサイズを有するコヒーレント書き込み演算である。このトランザクションアドレスに対するキャッシュラインは、開始側マスターのキャッシュに存在してはならない。

【0203】

WriteCleanコマンドは、開始側マスターのキャッシュにおいてラインがクリーン状態で保持される、場合によってはコヒーレントなビクティム書き込みトランザクションである。

30

【0204】

ライトバックコマンドは、開始側マスターにおけるキャッシュにおいてラインが保持されない、場合によってはコヒーレントなビクティム書き込みトランザクションである。

【0205】

表17は、スヌープカテゴリにおけるコマンドトランザクションタイプを列挙する。ACE等価トランザクションは、`SnoopReadOnce`、`ReadOnce`、`SnoopReadShared`、`ReadShared`、`SnoopCleanShared`、`CleanShared`、`SnoopCleanInvalid`、`CleanInvalid`、及び`SnoopMakeInvalid`、`MakeInvalid`である。

40

【表 17】

トランザクション タイプ	説明	マスター	シーケンス完了
SnoopReadOnce	状態変化なしにローカル キャッシュから読み出す	ファーサイド	スヌープ応答 (TAC)
SnoopReadShared	ローカルキャッシュから 読み出し、状態を共有へ変 更する	ファーサイド	スヌープ応答 (TAC)
SnoopCleanShared	ローカルキャッシュから 読み出し、状態をクリーン に変更する	ファーサイド	スヌープ応答 (TAC)
SnoopCleanInvalid	ローカルキャッシュから 読み出し、クリーンにし、 状態を無効に変更する	ファーサイド	スヌープ応答 (TAC)
SnoopMakeInvalid	ローカルキャッシュから 除去し、データ読み出しが リターンされない	ファーサイド	スヌープ応答 (TAC)

10

20

## 【0206】

SnoopReadOnce コマンドは、スヌープされたキャッシュの状態を変更しない、非キャッシング読み出しスヌープコマンドである。スヌープされたキャッシュにおいてラインが存在する場合、データはリターンされなければならない。

## 【0207】

SnoopReadShared コマンドは、完了の際、スヌープされたキャッシュラインを共有状態に移動させることを必要とする、キャッシング読み出しスヌープコマンドである。スヌープされたキャッシュにラインが存在する場合、データは、スヌープ応答トランザクションと共にリターンされなければならない。

30

## 【0208】

SnoopCleanShared コマンドは、完了の際、スヌープされたキャッシュラインをクリーン状態に移動させることを必要とする、キャッシング読み出しスヌープコマンドである。スヌープされたキャッシュにおいてラインが存在する場合、データは、スヌープ応答トランザクションと共にリターンされなければならない。

## 【0209】

SnoopCleanInvalid コマンドは、完了の際、スヌープされたキャッシュラインを無効な状態に移動することを必要とする、キャッシング読み出しスヌープコマンドである。スヌープされたキャッシュにおいてラインが存在する場合、データは、スヌープ応答トランザクションと共にリターンされなければならない。

40

## 【0210】

SnoopMakeInvalid コマンドは、完了の際、スヌープされたキャッシュラインを無効な状態に移動することを必要とする、キャッシングスヌープコマンドである。データは、スヌープ応答トランザクションと共にリターンされることはない。

## 【0211】

表 18 は、メッセージカテゴリにおけるコマンドトランザクションタイプを列挙する。DVM Operatation コマンド、DVM Sync コマンド、及び DVM Completer コマンドは、ACE 等価トランザクションを有する。Cache Warm コマンド、Cache Warm Notify コマンド、及び Evict コマンドは、A

50

C E 等価を有さない。メッセージコマンドトランザクションは、データの転送を含むことはなく、従って、トランザクションデータチャネル ( T D C ) の使用を必要としない。

【表 1 8】

トランザクション タイプ	説明	マスター	シーケンス完了
DVMOperation	一般の D V M 演算	コアサイド、 ファーサイド	メッセージ応答 ( T A C )
DVMSync	DVMSync は全ての下位レベル構成要素に、それ自体の DVMComplete を発行する前に、DVMComplete を送ることを要求する	コアサイド、 ファーサイド	メッセージ応答 ( T A C )
DVMComplete	D V M トランザクション完了	コアサイド、 ファーサイド	メッセージ応答 ( T A C )
CacheWarm	コアサイドに ReadClean コマンドを開始することを要求するファーサイドからのメッセージ	ファーサイド	メッセージ応答 ( T A C )
CacheWarmNotify	コアサイドに ReadClean コマンドを開始すること、及び割り当ての完了の際、後続の WriteSnoop を送ることを要求するファーサイドからのメッセージ	ファーサイド	メッセージ応答 ( T A C )
Evict	ローカルキャッシュがフラッシュなしにデータを削除する	コアサイド	メッセージ応答 ( T A C )

10

20

30

## 【 0 2 1 2 】

C a c h e W a r m コマンドは、そのキャッシュ内に所与のアドレスを割り当てるように R e a d C l e a n 演算の割り当てを開始することをスレーブエージェントに要求するメッセージである。M M U からのみ決定され得るメモリタイプ及びメモリ属性に対応するトランザクション属性チャネル ( T A C ) 属性は、C a c h e W a r m トランザクションから後続の R e a d C l e a n 演算にミラーリングされなければならない。C a c h e W a r m トランザクションは、キャッシュラインをプリフェッチするために、システムによって用いられる。メッセージ応答トランザクションは、R e a d C l e a n 演算が開始された後、開始され得る。

40

## 【 0 2 1 3 】

C a c h e W a r m N o t i f y コマンドは、そのキャッシュ内に所与のアドレスを割り当てるように R e a d C l e a n 演算の割り当てを開始することをスレーブエージェントに要求するメッセージである。M M U からのみ決定され得るメモリタイプ及びメモリ属性に対応するトランザクション属性チャネル ( T A C ) 属性は、C a c h e W a r m トランザクションから後続の R e a d C l e a n 演算にミラーリングされなければならない。R e a d C l e a n 演算に対して、読み出し応答トランザクションが受け取られると、スレーブエージェントは、C a c h e W a r m 演算の完了を示すために、W r i t e N o S n o o p メッセージコマンドトランザクションを開始しなければならない。

## 【 0 2 1 4 】

E v i c t コマンドは、所与のアドレスが開始側マスターにおけるキャッシュから除去

50

された（無効化された）ことの、場合によってはコヒーレントなキャッシュ状態通知である。

【0215】

メンテナンスコマンドトランザクションは、メンテナンスドメインによって区別される2つの特徴がある。2つのメンテナンスドメインは、ポイントオブユニフィケーション（POU）とポイントオブコヒーレンス（POC）である。ポイントオブユニフィケーションは、命令とデータキャッシュとが一体化されるポイントである。これは、典型的に第2レベルキャッシュである。ポイントオブコヒーレンスは、所与のコヒーレンスドメインにおける全てのコヒーレントマスターが、変更されたデータをライトバックしなければならないポイントであり、それによって、同じコヒーレンスドメインにおける他のコヒーレントマスターが変更されたデータを見ることができるようになるためのポイントである。これは典型的に最終レベルキャッシュである。

10

【表19】

トランザクションタイプ	説明	マスター	シーケンス完了
CleanSharedPOU	ローカルキャッシュに、ポイントオブユニフィケーションまでダーティラインをフラッシュし、共有状態に変更することを要求する	コアサイド	メンテナンス応答 (TAC)
CleanInvalidPOU	ローカルキャッシュに、ポイントオブユニフィケーションまでダーティラインをフラッシュし、無効化することを要求する	コアサイド	メンテナンス応答 (TAC)
MakeInvalidPOU	ローカルキャッシュに、ポイントオブユニフィケーションまで無効化することを要求する	コアサイド	メンテナンス応答 (TAC)
CleanSharedPOC	ローカルキャッシュに、ポイントオブコヒーレンスまでダーティラインをフラッシュし、共有状態に変更することを要求する	コアサイド	メンテナンス応答 (TAC)
CleanInvalidPOC	ローカルキャッシュに、ポイントオブコヒーレンスまでダーティラインをフラッシュし、無効化することを要求する	コアサイド	メンテナンス応答 (TAC)
MakeInvalidPOC	ローカルキャッシュに、ポイントオブコヒーレンスまで無効化することを要求する	コアサイド	メンテナンス応答 (TAC)

20

30

40

【0216】

CleanSharedPOUコマンドは、キャッシュラインが変更された状態に保たれる場合、コアからスレーブエージェントまでの全てのキャッシュに、ポイントオブユニフィケーションまでの所与のキャッシュラインをライトアウトすることを要求する、キャッシュメンテナンス演算である。CleanSharedPOUトランザクションは、スレーブエージェントとコアとの間の全てのキャッシュに伝播されなければならないが、こ

50

の伝搬は、`SnoopCleanShared` コマンドトランザクションを介して成され得る。

【0217】

`CleanInvalidPOU` コマンドは、キャッシュラインが変更された状態に保たれる場合、コアからスレーブエージェントまでの全てのキャッシュに、ポイントオブユニフィケーションまでの所与のキャッシュラインをライトアウトすることを要求する、キャッシュメンテナンス演算である。また、全てのキャッシュは、キャッシュラインを無効化しなければならない。`CleanInvalidPOU` トランザクションは、コアに近い方のキャッシュに伝播されなければならないが、この伝搬は、`SnoopCleanInvalid` コマンドトランザクションを介して成され得る。

10

【0218】

`MakeInvalidPOU` コマンドは、コアからスレーブエージェントまでの全てのキャッシュに、ポイントオブユニフィケーションまでの所与のキャッシュラインを無効化することを要求する、キャッシュメンテナンス演算である。`MakeInvalidPOU` トランザクションは、コアに近い方のキャッシュに伝搬されなければならないが、この伝搬は、`SnoopMakeInvalid` コマンドトランザクションを介して成され得る。

【0219】

`CleanSharedPOC` コマンドは、キャッシュラインが変更された状態に保たれる場合、コアからスレーブエージェントまでの全てのキャッシュに、ポイントオブコヒーレンスまでの所与のキャッシュラインをライトアウトすることを要求する、キャッシュメンテナンス演算である。`CleanSharedPOC` トランザクションは、スレーブエージェントとコアとの間の全てのキャッシュに伝播されなければならないが、この伝搬は、`SnoopCleanShared` コマンドトランザクションを介して成され得る。

20

【0220】

`CleanInvalidPOC` コマンドは、キャッシュラインが変更された状態に保たれる場合、コアからスレーブエージェントまでの全てのキャッシュに、ポイントオブコヒーレンスまでの所与のキャッシュラインをライトアウトすることを要求する、キャッシュメンテナンス演算である。また、全てのキャッシュは、キャッシュラインを無効化しなければならない。`CleanInvalidPOC` トランザクションは、コアに近い方のキャッシュに伝播されなければならないが、この伝搬は、`SnoopCleanInvalid` コマンドトランザクションを介して成され得る。

30

【0221】

`MakeInvalidPOC` コマンドは、コアからスレーブエージェントまでの全てのキャッシュに、ポイントオブコヒーレンスまでの所与のキャッシュラインを無効化することを要求する、キャッシュメンテナンス演算である。`MakeInvalidPOC` トランザクションは、コアに近い方のキャッシュに伝搬されなければならないが、この伝搬は、`SnoopMakeInvalid` コマンドトランザクションを介して成され得る。

【0222】

応答トランザクションは、2つのエージェント間のトランザクションシーケンスを完了するために用いられる。応答トランザクションは、元のコマンドトランザクションの `cid[11:0]` にマッチングする必要がある。応答トランザクションは、データの転送に関連する可能性も、しない可能性もある。表20は、応答トランザクションタイプを列挙する。`ACE` 等価トランザクションは、読み出し応答が `RESP` であり、スヌープ応答が `CR`、`CU` であり、メッセージ応答及びメンテナンス応答コマンドに対して `ACE` 等価がない。マスターの列は、各トランザクションタイプに対する予期される開始側を示す。

40

【表 2 0】

コマンドタイプ	説明	マスター
ReadResponse	データ及び状態を備える読み出し応答	コアサイド、ファーサイド
SnoopResponse	データ及び状態を備えるスヌープ応答	コアサイド
MessageResponse	メッセージが完了する	コアサイド、ファーサイド
MaintenanceResponse	メンテナンスが完了する	ファーサイド

## 【 0 2 2 3】

10

読み出しシーケンスを完了するために、読み出し応答トランザクションが必要とされる。読み出し応答トランザクションは、常にデータ転送を含む。失敗状態を示す `csatus[1:0]` 値を有する読み出し応答トランザクションは、それでもなお、`cdat a` 信号をアサートし、ゼロにされたデータをリターンしなければならない。成功を示す `csatus[1:0]` 値を有する読み出し応答トランザクションは、トランザクションデータチャンネル (TDC) 上にデータ転送を含まなければならない。

## 【 0 2 2 4】

20

スヌープシーケンスを完了するために、スヌープ応答トランザクションが必要とされる。スヌープ応答トランザクションは、データ転送を含む可能性も、含まない可能性もある。失敗状態を示す `csatus[1:0]` 値を有するスヌープ応答トランザクションは、`cdat a` 信号をディアサートし得、トランザクションデータチャンネル (TDC) の使用を防止し得る。成功を示す `csatus[1:0]` 値を有するスヌープ応答トランザクションは、トランザクションデータチャンネル (TDC) 上にデータ転送を含まなければならない。

## 【 0 2 2 5】

メッセージシーケンスを完了するために、メッセージ応答トランザクションが必要とされる。メッセージ応答トランザクションは、データ転送を含むことはない。

## 【 0 2 2 6】

30

メンテナンスシーケンスを完了するために、メンテナンス応答トランザクションが必要とされる。メンテナンス応答トランザクションは、データ転送を含むことはない。

## 【 0 2 2 7】

応答トランザクションは、トランザクション属性チャンネル (TAC) 上の対応する `csatus[1:0]` 信号をアサートすることによってエラー状態を示す。読み出し応答トランザクションは、読み出し演算を処理するように試みている間、エラーの発生に関係なく、1の `cdat a` 信号をアサートする必要がある。

## 【 0 2 2 8】

40

アドレス及び保護エラーは、典型的に、スレーブがエンドポイント読み出しアクセスを実施する前に検出される。この場合、それでもなお、データ転送が読み出し応答トランザクションと共に成されなければならない。データ転送を伴う全読み出し応答トランザクションは、次のように起こり得る。1の `cdat a` 信号が、トランザクション属性チャンネル (TAC) 上でトランザクションを用いてアサートされなければならない。 `csatus[1:0]` 信号は、発生したエラーがある場合はその種類を示す。これは、アドレスエラー及び保護エラーであり得る。データ転送がトランザクションデータチャンネル (TDC) 上で成されなければならない。データ転送のサイズは、元の読み出し演算の `cby te cnt[6:0]` に等しくなければならない。 `ddat a[CFG:0]` 信号は、セキュリティホールを回避するために、全てゼロに設定されなければならない。このセキュリティホールは、読み出しアクセスの間に、場合によっては読み出し応答トランザクションが開始された後に起こり得る、非コレクタブルビットエラー等のこの信号エラー上で前にレジスタされたデータのサンプリングを可能にし得る。1つ又は複数のデータフェーズが転送された後にエラーが発生する場合、データ転送は完了しなければならない、エラーのある

50

データフェーズに対して `data [CFG : 0]` を全てゼロに設定することに加え、`status [1 : 0]` 信号上の対応するエラー種類をアサートすることによって、エラーのあるデータフェーズが示されなければならない。

#### 【0229】

全てのMBAトランザクションは、クレジットの使用を介して動作する。トランザクション属性チャンネル(TAC)上で開始されるトランザクション、及び書き込み応答チャンネル(WRC)上で開始される書き込み応答は、少なくとも1つのクレジットの使用を必要とする。クレジットは、マスターによって所有及び使われ、その後、スレーブによってリターンされる。書き込み応答クレジットは、1種類のみを有し、より詳細な分類は必要ない。トランザクションクレジットは、複数のクレジットタイプに組織化される。例示の実施形態の多くの実装は、クレジットタイプをスレーブ上の物理リソースによって分類する。

10

#### 【0230】

全てのトランザクションでは、`ccredit [CFG : 0]` 信号は、使われているクレジットのタイプを示す。例示の実施形態は、クレジットタイプの固定数を定義する必要がなく、クレジットタイプの固定された意味も定義する必要がない。特定の実装によって、MBAインタフェース上の利用可能なクレジットタイプの数、並びに各クレジットタイプに関連する意味が構成される。

#### 【0231】

クレジットされたチャンネルに対するクレジットハンドシェイクは、非常にシンプルであり、クレジットスPENDの後、クレジットリターン、という時間的形式に従う。マスターは、`valid` をアサートすること、及び、`ccredit [CFG : 0]` ビットベクトルにおいて1つ又は複数のビットをアサートすることによって、1つ又は複数のクレジットを使う。対応するスレーブは、同じサイクル上でトランザクションをアクセプトする。これは強制的である。マスターが使うためのクレジットを有する場合、これは、スレーブがトランザクションをアクセプトしなければならないことを示す。スレーブは、トランザクションを完了し、その後、トランザクションによって用いられた1つ又は複数のリソースをフリーにする。リソースをフリーにすると、スレーブは、`creturn [CFG : 0]` ビットベクトルにおける対応するビットをアサートすることによって、クレジットをマスターにリターンする。その後、マスターは後の使用のためにクレジットを有する。

20

30

#### 【0232】

図20は、トランザクション属性チャンネル(TAC)に対するシンプルなクレジットハンドシェイクを図示する。図20において、2001は、マスター及びスレーブを両方制御するクロック信号である。マスターは、トランザクションを開始させるために、1クロックサイクルの間アクティブな`valid` 信号2002を生成する。また、マスターは、同じ1つのクロックサイクルの間に、`ccredit` 信号A2003を生成する。この`ccredit` 信号A2003は、現在のトランザクション上で使われているタイプクレジットを示す。例示の実施形態に従い、マスターが使うためのクレジットを有する場合、スレーブは、トランザクションをアクセプトしなければならない。このように、スレーブは、トランザクションをアクセプトし、不確定時間2010を演算する。アクセプトされたトランザクションによって必要とされる処理が完了すると、スレーブは、`creturn` 信号A2004を介してクレジットをリターンする。この`creturn` 信号A2004は、好ましくは、1クロックサイクルの長さを有する。

40

#### 【0233】

例示の実施形態において、クレジットされたチャンネル上でのクレジットリターンアクションに対して、クレジットスPENDの1対1のマッピングがある。同時期に同じチャンネル上で進行中の複数のクレジットハンドシェイクがあり得る。マスターは、所有するクレジットの数を常に把握する責任を持つ。各クレジットスPENDアクションが、マスターにおけるクレジットカウントを1だけ減少させる。各クレジットリターンアクションが、マスターにおけるクレジットカウントを1だけ増加させる。

50

## 【0234】

クレジットスPEND及びクレジットリターンアクションは、同じ又は異なるクレジットタイプに対する異なるハンドシェイクに対して、クレジットされた同じチャンネル上でインターリーブされ得る。

## 【0235】

図21は、同じチャンネル上のインターリーブされるクレジットハンドシェイクを図示する。図21において、2101は、マスター及びスレーブの両方を制御するクロック信号である。マスターは、第1のトランザクションを開始するために、1クロックサイクル2111の間、アクティブのc v a l i d信号2102を生成する。また、マスターは、同じクロックサイクル2111の間、c c r e d i t信号A2103を生成する。このc c r e d i t信号A2103は、現在のトランザクション上で使われているクレジットのタイプを示す。2クロックサイクル2112の間、マスターは、c c r e d i t信号B2103を送信し、その後c c r e d i t信号A2103が続く。これらのc c r e d i t信号は、2クロックサイクルのc v a l i d信号2102の期間内にある。マスターが、スレーブからのクレジットリターンの前に、第2のc c r e d i t信号A2103を送信したことに留意されたい。これは、マスターが元々、タイプAクレジットを少なくとも2つ有していた場合にのみ起こる。これは、スレーブが、第1のそのようなトランザクションを完了する前に、タイプAの第2の要求をアクセプトするための容量を有していることを必要とする。スレーブは、このようにして、第1のトランザクションタイプAをアクセプトする。第1のアクセプトされたトランザクションによって要求される処理が完了すると、スレーブは、時間2113の間に、c r e t u r n信号A2104を介してクレジットをリターンする。このc r e t u r n信号A2104は、好ましくは1クロックサイクルの長さを有する。第2のアクセプトされたトランザクション(タイプB)及び第3のアクセプトされたトランザクション(タイプA)によって要求される処理が完了すると、スレーブは、時間2114の間に、c r e t u r n信号A+B2104を介して、2つのクレジットをリターンする。好ましい実施形態において、クレジットリターンは、チャンネル上で用いられる各クレジットタイプに対して1ビットを備えるビットベクトルとしてエンコードされ得る。タイプA及びタイプBの両方に対してビットを設定することで、単一のクレジットリターン信号が両方のクレジットタイプをリターンすることが可能となる。

## 【0236】

図17は、ペアにされたインタフェースポートロージ(PIT)における各チャンネルを図示する。完全なトランザクションシーケンスを実施するために、エージェントペアが、ミラーリングされたペア構成の2つの別個のMBAインタフェースを用いて通信しなければならない。これは、ペアにされたインタフェースと称される。ペアにされたインタフェースは2つのエージェント間の双方向のデータ転送をサポートする。エージェントAとBの間の全てのクレジットされたシーケンスは、クレジットされた非書き込みシーケンス、クレジットされた書き込みシーケンス、及びクレジットされたDVMシーケンス、の3つの形式の1つに従う。

## 【0237】

図22は、クレジット非書き込みシーケンスにおけるステップを図示する。そのようなクレジット非書き込みシーケンスは、コアサイドエージェント1710がTAC 1712上でタイプA0のクレジットスPENDを始めること(ステップ2201)で開始する。ステップ2202で、ファーサイドエージェント1720が、タイプA0のクレジットリターンをTAC 1722上でコアサイドエージェント1710に送信する。これは、ファーサイドエージェント1720が初期コマンドトランザクション(ステップ2201)によって占有された物理リソースをクリアにすると成される。クレジットリターンに関連するトランザクションIDはない。ステップ2203で、ファーサイドエージェント1720は、新しい応答トランザクションを開始し、TAC 1722上でクレジットタイプB1の単一のクレジットを使う。この応答トランザクションのIDは、元のID(ステップ2201)にマッチングしなければならない。ステップ2204で、ファーサイドエー

10

20

30

40

50

ジェント 1720 は、TDC 1723 上で、1つ又は複数のデータフェーズのオプションのデータ転送を実施する。データ転送がステップ 2203 におけるクレジットスPENDに該当するので、このアクションに対してクレジットは必要とされない。TAC 1722 と TDC 1723 が異なるチャネルなので、TDC 1723 上でのデータ転送の初期フェーズが、ステップ 2203 におけるクレジットスPENDと同時に起こり得る。図 22 は、複数のデータ転送を有する例を図示する。ステップ 2205 において、コアサイドエージェント 1710 は、リターンクレジットタイプ B の単一のクレジットを、TAC 1712 上でファーストエージェント 1720 にリターンする。これは、コアサイド 1710 が、リターントランザクションによって占有された物理リソースをクリアする（ステップ 2203）と起こる。クレジットリターンに関連するトランザクション ID はない。

10

【0238】

読み出し演算、スヌープ演算、メッセージシーケンス、及びメンテナンスシーケンスの5つのクレジットされた非書き込みシーケンスタイプがある。表 21 は、ペアにされたインタフェースポートロジにおける、図 11 に図示されたステップに対する適法なトランザクションタイプ及びデータ転送を列挙する。

【0239】

【表 21】

	読み出しシーケンス	スヌープシーケンス	メッセージシーケンス	メンテナンスシーケンス
ステップ 1	ReadNoSnoop ReadOnce ReadClean ReadUnique	SnoopReadOnce SnoopReadShared SnoopCleanShared SnoopCleanInvalid SnoopMakeInvalid	CacheWarm CacheWarmNotify Evict	CleanSharedPOU CleanInvalidPOU MakeInvalidPOU CleanSharedPOC CleanInvalidPOC MakeInvalidPOC
ステップ 2	必要	必要	必要	必要
ステップ 3	ReadResponse	SnoopResponse	MessageResponse	MaintenanceResponse
ステップ 4	cstatus が成功を示す場合は必要であり、cstatus が失敗を示す場合は任意である	スヌープトランザクションと、キャッシュ内のスヌープされたラインの存在とに依存して任意である	許可されていない	許可されていない
ステップ 5	必要	必要	必要	必要

20

30

【0240】

図 23 は、クレジットされた書き込みコマンドシーケンスの一般的な形式を図示する。クレジットされた書き込みシーケンスは、シーケンスが完了する様式によって、上述のクレジットされた非書き込みシーケンスから差別化される。クレジットされた書き込みシーケンスは、書き込み応答チャネル(WRC)上での書き込み状態で完了する。トランザクション ID 及びクレジットタイプは、クレジットの性質、及びシーケンスにわたる ID フローを示すために、自由裁量による象徴的な名を用いて示され、それ以外は M B A インタフェース又はプロトコル内で如何なる意味も持たない。

40

【0241】

クレジットされた書き込みシーケンスは、コアサイドエージェント 1710 が TAC 1711 上でタイプ A5 の単一クレジットを使うステップ 2301 で開始する。これは、コアサイド 1710 が、ファーストエージェント 1720 が書き込みのデータを扱えることを確実にしなければならないので、図 22 のステップ 2201 のクレジットタイプ A0 とは異なる。従って、異なるクレジットタイプが必要とされる。ステップ 2301 において、コアサイドエージェント 1710 は、書き込みコマンドトランザクションに対して、TDC 1712 上の1つ又は複数のデータフェーズのデータ転送を実施する。このア

50

クシオンに対してクレジットは必要とされない。このデータ転送は、ステップ2301におけるクレジットスPENDに該当する。TAC 1711とTDC 1712は異なるチャネルであるので、TDC 1712上のデータ転送の初期フェーズが、ステップ2201におけるクレジットスPENDと同時に起こり得る。図22は、複数のデータ転送を有する例を図示する。ステップ2303において、ファーストエージェント1720は、タイプA5の単一クレジットをTAC 1722上でコアサイドエージェント1710にリターンバックする。これは、ファーストエージェント1720が、コマンドトランザクションによって占有された物理リソースをクリアにする(ステップ2301)と起こる。このクレジットリターンに関連するトランザクションIDはない。WRC 1724上で書き込みコマンドトランザクションの処理が完了すると、ステップ2304において、ファーストエージェント1720が書き込み応答を開始する。IDは、元のIDにマッチングしなければならない。書き込み応答は、専用の書き込み応答クレジットタイプの使用を必要とする。ステップ2304において、コアサイドエージェント1710は、WRC 1714を介して、単一の書き込み応答クレジットをファーストエージェント1720にリターンする。これは、ファーストエージェント1710によって送られた書き込み応答がコアサイドエージェント1710によって消費されると成される。書き込み応答クレジットリターンに関連するトランザクションID又は明示的なクレジットタイプはない。

10

#### 【0242】

下記は、例示のクレジットされたDVMシーケンスのステップである。この例は、相互接続として機能するファーストエージェントと、キャッシュマスターとして機能するコアサイドエージェントとの間の挙動を説明する。この例において、コアサイドエージェント1710/キャッシュマスターは、表では説明されていない、システムにおける別のキャッシュマスターからの相互接続に対して開始されたDVM - Operationに回答している。DVMシーケンスには、DVM Operation、DVM sync、及びDVM Completeの3つの部分がある。

20

#### 【0243】

DVM Operationは、ファーストエージェント1720が、新しいDVM - Operationコマンドトランザクションシーケンスを開始することで始まる。これは、適切なタイプの単一クレジットを使うことを含む。コアサイドエージェント1710は、コアサイドエージェント1710が、コマンドトランザクションによって占有された物理リソースをクリアにすると、そのタイプの単一クレジットをファーストエージェント1720にリターンバックすることによって応答する。このクレジットリターンに関連するトランザクションIDはない。コアサイドエージェント1710は、次に、DVM - Operationの受け取りを確認するために、ファーストエージェント1720に対して、クレジットされたトランザクションとしてのメッセージ応答を開始する。コアサイドエージェント1710は、元のDVM - OperationからのIDを用いる。ファーストエージェント1720は、メッセージ応答によって用いられるタイプにマッチングする単一クレジットをリターンする。クレジットリターンに関連するトランザクションIDはない。

30

40

#### 【0244】

その後、ファーストエージェント1720は、新しいDVM - Syncコマンドトランザクションシーケンスを開始し、適切な単一クレジットを使う。この例において、同じマスターからの全てのDVMトランザクションは、同じクレジットタイプを用いる。コアサイドエージェント1710は、対応するタイプA6の単一クレジットを、ファーストエージェント1720にリターンする。これは、コアサイドエージェント1710がコマンドトランザクションによって占有された物理リソースをクリアにすると成される。クレジットリターンに関連するトランザクションIDはない。次に、コアサイドエージェント1710は、DVM - Syncの受け取りを確認するために、ファーストエージェント1720へのメッセージ応答を、クレジットされたトランザクションにおいて開始する。

50

コアサイドエージェント 1710 は、元の DVM - Sync トランザクションからの ID を用いる。ファーサイドエージェント 1720 は、メッセージ応答によって用いられるタイプにマッチングする単一のクレジットタイプをリターンする。このクレジットリターンに関連するトランザクション ID はない。

【0245】

コアサイドエージェント 1710 は、新しい DVM - Complete コマンドトランザクションシーケンスを開始し、適切なタイプの単一クレジットを使う。ファーサイドエージェント 1720 は、そのタイプの単一クレジットをコアサイドエージェント 1710 にリターンバックする。これは、ファーサイドエージェント 1720 がコマンドトランザクションによって占有された物理リソースをクリアにすると成される。クレジットリターンに関連するトランザクション ID はない。ファーサイドエージェント 1720 は、DVM - Complete の受け取りを確認するために、エージェント B への、メッセージ応答のクレジットされたトランザクションを開始する。元の DVM - Sync からの ID が用いられる。コアサイドエージェント 1710 は、メッセージ応答によって用いられるタイプにマッチングする単一クレジットをリターンする。このクレジットリターンに関連するトランザクション ID はない。

10

【0246】

リセットの際、インタフェース上の各クレジットタイプに対するクレジットカウントは、静的インタフェース構成によって決定されるクレジットリミットタイオフ (tieoff) 値にリセットされる。各クレジットタイプに対して、このリミットは、対応するトランザクションをサービスするスレーブリソースに対応する。

20

【0247】

トランザクションは、`address [47:0]`、`cmstid [11:0]`、及び `cid [11:0]` のトランザクションプロパティに基づいて MBA プロトコルにおいて順序付けられている。オプションの `cmstid` 信号を含まない MBA インタフェースでは、順序付け規則の目的で、全てのトランザクションに対して一定の `cmstid` が仮定される。`address` 及び `cbytecnt` によって決定されるように、同じバイトの 1 つ又は複数上で 2 つのトランザクションが動作する場合、トランザクションは、オーバーラップするアドレスを有すると定義される。2 つのエージェント間のクレジットされた非書き込みシーケンスは、図 22 に示されるように順序付けされる。2 つのエージェント間のクレジットされた書き込みシーケンスは、図 23 に示されるように順序付けされる。

30

【0248】

複数のシーケンスが、同じ MBA のペアにされたインタフェース上でインターリーブされ得る (図 17)。単一シーケンスに対する順序付け規則は有効なままであり、別個のシーケンスに対するシーケンスステップは、任意の順にインターリーブされ得る。マスターが、全てがデータ転送を含む複数のトランザクションを送っている場合、マスターは、単一シーケンスの規則が順守される限り、任意の順序で、異なるトランザクション間で、TDC 上でデータフェーズをインターリーブし得る。

【0249】

マスターエージェントに対する順序付け規則は、次のように要約される。データ転送を含む任意のトランザクションに対して、第 1 の TDC DATA フェーズは、同じサイクルにおいて開始され得るが、TAC に先行することはできない。データ転送における全てのデータフェーズは、アドレスに基づいて単調増加順に起こらなければならない。マスターは、任意の単一データ転送内のデータフェーズが、アドレスに基づいて単調増加順に開始される限り、トランザクション属性チャンネル (TAC) 上で発行されるトランザクションに対して任意の順で、トランザクションデータチャンネル (TDC) データフェーズをインターリーブし得る。マスターは、応答トランザクション又は書き込み状態のいずれかが、前の `cmstid / cid` ペアに対して受け取られるまで、同じ `cmstid [11:0]` 及び `cid [11:0]` の組み合わせで、同じ MBA インタフェース上で複数のコマン

40

50

ドトランザクションを開始してはならない。

【0250】

スレーブエージェントに対する順序付け規則は、次のように要約される。スレーブは、同じ `cms tid` とオーバーラップするアドレスに対する全てのブロッキングトランザクションを、`ccredit` か又は `cid` に関係なく、それらが受け取られた順で完了しなければならない。スレーブは、同じ `cms tid` とオーバーラップするアドレスに対する全てのノンブロッキングトランザクションを、`ccredit` か又は `cid` に関係なく、それらが受け取られた順で完了しなければならない。スレーブが、同じ `cms tid` とオーバーラップするアドレスに対するノンブロッキングトランザクションを受け取った後に、ブロッキングトランザクションを受け取る場合、スレーブは、2つのトランザクションを、`ccredit` か又は `cid` に関係なく、それらが受け取られた順で完了しなければならない。ブロッキングトランザクションは、同じ `cms tid` とオーバーラップするアドレスに対するノンブロッキングトランザクションをバイパスしてはならない。スレーブが、同じ `cms tid` とオーバーラップするアドレスに対するブロッキングトランザクションを受け取った後に、ノンブロッキングトランザクションを受け取る場合、スレーブは、それらを受け取った順でトランザクションを完了することを必要としない。ノンブロッキングトランザクションは、同じ `cms tid` とオーバーラップするアドレスに対するブロッキングトランザクションをバイパスすることを許可される。スレーブは、`ccredit`、`cms tid`、又は `cid` にかかわらず、任意の順で、非オーバーラップするアドレスを用いてトランザクションを完了し得る。

10

20

【0251】

コアサイドエージェントは、メモリのシェアラブル領域に対するトランザクションに対する正確性を保証するために、下記の規則に従わなければならない。コアサイドエージェントは、`ccredit`、`cms tid`、又は `cid` に関係なく、オーバーラップするアドレスに対する、スヌープコマンドトランザクション及び読み出し応答トランザクションの順を維持しなければならない。

【0252】

ファーサイドエージェントは、メモリのシェアラブル領域に対するトランザクションの正確性を保証するために、下記の規則に従わなければならない。ファーサイドエージェントが、コアサイドエージェントに対してスヌープコマンドトランザクションを開始し、その後、オーバーラップするアドレスを備えるブロッキングコマンドトランザクションを受け取る場合、ファーサイドエージェントは、コアサイドエージェントからスヌープ応答トランザクションを受け取り、処理するまで、ブロッキングコマンドトランザクションを処理してはならない。

30

【0253】

マスターは、`cms tid / cid` ペアに対して、応答トランザクション又は書き込み状態のいずれかが受け取られるまで、同じ `cms tid [11:0]` 及び `cid [11:0]` の組み合わせを備える同じ M B A インタフェース上で複数のコマンドトランザクションを開始してはならない。

【0254】

マルチコアバスアーキテクチャ (M B A) プロトコルの導入は、全てのコマンドの特徴 (読み出し、書き込み、コヒーレンス、メッセージ)、並びに応答を含む全てのトランザクションタイプに対して、同じ物理チャネルを共有する新規の技術を含む。トランザクション属性チャネル (T A C) 及びトランザクションデータチャネル (T D C) の2つのチャネルは、トランザクションタイプに関係なく、全てのトランザクション属性及びデータを送信するために用いられる。書き込みコマンド、読み出し応答、コヒーレンス応答のどれに対するか等、単一方向におけるデータフローに対する幅に関係なく、単一のデータバスのみが必要とされる。

40

【0255】

図 2 4 は、図 1 7 に図示されるコアサイドエージェント 1 7 1 0 又はファーサイドエー

50

ジェント 1720 等の、エージェント 2400 のブロック図である。図 24 は、MBA における最小の信号ワイヤの使用を図示する。バックエンドプロセス 2401 は、エージェントに対する全ての回路及びシステムを表す。これは、内部キャッシュを備える CPU、メモリ、制御可能な周辺機器等を含み得る。この例において、バックエンドプロセス 2401 は、発信トランザクション及びクレジットリターンに対して使われる、クレジットカウンタ 2402 にストアされたバスクレジットのトラッキング（増加及び減少）を含むバスプロトコルを扱う。

**【0256】**

図 24 はまた、バックエンドプロセス 2401 に接続された読み出し/書き込みメモリ 2403 を図示する。読み出し/書き込みメモリ 2403 は、例示の実施形態において考慮される、制御された演算のタイプを表す。図 24 に図示された例において、バックエンドプロセス 2401 は、バストランザクションに従って、読み出し/書き込みメモリ 2403 から読み出し又は読み出し/書き込みメモリ 2403 に書き込み得る。バックエンドプロセス 2401 は、出力デバイスへ又は入力デバイスからなど、単一方向のデータフローを制御し得る。

10

**【0257】**

バックエンドプロセス 2401 は、この例におけるマスターサイド 2410 及びスレーブサイド 2420 を含む上述の MBA を介して通信する。他の実現可能な構成を更に以下に説明する。

**【0258】**

マスターサイド 2410 は、マスター属性チャンネル 2411、マスターデータチャンネル 2412、及びマスター書き込み応答チャンネル 2413 を含む。エージェント 2400 は、バス 2414 を介して通信する、上述の表 5 に列挙された *valid*、*credit*、*cid*、*ctype*、及び *data* 等の全ての信号上で、マスター属性チャンネル 2411 を用いる。エージェント 2400 は、バス 2415 を介して通信する、上述の表 12 に列挙した *invalid*、*did*、*dtype*、*dlast*、及び *data* などの全ての信号上で、マスターデータチャンネル 2412 を用いる。エージェント 2400 は、バス 2416 を介して通信する、上述の表 15 に列挙した *wvalid*、*wid*、及び *wstatus* 等のすべての信号上で、マスター書き込み応答チャンネル 2413 を用いる。

20

**【0259】**

スレーブサイド 2420 は、スレーブ属性チャンネル 2421、スレーブデータチャンネル 2422、及びスレーブ書き込み応答チャンネル 2423 を含む。エージェント 2400 は、バス 2424 を介して通信する、上述の表 5 に列挙したような *return* 信号上で、スレーブ属性チャンネル 2421 を用いる。エージェント 2400 は、バス 2425 を介して通信する、上述の表 12 に列挙した *invalid*、*did*、*dtype*、*dlast*、及び *data* 等の信号全ての上で、スレーブデータチャンネル 2412 を用いる。エージェント 2400 は、バス 2426 を介して通信する、上述の表 15 に列挙したような *wreturn* 信号上で、スレーブ書き込み応答チャンネル 2423 を用いる。

30

**【0260】**

図 24 は、マスター及びスレーブデバイスの両方として動作するために必要とされる全てのパーツを含むエージェント 2400 を図示する。MBA において、エージェントは、しばしば、両方の役目において動作する。例えば、外部メモリエージェントと通信するエージェントは、マスターとスレーブの両方として機能する必要がある。マスターとして、そのようなエージェントは、読み出し又は書き込みを始めるためにクレジットを使い、読み出し又は書き込み開始アドレス、及びデータ長を供給し、トランザクションが書き込みである場合はデータを供給し、及びクレジットリターンを受け取る必要がある。スレーブとして、そのようなエージェントは、クレジットをアクセプトし、読み出しトランザクションに回答してメモリによって供給されるデータをアクセプトし、及びクレジットをリターンする必要がある。この例は、2方向のデータフローを想定している。この例は、一般的に、図 24 において示したパーツ全てを必要とする。

40

50

## 【0261】

図24のパーツ全てよりも必要とするパーツが少ない他の例もある。出力デバイス等の書き込み専用周辺機器に結合されるエージェントを想定する。そのようなエージェントは、書き込みを始めるためにクレジットを使い、必要に応じて書き込み開始アドレス及びデータ長を供給し、書き込みデータを供給し、及びクレジットリターンを受け取る必要がある。また、そのようなエージェントは、書き込み応答を書き込み専用周辺機器に生成し得る。書き込み専用周辺機器はデータをリターンしないであろうし、そのため、そのようなエージェントは、バストランザクションを制御する必要もないであろうため、スレーブデバイスである必要がないであろう。そのようなエージェントは、マスター属性チャンネル2411、マスターデータチャンネル2412、及びマスター書き込み応答チャンネル2423を必要とし得る。

10

## 【0262】

入力デバイス等の読み出し専用周辺機器に結合されるエージェントの別の例を考える。そのようなエージェントは、読み出しを始めるためにクレジットを使い、クレジットリターンを受け取る必要がある。そのようなエージェントは、クレジットをアクセプトし、読み出しトランザクションに回答してメモリにより供給されるデータをアクセプトし、及びクレジットをリターンするために、スレーブとして動作する必要がある。そのようなエージェントは、マスター属性チャンネル2412、スレーブ属性チャンネル2421、スレーブデータチャンネル2422、及びスレーブ書き込み応答チャンネル2423を必要とし得る。このエージェントは、マスターデータチャンネル2412又はマスター書き込み応答チャンネル2413を必要としない。

20

## 【0263】

他の解決策が、異なるコマンドタイプ間で幾つかの信号を共有し得る（例えば、読み出し及び書き込みが、アドレスピンを共有し得る）。典型的に、これらの他の解決策は、読み出しデータと書き込みデータとを分離する。これは、512ビット及びそれ以上などの大きなデータ幅でのワイヤ輻輳の最大の原因である。MBAプロトコルは、トランザクションタイプに関係なく、同じピンを横切って全てのトランザクションを送信し、その結果、最大のワイヤ共有と最小の物理的オーバーヘッドを達成する。

## 【0264】

MBAプロトコルは、マスターからスレーブへトランザクションを送るためにトランザクション属性チャンネル(TAC)及びトランザクションデータチャンネル(TDC)の2つのチャンネルしか必要としない。TACは、アドレス、メモリ属性、及びトランザクションタイプを含む、全てのトランザクション属性を送信する。TDCは、データを1つ又は複数のフェーズにおいて送信する。全てのトランザクションタイプが同じ2つのチャンネルを用いるので、例示の実施形態は、全範囲の異なるトランザクションタイプを送信するために、最小数の物理的ワイヤしか必要としない。

30

## 【0265】

MBAプロトコルによって提供されるチャンネル共有は、結果として、大きな相互接続に対して物理的ワイヤ輻輳を有意に減少させ、複数のマスターを備えるスレーブデバイスを達成する。これは、面積及び電力を低減させ、性能を向上させ、物理的閉鎖に対する設計サイクルを短縮するという結果をもたらす。

40

【 図 1 】

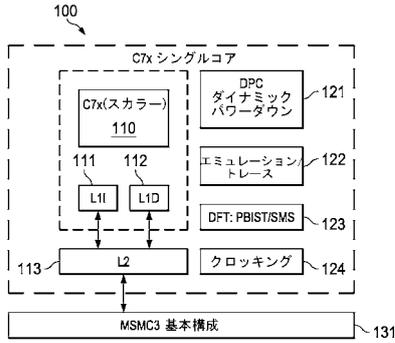


FIG. 1

【 図 3 】

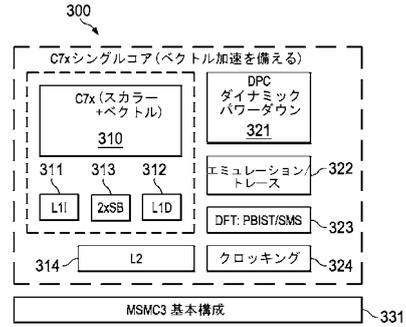


FIG. 3

【 図 2 】

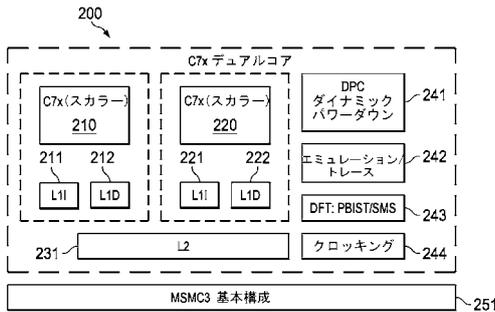


FIG. 2

【 図 4 】

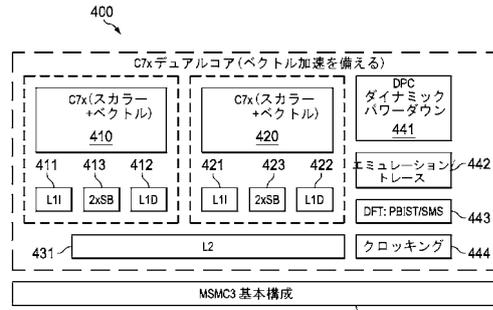


FIG. 4

【 図 5 】

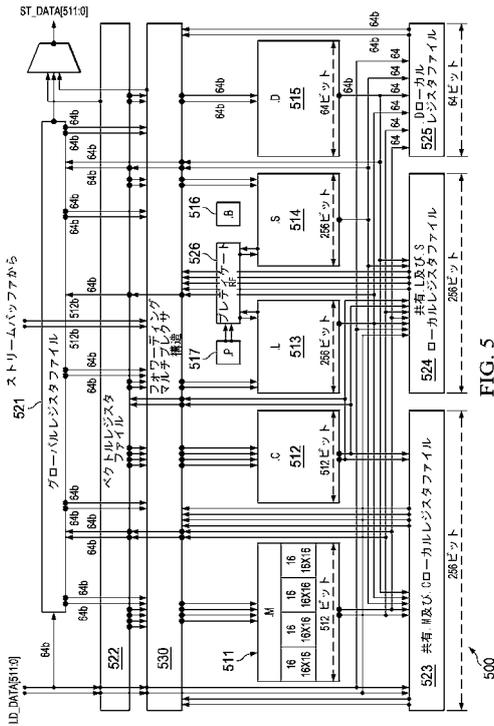


FIG. 5

【 図 6 】

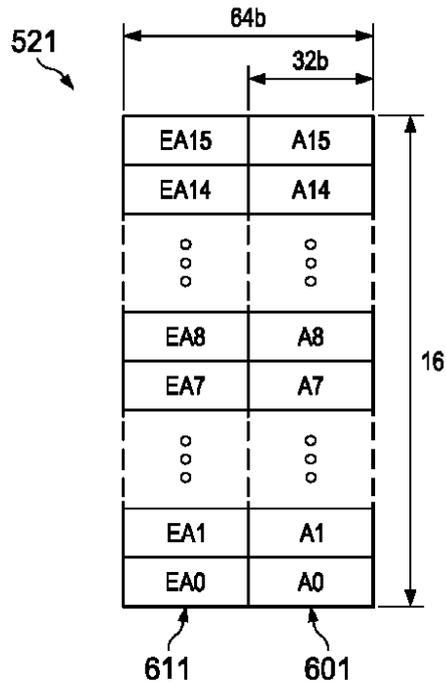


FIG. 6

【 図 7 】

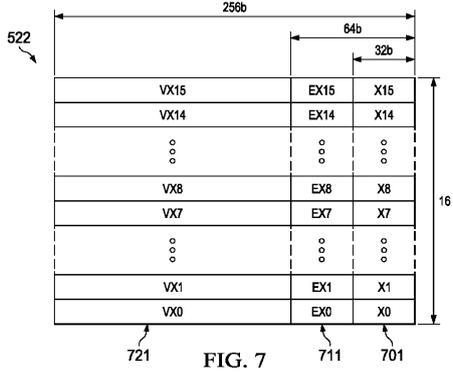


FIG. 7

【 図 9 】

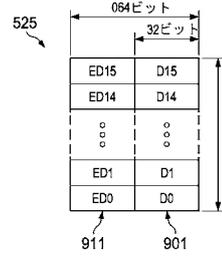


FIG. 9

【 図 8 】

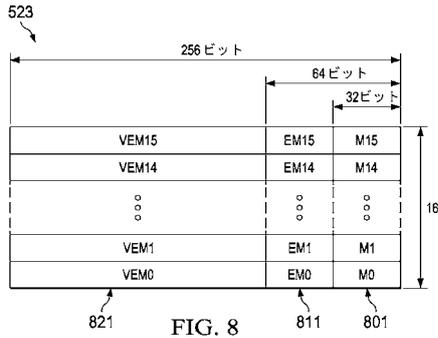


FIG. 8

【 図 1 0 】

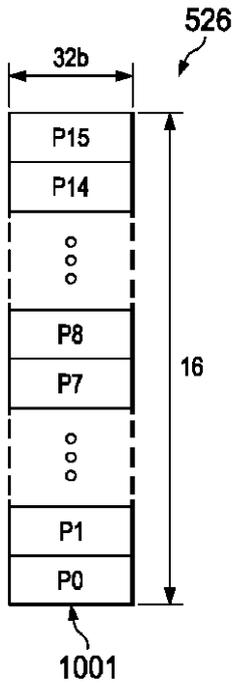


FIG. 10

【 図 1 1 】

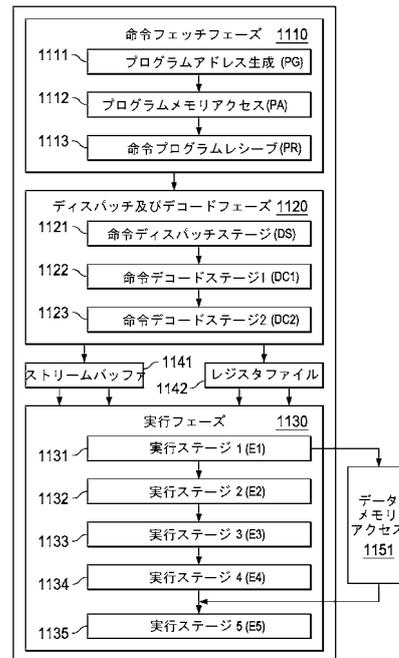
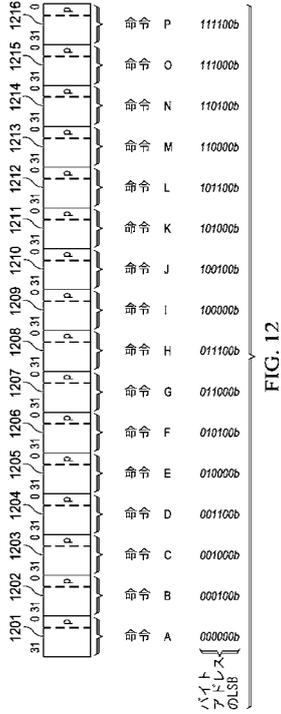
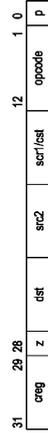


FIG. 11

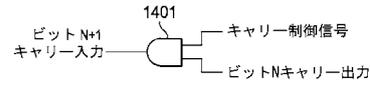
【 図 1 2 】



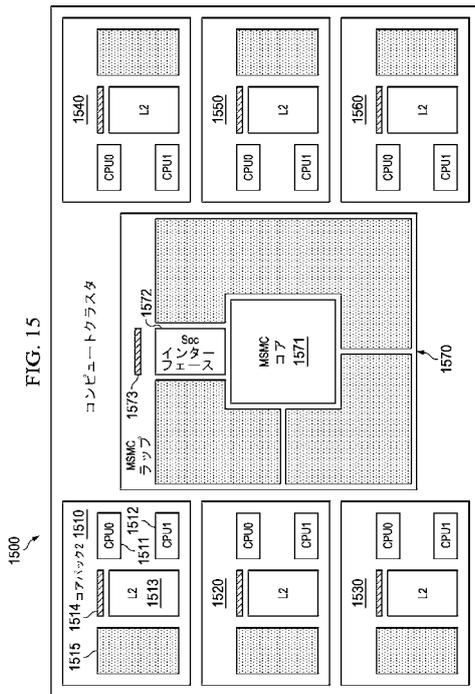
【 図 1 3 】



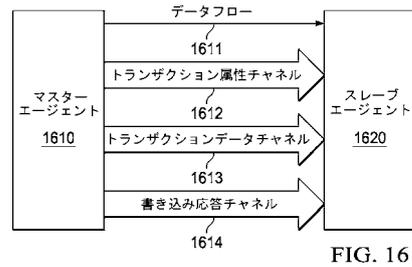
【 図 1 4 】



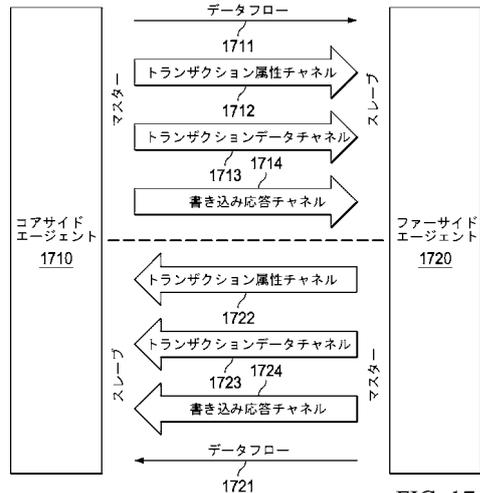
【 図 1 5 】



【 図 1 6 】



【 図 1 7 】



【図18】

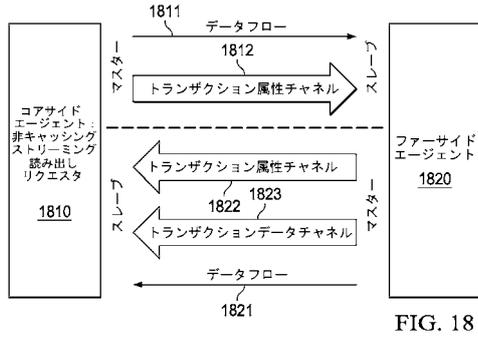


FIG. 18

【図19】

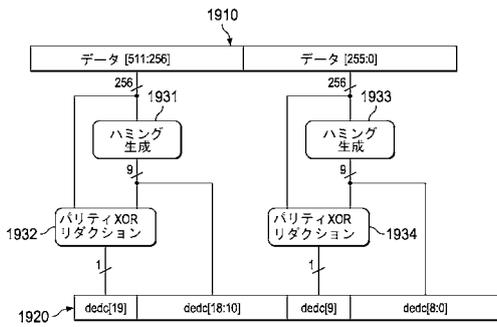


FIG. 19

【図22】

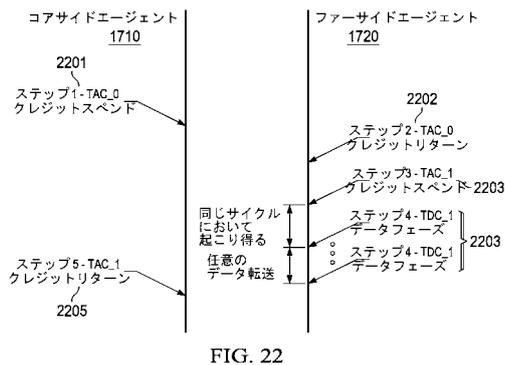


FIG. 22

【図23】

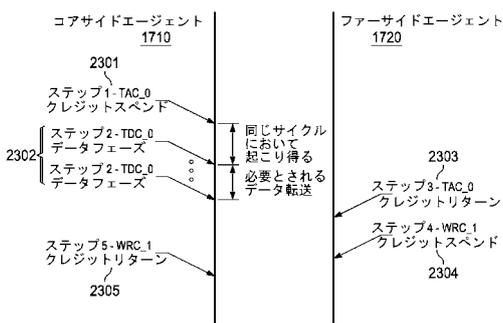


FIG. 23

【図20】

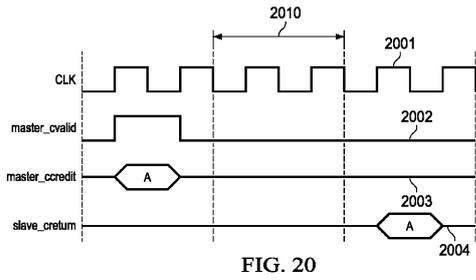


FIG. 20

【図21】

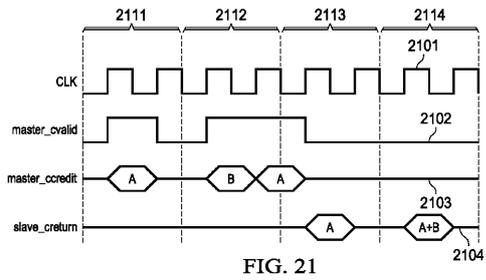


FIG. 21

【図24】

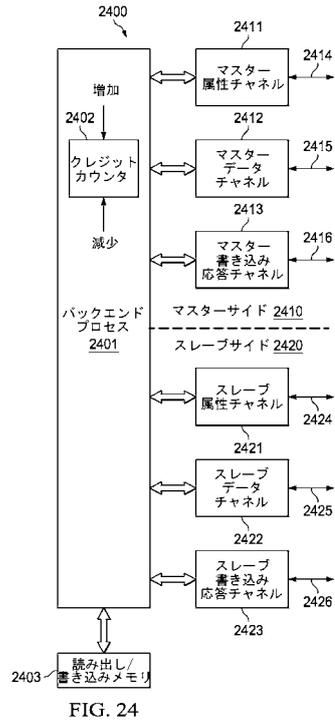


FIG. 24

## 【 国際調査報告 】

<b>INTERNATIONAL SEARCH REPORT</b>		International application No. <b>PCT/US 2015/058688</b>
<b>A. CLASSIFICATION OF SUBJECT MATTER</b>		
<i>G06F 13/38 (2006.01)</i> <i>G06F 15/16 (2006.01)</i>		
According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b>		
Minimum documentation searched (classification system followed by classification symbols)		
G06Q 13/00, 13/38, 15/00, 15/16		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
PatSearch (RUPTO internal), USPTO, PAJ, K-PION, Esp@cenet, Information Retrieval System of FIPS		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7911952 B1 (MIPS TECHNOLOGIES, INC) 22.03.2011, abstract, column 2, lines 14-25, column 3, lines 24-49, column 4, lines 6-15, lines 61-67, column 5, line 1 –column 8, line 45, column 9, line 41 –column 10, line 29, fig. 1, 2, 4, 5B, 6, 9	1-43
A	US 7103672 B1 (HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.) 05.09.2006	1-43
A	US 7165094 B2 (SONICS, INC) 16.01.2007	1-43
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents:		
"A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family	
Date of the actual completion of the international search	Date of mailing of the international search report	
14 January 2016 (14.01.2016)	21 January 2016 (21.01.2016)	
Name and mailing address of the ISA/RU: Federal Institute of Industrial Property, Berezhkovskaya nab., 30-1, Moscow, G-59, GSP-3, Russia, 125993 Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37	Authorized officer  D. Starshinov  Telephone No. (499) 240-25-91	

## フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US

(特許庁注：以下のものは登録商標)

1. W C D M A

(72)発明者 デヴィッド エム トンプソン

アメリカ合衆国 75225 テキサス州 ダラス, アマースト アヴェニュー 4020

(72)発明者 ティモシー ディー アンダーソン

アメリカ合衆国 75205 テキサス州 ユニバーシティ パーク, ミルトン アヴェニュー 3015

(72)発明者 ジョセフ アール エム ズビシアック

アメリカ合衆国 75234 テキサス州 ファーマーズ ブランチ, ブラインクレスト ドライブ 3268

(72)発明者 アブヒジート エイ チャチャド

アメリカ合衆国 75024 テキサス州 プラノ, ヘルストン ドライブ 4305

(72)発明者 カイ チルカ

アメリカ合衆国 75254 テキサス州 ダラス, スプリング ヴァレイ ロード 7038

(72)発明者 マシュー ディー ピールソン

アメリカ合衆国 75094 テキサス州 マーフィ, ラブ バード ランド 244

Fターム(参考) 5B061 BA01

5B077 AA17 BA09