

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号  
特許第5225105号  
(P5225105)

(45) 発行日 平成25年7月3日(2013.7.3)

(24) 登録日 平成25年3月22日(2013.3.22)

(51) Int.Cl.

G 0 6 F 9 / 3 8 ( 2 0 0 6 . 0 1 )

F I

G O 6 F 9 / 3 8 3 7 O C

請求項の数 37 (全 37 頁)

(21) 出願番号	特願2008-552538 (P2008-552538)	(73) 特許権者	508137947
(86) (22) 出願日	平成19年1月22日 (2007.1.22)		エクセジー・インコーポレイテッド
(65) 公表番号	特表2009-528584 (P2009-528584A)		アメリカ合衆国、ミズーリ・63127、
(43) 公表日	平成21年8月6日 (2009.8.6)		セント・ルイス、サウス・ガイヤー・ロー
(86) 国際出願番号	PCT/US2007/060835		ド・3668、スイート・300
(87) 国際公開番号	W02007/087507	(73) 特許権者	500204278
(87) 国際公開日	平成19年8月2日 (2007.8.2)		ワシントン ユニヴァーシティー
審査請求日	平成22年1月22日 (2010.1.22)		アメリカ合衆国、ミズーリ州 63130
(31) 優先権主張番号	11/339,892		、セント・ルイス、ワン・ブルッキングス
(32) 優先日	平成18年1月26日 (2006.1.26)		・ドライブ (番地なし)
(33) 優先権主張国	米国 (US)	(74) 代理人	100062007
			弁理士 川口 義雄
		(74) 代理人	100114188
			弁理士 小野 誠

最終頁に続く

(54) 【発明の名称】 F P G A ベースのパイプライン処理のためのファームウェアソケットモジュール

(57) 【特許請求の範囲】

【請求項 1】

再構成可能なロジックデバイスが再構成可能なロジックデバイスの外部のプロセッサ及びメモリとインタフェースする方法であって、

再構成可能なロジックデバイスが、少なくとも1つのファームウェアアプリケーションモジュールを有し、ファームウェアアプリケーションモジュールが、コマンドによって規定されるデータ処理動作を実行するように構成され、

プロセッサが複数の記述子をバッファに書込み、各記述子が、ダイレクトメモリアクセス (DMA) 転送のための情報の関連するブロックが格納された外部メモリ内のロケーションを特定し、記述子が複数のコマンド記述子と複数のターゲットデータ記述子を有し、各コマンド記述子が少なくとも1つのファームウェアアプリケーションモジュールに対するデータ処理動作を定義するコマンド情報ブロックに対応し、各ターゲットデータ記述子が、少なくとも1つのファームウェアアプリケーションモジュールがコマンドによって規定されるデータ処理動作をターゲットデータに実行するために、少なくとも1つのファームウェアアプリケーションモジュールへ供給するためのターゲットデータを定義するターゲットデータ情報ブロックに対応し、

再構成可能なロジックデバイスが、再構成可能なロジックデバイスへのDMA転送のために、情報が外部メモリ内で利用可能であるという、プロセッサからの指示を受信し、

受信した指示応答して、再構成可能なロジックデバイスが、(1) バッファから記述子を読み取り、(2) 外部メモリから再構成可能なロジックデバイスへ読み取られた記述子

により識別される情報ブロックのDMA転送を実行し、(3)DMA転送が継続すべきかどうかをモニタし、および(4)バッファ内の他の記述子に対して該読み取りおよび該実行するステップを繰り返し、それにより、モニタするステップがDMA転送を継続すべきと決定する限り、該書き込みステップを除き、プロセッサによる介入を必要とすることなしに、再構成可能なロジックデバイスへのストリームとして情報ブロックを転送し、ストリームは、インタリーブされたコマンド情報ブロックとターゲットデータ情報ブロックを含み、ストリームが、ターゲットデータ情報ブロックからコマンド情報ブロックへ及び、コマンド情報ブロックからターゲットデータ情報ブロックへ遷移しても、少なくとも1つのファームウェアアプリケーションモジュールが、中断なしに、コマンド情報ブロックとターゲットデータ情報ブロックを連続して処理する、方法。

10

【請求項2】

再構成可能なロジックデバイスが、ファームウェアソケットモジュールで構成され、受信するステップ、読み取るステップ、実行するステップ、モニタするステップ及び繰り返すステップがファームウェアソケットモジュールにより実行され、さらに、ファームウェアソケットモジュールが転送された情報ブロックを少なくとも1つのファームウェアアプリケーションモジュールへ供給することを含む、請求項1に記載の方法。

【請求項3】

少なくとも1つのファームウェアアプリケーションモジュールが、パイプラインに構成された複数のファームウェアアプリケーションモジュールを含む、請求項2に記載の方法。

20

【請求項4】

コマンド情報ブロックは、ファームウェアアプリケーションモジュールに対するデータ処理動作を定義する複数のコマンドを含む、請求項3に記載の方法。

【請求項5】

複数のコマンド記述子の各々が、外部メモリ内のコマンド情報ブロックに対するロケーションへのポインタを含み、複数のターゲットデータ記述子の各々が、外部メモリ内のターゲットデータ情報ブロックに対するロケーションへのポインタを含み、コマンド記述子とターゲットデータ記述子は定義された順序でバッファ内にインタリーブされて格納され、読み取りステップが定義された順序に従って記述子を読み取ることを含み、DMA転送を実行するステップは、定義された順序に従って、外部メモリから再構成可能なロジックデバイスへ、コマンド情報ブロックとターゲットデータ情報ブロックを転送することを含む、請求項4に記載の方法。

30

【請求項6】

提供するステップは、ファームウェアソケットモジュールが、転送されたコマンド情報ブロックおよびターゲットデータ情報ブロックを、パイプラインへ、定義された順序に従って提供することを含む、請求項5に記載の方法。

【請求項7】

DMA転送を実行するステップがコマンド情報ブロックとターゲットデータ情報ブロックのDMA転送を、共通の通信パスを介して実行することを含む、請求項5に記載の方法。

40

【請求項8】

各記述子が、関連付けされた情報ブロックが格納された外部メモリ内のアドレスを含む、請求項5に記載の方法。

【請求項9】

アドレスが、物理および仮想メモリロケーションアドレスを含む、請求項8に記載の方法。

【請求項10】

バッファ内の連続的に順序付けされた記述子に対応するターゲットデータ情報ブロックが、連続的なアドレスで外部メモリに格納される、請求項8に記載の方法。

【請求項11】

50

各記述子が、関連する情報ブロックのサイズを識別するバイトカウントフィールドをさらに含む、請求項 8 に記載の方法。

【請求項 1 2】

各記述子が、関連する情報ブロックがコマンドに対応するか又はターゲットデータに対応するかどうかを識別するフラグをさらに含む、請求項 1 1 に記載の方法。

【請求項 1 3】

読み取りステップ中に読み取られた記述子が、入力記述子を含み、  
書き込みステップは、プロセッサが、複数の出力記述子をバッファへ書き込むことをさらに有し、各出力記述子が、再構成可能なロジックデバイスから出力する関連する情報のブロックが格納されるべき、外部メモリ内のロケーションを特定し、

読み取りステップは、ファームウェアソケットモジュールが、出力記述子が格納されたバッファから出力記述子を読み取ることを含み、

DMA 転送を実行するステップは、ファームウェアソケットモジュールが、再構成可能なロジックデバイスから読み取られた出力記述子により識別された外部メモリ内のロケーションへ、出力する情報ブロックのアウトバウンド DMA 転送を実行することを含み、

繰り返すステップは、ファームウェアソケットモジュールが、出力記述子が格納されたバッファ内の他の出力記述子に対して、出力記述子を読み取るステップと、アウトバウンド DMA 転送を実行することを繰り返し、それにより、モニタするステップが DMA 転送を継続すべきと決定する限り、入力及び出力記述子書き込みステップを除き、プロセッサによる介入なしに、再構成可能なロジックデバイスからのストリームとして出力する情報ブロックを転送することを含む、請求項 5 に記載の方法。

【請求項 1 4】

出力する情報ブロックは、処理されたコマンド情報ブロック及び処理されたターゲットデータ情報ブロックを含む、請求項 1 3 に記載の方法。

【請求項 1 5】

出力記述子読み取りステップが、(1) 第 1 の出力記述子バッファから、処理されたコマンド情報ブロックについての出力記述子を読み取り、(2) 第 2 の出力記述子バッファから、処理されたターゲットデータ情報ブロックについての出力記述子を読み取ることを含む、請求項 1 4 に記載の方法。

【請求項 1 6】

ファームウェアソケットモジュールが、複数のレジスタを格納し、複数のレジスタが、(1) 読み取られるべき現在の入力記述子と、(2) 入力記述子が格納されたバッファのサイズを示すカウントと、(3) 読み取られるべき次の入力記述子と、(4) 読み取られるべき処理されたコマンド情報ブロックについての現在の出力記述子と、(5) 第 1 の出力記述子バッファのサイズを示すカウントと、(6) 読み取られるべき処理されたコマンド情報ブロックについての次の出力記述子と、(7) 読み取られるべき処理されたターゲットデータ情報ブロックについての現在の出力記述子と、(8) 第 2 の出力記述子バッファのサイズを示すカウントと、(9) 読み取られるべき処理されたターゲットデータ情報ブロックについての次の出力記述子を特定し、ファームウェアソケットモジュールが、レジスタ内に格納されたデータに基づいて、読み取りステップ、DMA 転送ステップ、モニタステップ及び繰り返しステップを実行する、請求項 1 5 に記載の方法。

【請求項 1 7】

ファームウェアソケットモジュールが、定義された優先順位に従って、出力する処理されたコマンド情報ブロックに最高の優先順位を与え、出力する処理されたターゲットデータ情報ブロックに次に高い優先順位を与え、入力するコマンド情報ブロックとターゲットデータ情報ブロックに最低の優先順位を与えることにより、再構成可能なロジックデバイスへのまたは再構成可能なロジックデバイスからのデータフローを管理することをさらに含む、請求項 1 4 に記載の方法。

【請求項 1 8】

ファームウェアソケットモジュールが、出力する情報ブロックに対する DMA 転送を実

10

20

30

40

50

行後に、プロセッサに、出力する情報ブロックが利用可能であることを通知することをさらに有する、請求項 13 に記載の方法。

【請求項 19】

ファームウェアソケットモジュールが、複数のレジスタを格納し、複数のレジスタが、(1) 読み取られるべき現在の入力記述子と、(2) バッファのサイズを示すカウンタと、(3) 読み取られるべき次の入力記述子を識別することをさらに有し、ファームウェアソケットモジュールが、レジスタ内に格納されたデータに基づいて、読み取りステップ、DMA 転送ステップ、モニタステップ及び繰り返しステップを実行する、請求項 5 に記載の方法。

【請求項 20】

複数のコマンド情報ブロックが、ファームウェアアプリケーションモジュールの個々のモジュールに固有である、請求項 5 に記載の方法。

【請求項 21】

コマンド情報ブロックの少なくとも 1 つが、ファームウェアアプリケーションモジュールの少なくとも 1 つへのパススルーコマンドを含む、請求項 5 に記載の方法。

【請求項 22】

コマンド情報ブロックによって定義されるデータ処理動作が、検索動作、データ縮約動作、暗号化動作、解読動作、圧縮動作、伸張動作、およびパススルー動作から成るグループの少なくとも 2 つの動作を含む、請求項 5 に記載の方法。

【請求項 23】

コマンド情報ブロックの少なくとも 1 つが、検索動作、データ縮約動作、暗号化動作、解読動作、圧縮動作、伸張動作、およびパススルー動作から成るグループから選択された少なくとも 1 つの動作を実行するようにファームウェアアプリケーションモジュール構成するコマンドを含む、請求項 4 に記載の方法。

【請求項 24】

バッファが、プロセッサにより管理されるメモリ空間内に配置されたバッファを含む、請求項 1 に記載の方法。

【請求項 25】

バッファが、RAM (ランダムアクセスメモリ) を含む、請求項 24 に記載の方法。

【請求項 26】

外部メモリが、プロセッサのメモリ空間内のメモリを含む請求項 1 に記載の方法。

【請求項 27】

メモリ空間が、プロセッサのカーネルメモリ空間を含む、請求項 26 に記載の方法。

【請求項 28】

外部メモリが、ランダムアクセスメモリ (RAM) を含む請求項 26 に記載の方法。

【請求項 29】

外部メモリが、物理的にアドレス可能なデータストレージ装置を含む請求項 1 に記載の方法。

【請求項 30】

再構成可能なロジックデバイスが、フィールドプログラマブルゲートアレイ (FPGA) を含む、請求項 1 に記載の方法。

【請求項 31】

再構成可能なロジックデバイスを再構成可能なロジックデバイスの外部のプロセッサ及びメモリとインタフェースする装置であって、

再構成可能なロジックデバイスが、少なくとも 1 つのファームウェアアプリケーションモジュールを有し、ファームウェアアプリケーションモジュールが、コマンドによって規定されるデータ処理動作を実行するように構成され、装置がさらに、

プロセッサを有し、プロセッサが複数の記述子をバッファに書込むように構成され、各記述子が、ダイレクトメモリアクセス (DMA) 転送のための情報の関連するブロックが格納された外部メモリ内のロケーションを特定し、記述子が数のコマンド記述子と複数の

10

20

30

40

50

ターゲットデータ記述子を有し、各コマンド記述子が少なくとも1つのファームウェアアプリケーションモジュールに対するデータ処理動作を定義するコマンド情報ブロックに対応し、各ターゲットデータ記述子が、少なくとも1つのファームウェアアプリケーションモジュールがコマンドによって規定されるデータ処理動作をターゲットデータに実行するために、少なくとも1つのファームウェアアプリケーションモジュールへ供給するためのターゲットデータを定義するターゲットデータ情報ブロックに対応し、

再構成可能なロジックデバイスを有し、再構成可能なロジックデバイスが外部メモリ及びプロセッサと通信し、再構成可能なロジックデバイスが、(1)DMA転送のために、情報が外部メモリ内で利用可能であるという、プロセッサからの指示を受信し、(2)受信した指示に応答して、(i)バッファ内の記述子の読み取り動作を実行し、(ii)外部メモリから再構成可能なロジックデバイスへの読み取られた記述子により識別される情報ブロックのDMA転送を実行し、(iii)DMA転送が継続すべきかどうかをモニタし、および(iv)バッファ内の他の記述子に対して該読み取りおよび該DMA転送を繰り返す、それにより、モニタ動作がDMA転送を継続すべきと決定する限り、該書き込み動作を除き、プロセッサによる介入を必要とすることなしに、再構成可能なロジックデバイスへのストリームとして情報ブロックを転送するように構成され、ストリームは、インタリーブされたコマンド情報ブロックとターゲットデータ情報ブロックを含み、ストリームが、ターゲットデータ情報ブロックからコマンド情報ブロックへ及び、コマンド情報ブロックからターゲットデータ情報ブロックへ遷移しても、少なくとも1つのファームウェアアプリケーションモジュールが、中断なしに、コマンド情報ブロックとターゲットデータ情報ブロックを連続して処理する、装置。

【請求項32】

データ処理システムであって、

外部メモリに格納されたコマンドデータとターゲットデータにアクセスするためのバスに接続するためのデータ処理カードを有し、データ処理カードがパイプラインに配置された複数のデータ処理モジュールを有するように構成され、各データ処理モジュールが、受信する任意のターゲットデータにコマンドによって規定されるデータ処理動作を実行するように構成され、

データ処理カードが、(1)データ処理カードへのダイレクトメモリアクセス(DMA)転送のために、情報が外部メモリ内で利用可能であるというバスと通信するプロセッサからの指示を受信し、(2)受信した指示に応答して、(i)バッファから複数の記述子を読み取り、バッファはプロセッサにより書き込まれた複数の記述子を有し、記述子が複数のコマンド記述子と複数のターゲットデータ記述子を含み、各コマンド記述子が、DMA転送のためのコマンドデータが格納された外部メモリ内のロケーションを特定し、コマンドデータがパイプライン内の少なくとも1つのデータ処理モジュールにより実行されるデータ処理動作を制御し、各ターゲットデータ記述子が、DMA転送のためのターゲットデータが格納された外部メモリ内のロケーションを特定し、コマンド記述子とターゲットデータ記述子がパイプラインを通して処理されるべきコマンドデータとターゲットデータに対する定義された順序に従ってバッファ内にインタリーブされて格納され、(ii)外部メモリからデータ処理カードへの読み取られた記述子により識別されるコマンドデータとターゲットデータのDMA転送を実行し、それにより、データ処理カードがDMA転送を継続すべきと決定する限り、プロセッサが記述子をバッファに書き込むことを除き、プロセッサによる介入を必要とすることなしに、データ処理カードへのストリームとしてコマンドデータとターゲットデータを転送し、ストリームは、インタリーブされたコマンドデータとターゲットデータを含み、ストリームがターゲットデータからコマンドデータへ及び、コマンドデータからターゲットデータへ遷移しても、パイプラインが、中断なしに、コマンドデータとターゲットデータを連続して処理する、ように構成される、データ処理システム。

【請求項33】

プロセッサと、外部メモリと、バスをさらに有する請求項32に記載のシステム。

## 【請求項 3 4】

コマンド記述子の各々が、外部メモリ内のコマンドデータについてのロケーションに対するポインタを含み、ターゲットデータ記述子の各々が、外部メモリ内のターゲットデータについてのロケーションに対するポインタを含み、プロセッサがデバイスドライバを含み、デバイスドライバが定義された順序でバッファ内にコマンド記述子とターゲットデータ記述子をインタリーブして格納するように構成された、請求項 3 3 に記載のシステム。

## 【請求項 3 5】

読み取られた記述子が入力記述子を含み、デバイスドライバが、もう一つのバッファも複数の出力記述子でポピュレートすることにより、ポピュレート動作を実行するようにさらに構成され、各出力記述子が、パイプラインを出力する関連するデータのブロックが格納されるべき、外部メモリ内のロケーションを特定し、

データ処理カードが、( 1 ) もう一つのバッファからも複数の出力記述子を読み取ることにより、読み取り動作を実行し、各出力記述子が、パイプラインから出力するデータのブロックがデータ処理カードから転送されるべき、外部メモリのロケーションを特定し、( 2 ) データ処理カードから読み取った出力記述子から識別される外部メモリ内のロケーションへ出力するデータブロックのアウトバウンド DMA 転送も実行することにより DMA 転送を実行し、それにより、データ処理カードが DMA 転送を継続すべきと決定する限り、入力及び出力記述子ポピュレート動作を除き、プロセッサによる介入なしに、データ処理カードからのストリームとして出力するデータブロックを転送するようにさらに構成された、請求項 3 4 に記載のシステム。

## 【請求項 3 6】

データ処理カードが、再構成可能なロジックデバイスを含み、パイプラインが、再構成可能なロジックデバイス上で実行される、請求項 3 2 に記載のシステム。

## 【請求項 3 7】

再構成可能なロジックデバイスが、読み取り動作と DMA 転送動作を実行するようにさらに構成された、請求項 3 6 に記載のシステム。

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明は、ハードウェアにおける高速データ処理動作をサポートするソフトウェアを有するインタフェースハードウェアの分野に向けられる。

## 【背景技術】

## 【0002】

大量のデータを処理する際に、コンピューティングシステムのデータスループット能力を向上させる能力は、常に存在する課題である。多くの場合、汎用プロセッサによって実行されるソフトウェアを使用して大量のデータ量を処理することは、ユーザのニーズを満たすのに遅すぎる。したがって、データ処理機能を、コンピュータの汎用プロセッサによって実行されるソフトウェアを離れて、ハードウェア上に配備されたファームウェアの中に再配置すること、またはそのようなファームウェアとソフトウェアとの間でデータ処理機能を分割することが望ましい。しかし、そのようにする際、ハードウェアデバイス(再構成可能なロジックデバイスなどの)が、ハードウェアとソフトウェアの間でデータおよびコマンドフローが行き来する際に、ネックを生じないような仕方で、データストレージやコンピュータの汎用プロセッサによって実行されるソフトウェアなどの外部リソースと互いに接続される必要がある。

## 【0003】

本明細書で使用される、「汎用プロセッサ」という用語は、命令をフェッチして、それらの命令を実行するハードウェアデバイス(例えば、Intel Xeon(登録商標)プロセッサまたはAMD Opteron(登録商標)プロセッサ)を指す。「再構成可能なロジック」という用語は、形態および機能が、製造後にフィールドにおいて大幅に変更される(すなわち、再構成される)ことが可能な任意のロジック技術を指す。このこと

は、機能が、製造後に変更されることが可能であるが、形態は、製造時に固定される汎用プロセッサと対比されるべきである。「ソフトウェア」という用語は、汎用プロセッサ上に配備されたデータ処理機能を指す。「ファームウェア」という用語は、再構成可能なロジック上に配備されたデータ処理機能を指す。

【発明の開示】

【課題を解決するための手段】

【0004】

当技術分野におけるニーズに対処しようとして、本発明人らは、本明細書において、外部メモリおよびソフトウェアを有する再構成可能なロジックデバイス上に配備された少なくとも1つのファームウェアアプリケーションモジュールとインタフェースをとるファームウェアソケットモジュールのための新たな設計を開示する。また、このファームウェアソケットモジュールは、好ましくは、再構成可能なロジックデバイス上にも配備される。この少なくとも1つのファームウェアアプリケーションモジュールは、このモジュールが受け取る任意のターゲットデータに対してデータ処理動作を実行するように構成され、この少なくとも1つのファームウェアアプリケーションモジュールが実行するデータ処理動作は、ソフトウェアベースのコマンドによって制御される。好ましくは、この少なくとも1つのファームウェアアプリケーションモジュールは、パイプラインに構成された複数のファームウェアアプリケーションモジュールを含む。パイプラインにおける各ファームウェアアプリケーションモジュールは、好ましくは、コマンド情報に関して個々にアドレス指定可能である。このため、コマンドは、パイプラインにおける特定のファームウェアアプリケーションモジュールに発行されて、それらの特定のファームウェアアプリケーションモジュールのデータ処理動作が制御されることが可能である。

【0005】

ファームウェアソケットモジュールは、外部メモリおよびソフトウェアにアクセスして、ファームウェアアプリケーションモジュールパイプラインを通して処理されるべきコマンドおよびターゲットデータを受け取るように構成される。好ましくは、ファームウェアソケットモジュールは、コンピュータシステムのメモリからファームウェアソケットモジュールへのコマンドおよびターゲットデータのDMA（ダイレクトメモリアクセス）転送を実行するようにシステムバスヘトランザクションを発行する。ファームウェアソケットモジュールは、これらのコマンドおよびターゲットデータを、ソフトウェアによって規定される所定の順序でパイプラインにおけるファームウェアアプリケーションモジュールの第1のモジュールに供給するように構成される。

【0006】

ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールパイプラインは、好ましくは、システムを通してコマンドおよびターゲットデータを伝搬する際に、インバウンドのコマンドおよびターゲットデータの、この所定の順序を厳密に保つように構成される。ターゲットデータが、パイプラインにおける或るファームウェアアプリケーションモジュールに到達すると、そのファームウェアアプリケーションモジュールは、このターゲットデータに対して、指定されたデータ処理動作を実行し、その後、そのように処理されたターゲットデータを、パイプラインにおける次のファームウェアアプリケーションモジュールに供給する。コマンドデータが、パイプラインにおける或るファームウェアアプリケーションモジュールに到達すると、そのファームウェアアプリケーションモジュールは、このコマンドが、そのモジュールに向けられているかどうかを確認、向けられている場合、そのコマンドを解釈して、適宜、そのモジュールのデータ処理動作を再構成する。このコマンドが、パイプラインをさらに下って伝搬されるべきである場合、そのファームウェアアプリケーションモジュールは、このコマンドを、パイプラインにおける次のファームウェアアプリケーションモジュールに送る。

【0007】

ファームウェアソケットモジュールに入る、またはファームウェアアプリケーションモジュールパイプラインの入口点に入るコマンドおよびターゲットデータの流れは、コマン

ドとターゲットデータがともに、規定された順序に従ってインタリーブされた単一のストリームと考えられることが可能である。コマンドとデータがインタリーブされるという言い方がされる場合、このことは、コマンド/データ/コマンド/データ/コマンド/データ...というストリームを要求しない(排除しないものの)。代わりに、本明細書で説明されるコマンドとデータのインタリーブされたストリームは、コマンド/コマンド/コマンド/データ/データ/データ/データ/データ/データ/コマンド...などのストリームを包含し、ストリームにおけるコマンドとデータの順序は、ソフトウェアによって規定され、ファームウェアソケットモジュールが、このストリームをファームウェアアプリケーションモジュールパイプラインに伝搬する際に、ファームウェアソケットモジュールによって保存される。

10

**【0008】**

ファームウェアアプリケーションモジュールのデータ処理動作を制御する適切なコマンドが、ファームウェアアプリケーションパイプラインに入るコマンドとターゲットデータのストリームの中の、そのターゲットデータに先行して、ファームウェアアプリケーションモジュールのデータ処理動作が、ターゲットデータを処理することに先立って適切に制御されることを可能にしなければならない。コマンドとターゲットデータの、この厳密な順序付けが容易に保たれることを促すのに、ファームウェアソケットモジュールは、コマンドとターゲットデータとともに、パイプラインにおける第1のファームウェアアプリケーションモジュールに、パイプラインの第1のファームウェアアプリケーションモジュールにファームウェアソケットモジュールをリンクするのと同じ通信パスを介して、供給

20

**【0009】**

このファームウェアソケットモジュール - ファームウェアアプリケーションモジュールパイプライン間の接続によってもたらされるコマンドとターゲットデータの間の自然な同期により、システムの複雑さが回避されるとともに、データスループットが増大する。本発明人らに知られている従来技術のソケットインタフェースにおいて、コマンドは、ターゲットデータをデータ処理モジュールに通信するのに使用される通信パスとは異なる通信パスを介して、データ処理モジュールに通信される。そのようなデータ処理モジュールが、パイプライン化されると、そのようなデュアル通信パスにより、コマンドをデータと同期しようと試みる際、管理の困難が生じる。そのような場合において、パイプラインにおけるデータ処理モジュールに新たなコマンドが発行されると、そのコマンド、およびさらなるターゲットデータが、そのパイプラインを通して処理され、それによってパイプラインのスループット能力を大きく損なう前に、パイプライン全体から、通常、以前のコマンド、および以前のデータがフラッシュされる必要がある。しかし、対照的に、本発明では、パイプラインの1つのファームウェアアプリケーションモジュールが、パイプラインにおける他のファームウェアアプリケーションモジュールが、それらのモジュールの規定されたデータ処理動作に従ってデータを同時に処理している間に、コマンドに対処することができる。このため、パイプライン全体がフラッシュされることを要求することなしに、ファームウェアアプリケーションモジュールのデータ処理動作を調整するコマンドが、ファームウェアアプリケーションモジュールに発行されることが可能である。

30

40

**【0010】**

コンピュータシステムの汎用プロセッサによって実行される高レベルソフトウェアが、好ましくは、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールパイプラインを通して最終的に伝搬されるコマンドとデータの順序を規定する。次に、やはりコンピュータシステムの汎用プロセッサによって実行される、より低いレベルのデバイスドライバソフトウェアが、好ましくは、コマンドとデータの、この規定された順序を保ち、そのような順序付けられたコマンドとデータをファームウェアソケットモジュールに提供する。このデバイスドライバソフトウェアは、好ましくは、コマンドおよびターゲットデータに対するポインタが格納される入力記述子プールバッファを管理することによって、この順序を保つ。ファームウェアソケットモジュールは、入力記述子プ

50



ールバッファにアクセスして、ファームウェアアプリケーションモジュールパイプラインに送られるべきコマンドおよびターゲットデータについて知る。

【 0 0 1 1 】

ファームウェアソケットモジュールのアウトバウンド側（ソフトウェアに対してアウトバウンド）で、デバイスドライバソフトウェアは、好ましくは、出力コマンドおよびアウトバウンドデータに関して別々のバッファを保持して、ファームウェアアプリケーションモジュールによって処理されたコマンドおよびデータが、コンピュータシステムソフトウェアによる後のアクセスのために、メモリの中でどこに格納されるべきかについて、ファームウェアソケットモジュールに通知する。

【 0 0 1 2 】

本発明の好ましい実施形態が提供する利点の中には、ソフトウェアから再構成可能なロジックデバイスに、さらに再構成可能なロジックデバイスからソフトウェアに、フロー制御されたデータを実際に送る能力、およびファームウェアアプリケーションモジュールが配備されるコンピュータシステムとは独立した（ファームウェアアプリケーションモジュールが、ファームウェアソケットモジュールの通知要件を遵守する限り）ファームウェアアプリケーションモジュールを開発する能力がある。本発明のこれら、およびその他の特徴は、以下の説明および図を検討すると、一部は指摘され、一部は、当業者には明白となる。

【発明を実施するための最良の形態】

【 0 0 1 3 】

図 1 は、本発明による好ましいシステム 1 0 0 を示す。このシステムにおいて、再構成可能なロジックデバイス 1 0 2 が、ディスクコントローラ 1 0 6 およびデータストア 1 0 4 によって規定されるディスクサブシステムを離れて流れる（直接に、または R A M 1 1 0 などのシステムメモリを介して間接的に）データを受け取るように位置付けられる。好ましくは、このデータは、システムバス 1 1 2 を介して再構成可能なロジックデバイスに流れ込むが、他の設計アーキテクチャも可能である（図 3 b 参照）。好ましくは、再構成可能なロジックデバイスは、F P G A（フィールドプログラマブルゲートアレイ）であるが、そうでなくてもよい。また、システムバス 1 1 2 は、再構成可能なロジックデバイス 1 0 2 を、コンピュータシステムのメインプロセッサ 1 0 8、ならびにコンピュータシステムの R A M 1 1 0 と互いに接続する。本明細書で使用される「バス」という用語は、デバイスおよびロケーションがアドレスによってアクセスされる任意の物理的相互接続を包含する論理バスを指す。本発明の実施の際に使用されることが可能なバスの例には、P C I ファミリのバス（例えば、P C I - X および P C I - E x p r e s s）、および H y p e r T r a n s p o r t バスが含まれるが、以上には限定されない。好ましい実施形態において、システムバス 1 1 2 は、P C I - X バスであることが可能であるが、そうでなくてもよい。

【 0 0 1 4 】

データストアは、任意のデータ記憶デバイス/システムであることが可能であるが、好ましくは、何らかの形態の大容量記憶媒体である。例えば、データストア 1 0 4 は、S e a g a t e（登録商標）ディスクのアレイなどの磁気記憶デバイスであることが可能である。しかし、他のタイプの記憶媒体も、本発明の実施の際に使用するのに適していることに留意されたい。例えば、データストアは、インターネットまたは何らかの L A N（ローカルエリアネットワーク）などのネットワークを介してアクセスされる 1 つまたは複数の遠隔データ記憶デバイスであることも可能である。

【 0 0 1 5 】

メインプロセッサ 1 0 8 および R A M 1 1 0 によって規定されるコンピュータシステムは、好ましくは、当業者によって理解される通り、任意の商品コンピュータシステムである。例えば、コンピュータシステムは、I n t e l X e o n（登録商標）システムまたは A M D O p t e r o n（登録商標）システムであることが可能である。

【 0 0 1 6 】

再構成可能なロジックデバイス102は、デバイス102の機能を規定するデバイス102上に配備されたファームウェアモジュールを有する。ファームウェアソケットモジュール120が、再構成可能なロジックデバイスを出入りするデータ移動要件(コマンドデータとターゲットデータの両方)を扱い、やはり再構成可能なロジックデバイス上に配備されるFAM(ファームウェアアプリケーションモジュール)チェーン130に対する一貫したインタフェースを提供する。FAMチェーン130のFAM130iが、ファームウェアソケットモジュール120からチェーン130を通して流れるターゲットデータに対して、指定されたデータ処理動作を実行するように構成される。再構成可能なロジック上に配備されることが可能なFAMの好ましい実施例が、米国特許第6,711,558号(「Associative Database Scanning and Information Retrieval」という名称の)、係属中の米国特許出願第10/153,151号(「Associative Database Scanning and Information Retrieval using FPGA Devices」という題名で2002年5月21日に出願され、2003/0018630として公開された)、公開PCT出願、WO 05/048134およびWO 05/026925(ともに、2004年5月21日に出版され、「Intelligent Data Storage and Processing Using FPGA Devices」という題名を有する)、米国特許仮出願第60/658,418号(2005年3月3日に出版され、「Biosequence Similarity Searching Using FPGA Devices」という題名を有する)、米国特許仮出願第60/736,081号(2005年11月11日に出版され、「Method and Apparatus for Performing Biosequence Similarity Searching」という題名を有する)および米国特許出願第11/293,619号(2005年12月2日に出版され、「Method and Device for High Performance Regular Expression Pattern Matching」という題名を有する)において説明されており、これらの特許文献の各文献の開示全体が、参照により本明細書に組み込まれている。例えば、FAMによって実行されることが可能である例示的なデータ処理動作の網羅的ではないリストには、データ検索動作(様々なタイプの)、データ暗号化動作(いくつかの暗号化技術および/または暗号化キーのいずれかを使用する)、データ解読動作(いくつかの解読技術および/または解読キーのいずれかを使用する)、データ圧縮動作(いくつかのデータ圧縮技術のいずれかを使用する)、データ伸張動作(いくつかのデータ伸張技術のいずれかを使用する)、およびデータ縮約動作が含まれる。

#### 【0017】

FAMによって実行される特定のデータ処理動作は、FAMが、ファームウェアソケットモジュール120から受け取るコマンドデータによって制御/パラメータ化される。このコマンドデータは、FAM固有であることが可能であり、このコマンドが受け取られると、FAMは、受け取られたコマンドによって制御されるデータ処理動作を実行するようにFAMを構成する。

#### 【0018】

FAMが、受け取られたコマンドによって指定されるデータ処理動作を実行するように構成されると、そのFAMは、ファームウェアソケットモジュールからそのFAMが受け取るターゲットデータに対して、そのFAMの指定されたデータ処理動作を実行する準備ができています。このため、FAMは、ターゲットデータの指定されたストリームの中で、「Smith」という語の存在を検索するように、適切なコマンドを介して構成されることが可能である。FAMが、「Smith」という語に関して、ターゲットデータストリームに対して検索動作を実行すると、FAMに、「Jones」という語を検索するようにFAMを再構成させる別のコマンドが、そのFAMに送られることが可能である。FAMは、ハードウェア速度で動作する(その結果、FAMを通るターゲットデータの高速スループットをもたらす)だけでなく、FAMは、FAMのデータ処理動作のパラメータを

変更するように柔軟に再プログラミングされることも可能である。

【 0 0 1 9 】

F A Mチェーン 1 3 0 は、好ましくは、パイプライン化されたシーケンスに構成される、複数の F A M (ファームウェアアプリケーションモジュール) 1 3 0 a、1 3 0 b . . . を含む。本明細書で使用する「パイプライン」、「パイプライン化されたシーケンス」、または「チェーン」とは、F A Mの構成を指し、1つの F A Mの出力が、シーケンスにおける次の F A Mの入力に接続される。このパイプライン構成は、各 F A Mが、所与のクロックサイクル中にその F A Mが受け取るデータに対して独立に動作し、次に、その F A Mの出力を、別のクロックサイクル中に、シーケンスにおける次の下流の F A Mに送ることを可能にする。

10

【 0 0 2 0 】

通信バス 1 3 2 が、ファームウェアソケットモジュール 1 2 0 を、パイプライン化された F A Mの第 1 の F A M 1 3 0 aの入力に接続する。第 1 の F A M 1 3 0 aの入力は、F A Mチェーン 1 3 0 への入口点の役割をする。通信バス 1 3 4 が、パイプライン化された F A Mの最後の F A M 1 3 0 mの出力をファームウェアソケットモジュール 1 2 0 に接続する。最後の F A M 1 3 0 mの出力は、F A Mチェーン 1 3 0 の出口点の役割をする。通信バス 1 3 2 と通信バス 1 3 4 はともに、好ましくは、マルチビットバスである。

【 0 0 2 1 】

図 2 は、図 1 のシステム 1 0 0 上のアプリケーションの配備に関する例示的なフレームワークを示す。図 2 の上位の 3 つの層は、コンピュータシステムの汎用プロセッサ 1 0 8 上のソフトウェアにおいて実行される機能を表す。下位の 2 つの層は、再構成可能なロジックデバイス 1 0 2 上のファームウェアにおいて実行される機能を表す。

20

【 0 0 2 2 】

アプリケーションソフトウェア層 2 0 0 は、1 名または複数名のユーザが、アプリケーションと対話して、いずれのデータ処理動作が、F A Mによって実行されるべきであるかを規定し、さらに、それらのデータ処理動作が、どのようなターゲットデータに対して実行されるべきであるかを規定するタイプの機能などの、高レベル機能に対応する。

【 0 0 2 3 】

次の層は、高レベルモジュール A P I 2 0 2 a と、低レベルモジュール A P I 2 0 2 b とを含むモジュール A P I (アプリケーションプログラミングインタフェース) 層 2 0 2 である。高レベルモジュール A P I 2 0 2 a は、アプリケーションレベルソフトウェアに一般的なサービス (例えば、呼び戻しを管理すること) を提供することができる。低レベルモジュール A P I 2 0 2 b は、O S (オペレーティングシステム) レベル / デバイスドライバソフトウェア 2 0 4 の動作を管理する。ソフトウェアライブラリインタフェース 2 1 0 が、高レベルモジュール A P I 2 0 2 a を低レベルモジュール A P I 2 0 2 b に結び付ける。このソフトウェアライブラリインタフェースについての詳細は、後段で提供される。

30

【 0 0 2 4 】

デバイスドライバソフトウェア 2 0 4 とファームウェアソケットモジュール 1 2 0 との間のインタフェースは、システム 1 0 0 に関するハードウェア / ソフトウェアインタフェース 2 1 2 の役割をする。このインタフェース 2 1 2 の詳細は、図 5 に関連して、より詳細に説明される。

40

【 0 0 2 5 】

ファームウェアソケットモジュール 1 2 0 と、F A Mチェーン 1 3 0 との間のインタフェースは、ファームウェアモジュールインタフェース 2 1 4 である。このインタフェースの詳細は、図 4 に関連して、より詳細に説明される。

【 0 0 2 6 】

図 3 a は、商品コンピュータシステムの P C I - X バス 1 1 2 に接続されることが可能なプリント回路基板またはプリント回路カード 3 0 0 を示す。図 3 a の例において、プリント回路基板は、メモリデバイス 3 0 4 および P C I - X バスコネクタ 3 0 6 と通信する

50

FPGA302 (Xilinx Virtex (登録商標) II FPGAなどの)を含む。好ましいメモリデバイス304は、SRAMメモリとDRAMメモリとを含む。好ましいPCI-Xバスコネクタ306は、標準のカードエッジコネクタである。

【0027】

図3bは、プリント回路基板/カード300に関する代替の構成を示す。図3bの例において、プライベートバス308 (PCI-Xバスなどの)、ディスクコントローラ310、およびディスクコネクタ312も、プリント回路基板300上に実装される。SCSI、SATA、FC (Fibre Channel) などを含むが、以上には限定されない、任意の商品ディスク技術が、サポートされることが可能である。この構成において、ファームウェアソケット120は、PCI-X-PCI-X間ブリッジの役割もして、プライベートPCI-Xバス308を介して接続されたディスクへの通常のアクセスをプロセッサ108に与える。

10

【0028】

図3aまたは図3bの構成において、ファームウェアソケット120は、PCI-Xバスにメモリ304をアクセス可能にすることができ、その結果、メモリ304が、ディスクコントローラからFAMへの転送のためのバッファとしてOSカーネル204によって使用されることが可能になることは、注目に値する。また、単一のFPGA302が、図3aおよび図3bのプリント回路基板上に示されているが、複数のFPGAが、プリント回路基板300上の複数のFPGAを含めること、またはコンピュータシステムに複数のプリント回路基板300を実装することによって、サポートされることが可能であるものと理解されるべきことも注目に値する。図8は、単一のパイプラインにおける多数のFAMが、複数のFPGAにまたがって配備される例を示す。

20

【0029】

図1 - 図3に示される通り、インバウンドデータ (カーネル204からカード300への) が、コンピュータシステム内のバス112をわたってファームウェアソケットモジュール120まで移動され、次に、ファームウェアソケットモジュール120によってFAMチェーン130に送られる。アウトバウンドデータ (カード300からカーネル204への) が、FAMチェーン130からファームウェアソケットモジュール120に送られ、次に、ファームウェアソケットモジュール120によって、PCI-Xバスをわたって、コンピュータシステム上で実行されるソフトウェアアプリケーションに送られる。図2に示される通り、使用される3つの対話するインタフェースは、ファームウェアモジュールインタフェース214、ハードウェア/ソフトウェアインタフェース212、およびソフトウェアライブラリインタフェース210である。

30

【0030】

ファームウェアモジュールインタフェース214 - 信号層

図4に示されるファームウェアモジュールインタフェース214は、ファームウェアソケットモジュール120間で交換される信号、およびFAMチェーン130内のFAM間で交換される信号を含む。表1、および後段の説明が、これらの信号をさらに説明する。

【表 1】

表1:

信号名	説明	信号方向	信号サイズ
clk	クロック	入力	信号
reset	リセット	入力	信号
module_id	モジュールID	入力	6ビットバス
data_in	入力データ	入力	64ビットバス
data_cnt_in	データカウント	入力	3ビットバス
data_vld_in	データ有効	入力	信号
ctrl_vld_in	制御有効	入力	信号
wait_upstream	上流で待ち	出力	信号
data_out	出力データ	出力	64ビットバス
data_cnt_out	データカウント	出力	3ビットバス
data_vld_out	データ有効	出力	信号
ctrl_vld_out	制御有効	出力	信号
wait_dnstrm	下流で待ち	入力	信号

この表における信号方向エントリは、ファームウェアモジュールの見地からリストアップされる。このため、信号(4)から(8)まで(「data\_in」、「data\_cnt\_in」、「data\_vld\_in」、「ctrl\_vld\_in」、および「wait\_upstream」)は、上流のファームウェアモジュール(例えば、通信パス132)に対するインタフェース信号を含む。信号(9)から(13)まで(「data\_out」、「data\_cnt\_out」、「data\_vld\_out」、「ctrl\_vld\_out」、および「wait\_dnstrm」)は、下流のファームウェアモジュール(例えば、通信パス134)に対するインタフェース信号を含む。

## 【0031】

信号「clk」は、ファームウェアソケットモジュールおよびFAMチェーンをクロック制御するのに使用される1ビットの入力信号である。

## 【0032】

信号「reset」は、ファームウェアソケットモジュールおよびFAMチェーンを起動時の状態にリセットするのに使用される1ビットの入力信号である。

## 【0033】

信号「module\_ID」は、各ファームウェアモジュールに、FAMチェーンにおけるそのモジュールの位置を知らせる6ビットの入力バスである。

## 【0034】

信号「data\_in」は、ターゲットデータおよびコマンドデータをFAMチェーンに転送するのに使用される64ビットのバスである。注目すべきこととして、データとコマンドの両方を転送するのに同一のバスを使用することにより、コマンドとターゲットデータの間の同期が、ほとんど複雑さなしに保たれることが可能である。

## 【0035】

信号「data\_cnt\_in」は、入力バス「data\_in」上の8バイトのうちどれだけが、意味のあるデータを表すかを決定するのに使用される3ビットのバスである。好ましくは、1-7という値は、文字どおり解釈され、0という値は、8バイトすべてが有効であることを意味する。

## 【0036】

信号「data\_vld\_in」は、「data\_in」バス上、および「data\_cnt\_in」バス上のデータが、所与のクロックサイクル上で有効であることを示すの

に使用される 1 ビットの入力信号である。

【0037】

信号「ctrl\_vld\_in」は、「data\_in」バス上のデータが、所与のクロックサイクル上で有効なコマンド（制御）情報であることを示すのに使用される 1 ビットの入力信号である。同一の F A M に入る「ctrl\_vld\_in」信号と「data\_vld\_in」信号は、同時にアサートされてはならないことは注目に値する。

【0038】

信号「wait\_upstrm」は、現在のファームウェアモジュールにデータをプッシュすることを停止するように上流のファームウェアモジュールに告げるのに使用される 1 ビットの出力信号である。好ましくは、ファームウェアモジュールは、待機がアサートされた後、2 つのデータ値を吸収するように構成される。

10

【0039】

信号「data\_out」は、所与の F A M からデータとコマンドの両方を転送するのに使用される 64 ビットの出力バスである。この場合も、データとコマンドの両方を転送するのに同一のバスが使用されるため、コマンドとターゲットデータの間の同期は、各 F A M が、受け取られたコマンドおよびターゲットデータに対して動作するので、F A M チェーンを通して保たれることが可能である。

【0040】

信号「data\_cnt\_out」は、出力バス「data\_out」上の 8 バイトのうちどれだけが、意味のあるデータを表すかを決定するのに使用される 3 ビットバスである。好ましくは、1 - 7 という値は、文字どおり解釈され、0 という値は、8 バイトすべてが有効であることを意味する。

20

【0041】

信号「data\_vld\_out」は、「data\_out」バス上、および「data\_cnt\_out」バス上のデータが、所与のクロックサイクル上で有効であることを示すのに使用される 1 ビットの出力信号である。

【0042】

信号「ctrl\_vld\_out」は、「data\_out」バス上のデータが、所与のクロックサイクル上で有効なコマンド（制御）情報であることを示すのに使用される 1 ビットの出力信号である。同一の F A M から出る「ctrl\_vld\_out」信号と「data\_vld\_out」信号は、同時にアサートされてはならないことは注目に値する。

30

【0043】

信号「wait\_dnstrm」は、ファームウェアソケットモジュールまたは下流のモジュールが、その時点でデータを吸収することができないことを F A M に示すのに使用される 1 ビットの入力信号である。好ましくは、「wait\_dnstrm」がディアサートされるまで、その F A M からそれ以上データが送り出されてはならない。

【0044】

F A M チェーンにおけるいくつかの隣接する F A M は、F A M サブセットを形成することが可能であり、この F A M サブセットは、独自の入口点と出口点を有することは注目に値する。そのような事例において、前掲の表において説明される信号インタフェースは、この F A M サブセットの入口点および出口点に関して保持されるが、異なる通知インタフェースが、オプションとして、F A M サブセットの F A M 間において内部で使用されることも可能である。また、F A M 自体、ファームウェアアプリケーションサブモジュールのチェーンから成ることが可能であり、これらのサブモジュール間の通信は、オプションとして、前掲の表において説明されるのとは異なる通知インタフェースに従うことも可能であることも注目に値する。

40

【0045】

ファームウェアモジュールインタフェース 214 - 論理層

前述した通り、64 ビットの「data\_in」バス上、および 64 ビットの「data

50

a\_\_out」バス上のファームウェアモジュールインタフェースを介して伝送される情報は、アプリケーションデータまたはコマンド（制御）情報であることが可能である。ファームウェアソケットモジュール120と個々のFAM130iはともに、コマンドとデータの順序付けを保ち、したがって、この順序は、FAMチェーン全体にわたって保たれる。データである、「data\_\_in」バスおよび「data\_\_out」バスに沿ってチェーンを下って流れる情報（ファームウェアソケットモジュールからチェーンにおける第1のFAMに、1つのFAMから次の下流のFAMに、チェーンにおける最終のFAMからファームウェアソケットモジュールに）は、データチャンネルと呼ばれる。コマンド（制御）情報である、「data\_\_in」バスおよび「data\_\_out」バスに沿ってチェーンを下って流れる情報は、コマンドチャンネルと呼ばれる。コマンドチャンネルを介して伝送されるコマンドは、通常、消費されないが、解釈されて（適宜）、下流のモジュールに送られる。この結果、コマンドチャンネルに入る多くのコマンドは、ソフトウェアに送り返される。

10

#### 【0046】

コマンドチャンネル上で、個々のコマンドに関する好ましいフォーマットは、コマンドが、以下のフィールドを有する64ビット長を有することである。すなわち、コマンド（16ビット）、誤り（1ビット）、同期（1ビット）、予約済み（8ビット）、モジュールID（6ビット）、およびパラメータ（32ビット）である。

#### 【0047】

コマンドフィールドは、好ましくは、2つのASCII文字（例えば、「QY」が、クエリコマンドであることが可能である）として符号化されるが、そうである必要はなく、任意の16ビットの値が、有効である。コマンドチャンネルに入ると、誤りビットは、クリアである。任意のモジュールが、その後、適宜、誤りビットを設定することができ、次に、誤りビットは、コマンドが、FAMチェーンおよびファームウェアソケットモジュールを抜けると、ソフトウェアに戻るよう通信される。同期ビットは、好ましくは、アウトバウンドのコマンドおよびデータの同期を、コマンドおよびデータがソフトウェアに戻る途中に制御するように設定される。

20

#### 【0048】

モジュールIDフィールドは、そのコマンドに応答すべき特定のファームウェアモジュールを特定する。好ましくは、モジュールIDに関する0という値は、すべてのモジュールが、そのコマンドに応答すべきことを意味する（FAMのすべてだけでなく、ファームウェアソケットモジュールも含む）。また、モジュールIDに関する1という値は、ファームウェアソケットモジュールを除くすべてのモジュールが、そのコマンドに応答すべきことを意味することが好ましい。好ましくは、2というモジュールID値は、ファームウェアソケットモジュールだけのために取っておかれ、3以上のモジュールID値は、特定のFAMに向けられる。各FAMは、ファームウェアモジュールインタフェースの信号層に関連して説明される入力信号module\_\_IDのお陰で、そのFAM自らのモジュールID値を知る。

30

#### 【0049】

ファームウェアモジュールインタフェースに関する好ましいグローバルコマンドには、リセットコマンド、クエリコマンド、クエリ応答コマンド、パススルーコマンド、データ開始コマンド、およびデータ終了コマンドが含まれる。

40

#### 【0050】

リセットコマンド（RS）は、コマンドのモジュールIDフィールドによって示される通り、FAMチェーン全体、または個々のFAMをリセットするように動作する。好ましくは、このパラメータフィールドは、このコマンドに関して使用されない。

#### 【0051】

クエリコマンド（QY）は、モジュールにクエリを行って、モジュールの現在の状態を評価するように動作する。モジュールは、1つまたは複数のクエリ応答（QR）コマンドで応答しなければならない。モジュールが、そのモジュールのクエリ応答で応答した後、

50

そのモジュールは、元の Q Y コマンドを、チェーンにおける次のモジュールに送る。好ましくは、パラメータフィールドは、このコマンドに関して使用されていない。

【 0 0 5 2 】

モジュールが、Q Y コマンドを受け取ると、1 つまたは複数のクエリ応答コマンド ( Q R ) が、モジュールによって生成される。クエリ応答を受け取ると、F A M は、その応答を単に下流に転送する。好ましくは、Q R コマンドに関するパラメータフィールドは、モジュール固有であり、この 3 2 ビットのフィールドは、2 つのサブフィールド、すなわち、どのようなタイプの情報が送られているかを示すタグである最上位の 8 ビット、および所与のタグに関連するデータである最下位の 2 4 ビットに分割される。

【 0 0 5 3 】

パススルーコマンド ( P S ) は、モジュールが、データを、変更なしにパススルーすべきか、または通常の実行動作を実行すべきかをモジュールに知らせる。パススルーモードは、コマンド処理またはコマンド転送に影響を与えず、このモードの影響は、データチャネルに対してだけに限られる。好ましくは、パラメータフィールドの最下位ビット ( ビット 0 ) が、パススルーモードをイネーブル / ディセーブルにし、1 というパススルービット値が、パススルーモードをイネーブルにし、0 というパススルービット値が、パススルーモードをディセーブルにする。

【 0 0 5 4 】

データ開始コマンド ( S D ) は、データストリームの開始に ( 通常、フィールド境界上に ) マークを付けるのに使用される。データ終了コマンド ( E D ) は、データストリームの終了に ( この場合も、通常、フィールド境界上に ) マークを付けるのに使用される。S D コマンドおよび E D コマンドに関するパラメータフィールドは、好ましくは、使用されないが、一部の事例において、パラメータフィールドは、ファイル I D またはフレーム I D を指定するのに使用されることが可能である。

【 0 0 5 5 】

F A M チェーンを通るコマンド伝搬は、好ましくは、以下の通り進む。すなわち、( 1 ) グローバルコマンド ( モジュール I D 値が 0 または 1 であるコマンド ) が、F A M チェーンにおける各 F A M によって順方向に伝搬され、( 2 ) F A M は、モジュール I D 値が、異なる F A M のために設定されているコマンドを順方向に伝搬し、( 3 ) F A M は、同期ビットが設定されているコマンドを順方向で伝播する。

【 0 0 5 6 】

ファームウェアソケットモジュールに固有である、またはそうである可能性がある好ましいコマンドには、データカウントコマンド、リセットコマンド、およびクエリ応答コマンドが含まれる。

【 0 0 5 7 】

データカウントコマンド ( D C ) は、ファームウェアソケットモジュールが前回にリセットされて以来、どれだけのデータが、F A M チェーンを通過したかをファームウェアソケットモジュールに尋ねるのに使用されることが可能である。ファームウェアソケットモジュールに 1 つの D C コマンドが送信されることは、3 つの D C コマンドが戻されることをもたらす。最初の応答の D C コマンドは、F A M チェーンに入ったバイト数を特定する。第 2 番の応答の D C コマンドは、F A M チェーンを出たデータのバイト数を特定する。第 3 番の応答の D C コマンドは、戻される元の送られた D C コマンドである。好ましくは、S D コマンドまたは R S コマンドは、データカウントをリセットするように動作する。パラメータフィールドに関して、送られる D C コマンドのパラメータフィールドは、好ましくは、無視されるのに対して、応答の D C コマンドのパラメータフィールドは、好ましくは、前述した通り、関係のあるバイト数を特定する。

【 0 0 5 8 】

R S ( リセットコマンド ) フォーマットについては、グローバルコマンドに関連して前段で説明した。ファームウェアソケットモジュールは、R S コマンドを受信すると、好ましくは、F A M チェーン全体をリセットし、さらに、あらゆる入力キューおよび出力キュー

10

20

30

40

50



ーをクリアする。ファームウェアソケットモジュールによってカードから戻されていない可能性があるデータは、R S コマンドによってクリアされ、このため、アプリケーションは、ファームウェアソケットモジュールにリセットコマンドを送信する前に、予期されるすべてのデータを受信していることを確実にしなければならない。このことは、同期ビットセットを有するコマンドを送ること、またはファームウェアソケットモジュールのダブルレジスタの中のフラッシュビットのストローピングを行うことによって達せられることが可能である。

#### 【 0 0 5 9 】

また、クエリ応答コマンド ( Q R ) フォーマットについても、グローバルコマンドに関連して前段で説明した。好ましくは、ファームウェアソケットモジュールが、Q R コマンドを発行すると、ファームウェアソケットモジュールのパラメータフィールドは、以下の表 2 の中で以下の通り設定される。すなわち、

#### 【 表 2 】

表2:

タグ(8ビット)	値(24ビット)
0	モジュールタイプ(「ファームウェアソケットモジュール」)
1	バージョン番号
2	ビルド(改訂番号)

#### 【 0 0 6 0 】

F A M に固有であるコマンドは、各 F A M のデータ処理動作に応じて変化する。例えば、ターゲットデータのストリームとパターンがマッチングされる検索動作を実行するように構成された F A M に適用可能なコマンドのセットには、好ましくはデータストリームの検索が照らし合わせて行われるパターンを定義するコマンド、パターンとの一致が、データストリームの中で見つかった場合に、データストリームからどれだけのコンテキストが戻されるべきか (例えば、データストリーム内の一致の前と後の X バイトを戻す) を特定するコマンド、所与のパターンに関して許されるミスマッチの数を定義するコマンド (例えば、パターンが、長さ 1 0 である場合、K というミスマッチ値は、1 0 のパターン文字のいずれか ( 1 0 - K ) が、データストリーム内で一致した場合に、一致が生じることを許す)、マッチが宣言されるのに、パターンの中のいずれのビットが、データストリームの中で一致を見出さなければならないかを定義するコマンド、検索されるべきデータ内のファイルを定義するコマンド、および検索されなければならないファイル内のデータのサブセットを定義するコマンドが含まれる。そのようなコマンドのパラメータフィールド、またはオプションとして、他のフィールドが、F A M によって実行されるパターンマッチング動作の、これらの態様を定義するのに使用されることが可能である。また、パターンマッチング動作を実行するように構成された F A M からの Q R コマンドは、好ましくは、値が、F A M タイプの識別、F A M タイプのバージョンの識別、F A M タイプのビルドの識別、および F A M によってサポートされるパターンの数の識別に対応するタグで応答する。

#### 【 0 0 6 1 】

別の例として、暗号化動作または解読動作を実行するように構成された F A M に適用可能なコマンドのセットには、好ましくは、暗号化 / 解読動作において使用されるべきキーを特定するコマンドが含まれる。

#### 【 0 0 6 2 】

F A M によって実行されることが可能な前段でリストアップされる他のデータ処理動作に加え、可能なデータ処理動作には、固定長のレコードを取り込み、選択されているレコードのバイトだけを送るフィールド選択動作 (そのような F A M へのコマンドは、いずれのバイトが選択されるべきかを示す)、サイズが、F A M を通して処理されるデータサイ

ズ（本明細書で説明される好ましい実施例において 8 バイト）の倍数ではないレコードにパディングを追加するレコードゲート動作（そのような F A M へのコマンドは、所望されるレコードサイズを示す）、レコードからパディングバイトを除去する逆レコードゲート動作（そのような F A M へのコマンドは、除去されるべきレコードのバイト数を示す）、および 1 つまたは複数のパターンに対する一致を、固定長のレコードの指定された列境界内で検索するレコード選択動作（コマンドが、前述したパターンマッチングコマンドと同様であるが、指定された列境界内で行われる検索を制限するのに、レコードがどのようにマスクされるかを特定するコマンドも含む動作）も含まれる。

#### 【 0 0 6 3 】

好ましくは、F P G A 上に配備される F A M は、1 つのタイプのデータ処理動作を実行することに専用であり、コマンド信号は、そのタイプのデータ処理動作の特定の態様を定義するように作用する。例えば、図 7 に示される通り、F A M チェーンにおける各 F A M は、或るタイプのデータ処理動作に専用であり、例えば、第 1 の F A M は、伸張動作を実行するように構成され、第 4 番の F A M は、1 つのタイプの検索動作を実行するように構成され、第 5 番の F A M は、別のタイプの検索動作を実行するように構成され、最後から 2 番目の F A M は、暗号化動作を実行するように構成される。適切なコマンドを通して、各 F A M のデータ処理動作の詳細が、制御されることが可能である（例えば、「S m i t h」という語を検索するように第 4 番の F A M を構成する、またはキー A を使用して暗号化動作を実行するよう最後から 2 番目の F A M を構成する）。F A M が、異なるタイプのデータ処理動作を実行するように完全に再プログラミングされるべき場合、F P G A が、この目的を達するように再構成されることが好ましいが、そのように行われる必要はない。

#### 【 0 0 6 4 】

さらに、F A M チェーンにおける各 F A M を適切にオンまたはオフにすることにより、特化された処理動作が、F A M チェーンによって達せられることが可能である。例えば、データが、暗号化されたフォーマットで格納される所与のパターンに関してデータストリーム内で検索を実行するのに、（ 1 ）検索 F A M の 1 つをオンにし、その F A M を、その所与のパターンを使用して構成し、（ 2 ）解読 F A M をオンにし、その F A M を、データストリームを解読する適切なキーを使用して構成し、（ 3 ）チェーンにおける他のすべての F A M をオフにするコマンドが、F A M チェーンに送られることが可能である。そのような事例において、暗号化されたデータストリームは、第 2 番の F A M によって解読され、解読されたデータストリームは、その所与のパターンに関して検索 F A M の 1 つによって検索される。

#### 【 0 0 6 5 】

##### ハードウェア / ソフトウェアインタフェース 2 1 2

好ましくは、ファームウェアソケットモジュールは、ドライバレベルシステムソフトウェア 2 0 4 と連携して使用される場合、システムのメモリ空間に / からデータを転送するバックエンドを有する X i l i n x（登録商標） P C I - X を使用する。このバックエンドインタフェースの性質について、本明細書でより詳細に説明する。

#### 【 0 0 6 6 】

X i l i n x（登録商標） P C I - X C o r e V e r s i o n 5 . 0 は、P C I - X 規格によって説明されているインタフェースから、X i l i n x（登録商標） L o g i C o r e P C I - X D e s i g n G u i d e V e r s i o n 5 . 0 および X i l i n x（登録商標） L o g i C o r e P C I - X I m p l e m e n t a t i o n G u i d e V e r s i o n 5 . 0 によって説明されている X i l i n x（登録商標）によって設計されたインタフェースへの変換を扱う。好ましくは、構成パラメータは、P C I - X 接続プリント回路基板 3 0 0 が、6 4 ビットおよび 1 3 3 M H z 対応であることを示すように設定される。システム B I O S が、デバイスの速度を 1 0 0 M H z に設定するか、または 1 3 3 M H z に設定するかは、1 3 3 M H z 能力ビット、ならびに P C I X C A P ピン上の物理的電圧および時間遅延に依存する。F P G A 3 0 2 上に配備

されたモジュールが、100MHzだけで実行される場合、PCI-XCAPピンは、このことを示すように設定されなければならない。また、PCI-X構成は、3つまでの64ビットBAR（ベースアドレスレジスタ）、または6つまでの32ビットBARを使用するように設定されることが可能である。好ましくは、ファームウェアソケットモジュール120は、64ビットBARとして構成された第1のBARを使用して、デバイスドライバソフトウェア204との通信のためにモジュール120のデバイスレジスタをマップする。

#### 【0067】

ファームウェアソケットモジュール120とデバイスドライバソフトウェア204の間の通信の大部分は、好ましくは、3つの記述子プールセットを介して行われる。図5に示される通り、これらの記述子プールに関するバッファは、カーネル-カード入力記述子プールバッファ500、データに関するカード-カーネル出力記述子プールバッファ504と、コマンドに関するカード-カーネル出力記述子プールバッファ506とを含む。これらの記述子プールバッファは、好ましくは、OSカーネル/デバイスドライバソフトウェア204によって管理されるメモリ空間の中（例えば、RAM110内）に存在する。バッファ500、504、および506の中の各エントリは、記述子を含み、各記述子は、そのバッファ内の複数のアドレスにわたることが可能である。

#### 【0068】

後段で説明されるバイトカウントフィールドおよび他のフラグに加え、記述子はそれぞれ、好ましくは、ソフトウェアからハードウェアへの（入力記述子に関する）、またはハードウェアからソフトウェアへの（出力記述子に関する）データ移動に関するメモリロケーションの物理アドレスおよび仮想アドレスを含む。このため、記述子プールバッファの中のエントリは、ファームウェアソケットモジュールが、データを読み取る、またはデータを書き込む位置へのポインタの役割をする。

#### 【0069】

ファームウェアソケットモジュールが、データ（コマンドとターゲットデータの両方）を読み取る/書き込むメモリ空間502は、コンピュータシステム、およびコンピュータシステムに接続された物理的にアドレス指定可能な任意のデータ記憶デバイスの物理的アドレス空間内の任意のメモリであることが可能である。このため、メモリ空間502は、コンピュータシステムのマザーボード（例えば、RAM110）に限定される必要がない。例えば、カードのメモリ304が、ドライバレベルソフトウェア204に利用可能にされた場合、コマンドおよびターゲットデータを格納するためのメモリ空間502は、メモリ304を含むことが可能である。また、メモリ空間502は、データストア104内、または他の何らかのシステム接続された記憶デバイス内のアドレスを、データストア104、またはそのような記憶デバイスのアドレス空間が、ドライバレベルソフトウェアに利用可能にされた場合（例えば、データストア104に関するディスクコントローラ106を介して）、含むことが可能である。

#### 【0070】

ハードウェア/ソフトウェアインタフェース212に関する好ましい通常の動作は、以下の通りである。すなわち、

1) ソフトウェアデバイスドライバ204が、FAMに送られるべきコマンドおよびデータを、メモリ空間502の中のバッファのセットの中に入れる。好ましくは、ドライバレベルソフトウェア204は、再構成可能なロジックデバイスを介してストリーミングされるべきターゲットデータを、メモリ空間502の隣接するアドレスの中に単一のセットとして格納し、これにより、ターゲットデータがFPGAに読み込まれることが可能な速度が向上するだけでなく、そのような隣接して格納されたターゲットデータをポイントする連続する記述子が、場合により、記述子のバイトカウントフィールドを適切に更新することによって、より少ない記述子に集約されることが可能であるという点で、柔軟性をもたらしもする。

2) デバイスドライバ204が、次に、メモリ空間502内のバッファの内容についてフ

10

20

30

40

50

ファームウェアソケットモジュールに知らせる入力記述子を構築する。好ましくは、1つの記述子は、ターゲットデータのMバイトの各コマンドまたは各セグメントに関して使用され、Mのサイズは、様々な要因に応じて可変である。Mのサイズは、デフォルトで32キロバイトにされることが可能である。しかし、前述した通り、この値は、依然として、各記述子に関するカードを介して大量のデータを移動しながら、バッファセットアップ、記述子管理などに関連するプロセッサオーバヘッドを最小限に抑える要望を考慮して、ソフトウェアによって可変であることが可能である。また、デバイスドライバ204は、好ましくは、FAMから戻ってくるデータおよびコマンドをどこに置くかをファームウェアソケットモジュールに告げる出力記述子を構築する。入力記述子は、デバイスドライバ204によって、入力記述子プールバッファ500の中に格納される。データに対応する出力記述子は、デバイスドライバ204によって、出力記述子プールバッファ504の中に格納され、コマンドに対応する出力記述子は、デバイスドライバ204によって、出力記述子プールバッファ506の中に格納される。

10

3) デバイスドライバ204は、次に、新たな入力記述子、および新たな出力記述子が、通信パス510上でドアベル信号を介して利用可能であることをファームウェアソケットモジュールに知らせる。

4) ファームウェアソケットが、次に、バッファ500の中の第1の入力記述子を読み取り、FAMチェーンに送られるべきデータ(コマンドまたはターゲットデータ)が格納されるロケーションを特定し、そのロケーションからFAMチェーンへのデータ(コマンドまたはターゲットデータ)のDMA転送を実行し、バッファ500の中の次の入力記述子に進む。このため、コマンドおよびターゲットデータは、入力記述子プールバッファ500内で設定されているのと同じ順序でFAMチェーンに供給される。

20

5) ファームウェアソケットモジュールが、次に、FAMチェーンを出るコマンドを、バッファ506の中のコマンド出力記述子によって指定されたメモリ空間502c内のバッファに送る。また、ファームウェアソケットモジュールは、FAMチェーンから出るデータを、バッファ504の中のデータ出力記述子によって指定されたメモリ空間502b内のバッファに送る。

6) ファームウェアソケットモジュールは、次に、コマンドおよび/またはデータが利用可能であることをデバイスドライバに通知する(割り込みを介して)。

7) ソフトウェアが、次に、適宜、戻されたデータおよび/またはコマンドに対処する適切なデータハンドラまたはコマンドハンドラを呼び出す(このハンドラは、ソフトウェアライブラリインタフェース210を介して制御される)。

30

#### 【0071】

ソフトウェアから再構成可能なロジックデバイスに情報を送る際、すべての転送(コマンドとデータの両方)は、好ましくは、バッファ500の中に格納された共通の入力記述子によって管理される。FAMチェーンの中の個々のFAMが、データおよびコマンドを並べ替えない限り、バッファ500の中の入力記述子の順序によって定義されるデータおよびコマンドの順序は、再構成可能なロジックデバイス内の処理全体にわたって保たれる。

#### 【0072】

40

再構成可能なロジックデバイスからソフトウェアに戻るよう送られるデータおよびコマンドに関して、厳密な順序付けは、必ずしも要求されない。厳密な同期が、プロセス全体(ソフトウェアへのコマンドおよびデータの戻しを含む)にわたって所望される場合、デバイスドライバソフトウェアは、個別のコマンドの中で同期ビットを設定することができる。同期ビットセットを有するコマンドが、FAMチェーンの出口点からファームウェアソケットモジュールに到着すると、ファームウェアソケットモジュールは、好ましくは、以下の機能を実行する。すなわち、(1)データチャネルバッファをメモリにフラッシュし、(2)データ出力記述子の中で設定された同期フラグを有するデータチャネルにおけるNull(空の)バッファを作成し、(3)このデータ出力記述子が、メモリにフラッシュされることを確実にし、(4)コマンドチャネルバッファ(コマンド自体の中で同

50

期ビットが設定されている)をフラッシュする。

【 0 0 7 3 】

ファームウェアソケットモジュール 1 2 0 は、好ましくは、前段で特定された(さらに、64ビットBARとして構成されている)第1のBARのメモリ領域の中にマップされた16個のデバイスレジスタを有する。各デバイスレジスタの物理的地址は、ベースアドレスに或るオフセットを足したアドレスである。これらのレジスタ、およびBARからの、これらのレジスタのオフセットが、以下の表3にリストアップされる。すなわち、

【表3】

表3:

BARからのオフセット	デバイスレジスタ
0x00	ファームウェアID
0x08	FPGA情報
0x10	デバイスステータス
0x18	実装されたRAMサイズ
0x20	ドアベル
0x28	割り込みステータス
0x30	Data-to-Cardアドレス
0x38	Data-to-Cardカウント
0x40	Data-to-Cardネクスト
0x48	Data-to-Kernelアドレス
0x50	Data-to-Kernelカウント
0x58	Data-to-Kernelネクスト
0x60	Data-to-Kernelアドレス
0x68	Data-to-Kernelカウント
0x70	Data-to-Kernelネクスト
0x78	パラメータ

【 0 0 7 4 】

ファームウェアIDは、2つの値から成る64ビットの読み取り専用レジスタである。最下位の32ビットは、ファームウェアID番号であり、最上位の32ビットは、そのIDによって示される特定のファームウェアの改訂番号である。

【 0 0 7 5 】

FPGA情報レジスタは、2つの情報、すなわち、FPGAタイプおよびFPGAサイズを含む64ビットの読み取り専用レジスタである。FPGAタイプは、FPGAのタイプを示す4つのASCII文字(32ビット)である。例えば、FPGAデバイスが、Xilinx Virtex(登録商標) - IIデバイスである場合、FPGAタイプは、「xc2v」である。FPGAサイズは、どのようなサイズのFPGAが使用されているかを示す32ビットの整数である。例えば、FPGAデバイスが、Xilinx Virtex(登録商標) - II 4000である場合、FPGAサイズは、4000である。FPGAタイプは、FPGA情報レジスタの最下位の32ビットであるのに対して、FPGAサイズは、レジスタの最上位の32ビットである。

【 0 0 7 6 】

デバイスステータスレジスタは、ハードウェアデバイスに接続されたハードウェアシステムについての2つの情報を含む64ビットの読み取り専用レジスタである。最下位の32ビットは、ハードウェアデバイスに接続されたハードウェアシステムのタイプを示す。以下の表4は、デバイスタイプ番号付けの例を示す。

【表 4】

表4:

デバイスタイプ番号	接続されたシステムのタイプ
0	無効なデバイスタイプ
1	スタンドアロンのシステム (接続されたデバイスなし)
2	SCSI
3	IDE
4	FibreChannel
5	iSCSI
6	予約済み

10

デバイスステータスレジスタの最上位の 32 ビットは、ハードウェアシステムに接続されたデバイスの数を示す整数である。

【0077】

実装された RAM サイズは、FPGA デバイスに接続されたメモリ 304 の量を（キロバイト単位で）示す 64 ビットのレジスタである。

【0078】

20

ドアベルレジスタは、ドライバレベルソフトウェア 204 とファームウェアソケットモジュール 120 の間の通信のために使用されるフラグを含む 64 ビットの読み取り / 書き込みレジスタである。ドアベルフラグ、およびこれらのフラグの機能は、以下の表 5 にリストアップされる。

【表 5】

表5:

ドアベルビット	機能の名前
0	実行
1	停止
2	ファームウェアソケットモジュールリセット
3	フラッシュ
4	ストローブデバッグ
5	クロックカウンタリセット
6-63	予約済み

30

【0079】

実行ビットは、ドライバレベルソフトウェア 204 によって、実行を開始する、または再開するようファームウェアソケットモジュールに告げるのに設定される。このビットは、ハードウェアによって確認されると、ファームウェアソケットモジュールによってクリアされる。停止ビットは、ドライバレベルソフトウェア 204 によって、実行を停止する、または一時停止するようファームウェアソケットモジュールに告げるのに設定される。このビットは、ハードウェアによって確認されると、ファームウェアソケットモジュールによってクリアされる。リセットビットは、ドライバレベルソフトウェア 204 によって、ファームウェアソケットモジュール、および FAM チェーンにおける FAM のすべてをリセットするのに設定される。このビットは、ハードウェアによって確認されると、ファームウェアソケットモジュールによってクリアされる。フラッシュビットは、ドライバレベルソフトウェア 204 によって、ソフトウェアを宛先とするコマンドおよびデータを保

40

50

持するファームウェアソケットモジュール上のアウトバウンドバッファをフラッシュするのに設定される。このビットは、ハードウェアによって確認されると、ファームウェアソケットモジュールによってクリアされる。ストローブデバッグビットは、デバッグ目的で使用されることが可能なビットである。クロックカウンタリセットビットは、P C I - Xバスのクロック速度を決定するために使用されるファームウェアソケットモジュールの中のカウンタをリセットする。P C I - Xバスのクロック速度は、ファームウェアソケットモジュールパラメータレジスタの上位の9ビットを読み取ることによって決定されることが可能である。

#### 【 0 0 8 0 】

割り込みステータスレジスタは、以下の表 6 にリストアップされる割り込み条件についてドライバレベルソフトウェア 2 0 4 に通知するのに使用される 6 4 ビットの読み込み / 書き込みレジスタである。割り込み条件が設定されるといつでも、オペレーティングシステムに割り込みを行う割り込みが、P C I - Xバス上でトリガされ、ドライバレベルソフトウェア 2 0 4 によって応答が行われなければならない。ドライバレベルソフトウェア 2 0 4 が、割り込みステータスレジスタを読み取る前に、複数の割り込み条件が生じると、ソフトウェアが、割り込みステータスレジスタを読み取った際に、複数のビットが、設定される。割り込みステータスレジスタは、読み取られると、クリアされる。

#### 【 表 6 】

表6:

割り込みビット	名前
0	Data-to-Card入力記述子バッファ空
1	Data-to-Card誤り
2	Data-to-Card入力記述子バッファ完了
3	Data-to-Kernel出力記述子バッファ満杯
4	Data-to-Kernel誤り
5	Data-to-Kernel出力記述子バッファレディ
6	見つかった結果なし
7	Command-to-Kernel出力記述子バッファ満杯
8	Command-to-Kernel誤り
9	Command-to-Kernel出力記述子バッファレディ
10-63	予約済み

#### 【 0 0 8 1 】

ファームウェアソケットモジュールは、データの入力記述子バッファを読み取ろうと試みて、次の入力記述子バッファが、読み取られる準備ができていない（バッファ 5 0 0 の中でキュー状態にある次の入力記述子のレディビットを調べることにより）場合、割り込みビット 0 を設定する。割り込みビット 0 が、書き込まれた場合、ドライバレベルソフトウェア 2 0 4 は、ファームウェアソケットモジュールが、入力記述子プールバッファを再び読み取ろうと試みる前に、ドアベルレジスタの実行ビットに書き込まなければならない。

#### 【 0 0 8 2 】

割り込みビット 1 は、ファームウェアソケットモジュールへの入力として定められているデータの入力記述子を読み取る際、誤りが生じたことを知らせる。

#### 【 0 0 8 3 】

入力記述子プールバッファ 5 0 0 の割り込みフラグが、設定された場合、ファームウェアソケットモジュールは、入力データの入力記述子バッファを読み取ることを終えた後、割り込みビット 2 を設定する。

#### 【 0 0 8 4 】

ファームウェアソケットモジュールは、カーネルに出力すべきデータを有するが、次の `Data - to - Kernel` 出力記述子バッファが、利用できない（バッファ 504 の中でキュー状態にある次のデータ出力記述子のレディビットを調べることにより）場合、割り込みビット 3 を設定する。ファームウェアソケットモジュールが、この割り込みを設定すると、ドライバレベルソフトウェア 204 は、ファームウェアソケットモジュールが、データを再び出力しようとする前に、`Data - to - Kernel` 出力記述子バッファを利用可能にした後、ドアベルレジスタの実行ビットに書き込まなければならない。

【0085】

割り込みビット 4 は、ファームウェアソケットモジュールが、出力記述子バッファにデータを書き込もうと試みた際に、誤りが生じたことを知らせる。

10

【0086】

ファームウェアソケットモジュールは、`Data - to - Kernel` 出力記述子バッファにデータを書き込み、記述子を閉じるといつでも（すなわち、ファームウェアソケットモジュールは、データを書き込むために次の `Data - to - Kernel` 出力記述子バッファに進んでいる）、割り込みビット 5 を設定する。

【0087】

割り込みビット 6 は、ドアベルレジスタの中でフラッシュビットが設定されている場合、戻されるべき出力データが全く存在しないことを示すように定義される。

【0088】

ファームウェアソケットモジュールは、カーネルに出力すべきコマンドデータを有するが、次の `Command - to - Kernel` 出力記述子バッファが、利用できない（バッファ 506 の中でキュー状態にある次のコマンド出力記述子のレディビットを調べることにより）場合、割り込みビット 7 を設定する。ファームウェアソケットモジュールが、この割り込みを設定すると、ドライバレベルソフトウェア 204 は、ファームウェアソケットモジュールが、コマンドデータを再び出力しようとする前に、`Command - to - Kernel` 出力記述子バッファを利用可能にした後、ドアベルレジスタの実行ビットに書き込まなければならない。

20

【0089】

割り込みビット 8 は、ファームウェアソケットモジュールが、コマンド出力記述子バッファにコマンドデータを書き込もうと試みた際に、誤りが生じたことを知らせる。

30

【0090】

ファームウェアソケットモジュールは、`Command - to - Kernel` 出力記述子バッファにデータを書き込み、記述子を閉じるといつでも（すなわち、ファームウェアソケットモジュールは、コマンドデータを書き込むために次の `Command - to - Kernel` 出力記述子バッファに進んでいる）、割り込みビット 9 を設定する。

【0091】

`Data - to - Card` アドレスレジスタは、`Data - to - Card` 入力記述子プールバッファ 500 の先頭の物理的アドレスを示すのに使用される 64 ビットの読み取り / 書き込みレジスタである。ドライバレベルソフトウェア 204 は、ドアベルレジスタの実行ビットに初めて書き込みが行われる前に、このレジスタに書き込まなければならない。

40

【0092】

`Data - to - Card` カウントレジスタは、データを読み取るために利用可能な `Data - to - Card` 入力記述子バッファの数を示すのに使用される 32 ビットの読み取り / 書き込みレジスタである。ドライバレベルソフトウェア 204 は、ドアベルレジスタの実行ビットに初めて書き込みが行われる前に、このレジスタに書き込まなければならない。このアドレスから読み取りが行われると、上位の 32 ビットが 0 で埋められた 64 ビットの値が、戻される。

【0093】

`Data - to - Card` ネクストレジスタは、読み取りが行われるべき次の `Data`

50



- t o - C a r d 入力記述子バッファに対する 3 2 ビットの読み取り / 書き込みポイントである。このポイントは、ファームウェアソケットモジュールとドライバレベルソフトウェアが同期していることを確実にするのに使用される。このアドレスから読み取りが行われると、上位の 3 2 ビットが 0 で埋められた 6 4 ビットの値が、戻される。

【 0 0 9 4 】

D a t a - t o - K e r n e l アドレスレジスタは、D a t a - t o - K e r n e l 出力記述子プールバッファ 5 0 4 の先頭の物理的地址を示すのに使用される 4 8 ビットの読み取り / 書き込みレジスタである。ドライバレベルソフトウェア 2 0 4 は、ドアベルレジスタの実行ビットに初めて書き込みが行われる前に、このレジスタに書き込まなければならない。このアドレスから読み取りが行われると、上位の 1 6 ビットが 0 で埋められた 6 4 ビットの値が、戻される。

10

【 0 0 9 5 】

D a t a - t o - K e r n e l カウントレジスタは、データを書き込むために利用可能な D a t a - t o - K e r n e l 出力記述子バッファの数を示すのに使用される 3 2 ビットの読み取り / 書き込みレジスタである。ドライバレベルソフトウェア 2 0 4 は、ドアベルレジスタの実行ビットに初めて書き込みが行われる前に、このレジスタに書き込まなければならない。このアドレスから読み取りが行われると、上位の 3 2 ビットが 0 で埋められた 6 4 ビットの値が、戻される。

【 0 0 9 6 】

D a t a - t o - K e r n e l ネクストレジスタは、書き込みが行われるべき次の D a t a - t o - K e r n e l 出力記述子バッファに対する 3 2 ビットの読み取り / 書き込みポイントである。このポイントは、ファームウェアソケットモジュールとドライバレベルソフトウェアが同期していることを確実にするのに使用される。このアドレスから読み取りが行われると、上位の 3 2 ビットが 0 で埋められた 6 4 ビットの値が、戻される。

20

【 0 0 9 7 】

C o m m a n d - t o - K e r n e l アドレスレジスタは、C o m m a n d - t o - K e r n e l 出力記述子プールバッファ 5 0 6 の先頭の物理的地址を示すのに使用される 4 8 ビットの読み取り / 書き込みレジスタである。ドライバレベルソフトウェア 2 0 4 は、ドアベルレジスタの実行ビットに初めて書き込みが行われる前に、このレジスタに書き込まなければならない。このアドレスから読み取りが行われると、上位の 1 6 ビットが 0 で埋められた 6 4 ビットの値が、戻される。

30

【 0 0 9 8 】

C o m m a n d - t o - K e r n e l カウントレジスタは、コマンドデータを書き込むために利用可能な C o m m a n d - t o - K e r n e l 出力記述子バッファの数を示すのに使用される 3 2 ビットの読み取り / 書き込みレジスタである。ドライバレベルソフトウェア 2 0 4 は、ドアベルレジスタの実行ビットに初めて書き込みが行われる前に、このレジスタに書き込まなければならない。このアドレスから読み取りが行われると、上位の 3 2 ビットが 0 で埋められた 6 4 ビットの値が、戻される。

【 0 0 9 9 】

C o m m a n d - t o - K e r n e l ネクストレジスタは、書き込みが行われるべき次の C o m m a n d - t o - K e r n e l 出力記述子バッファに対する 3 2 ビットの読み取り / 書き込みポイントである。このポイントは、ファームウェアソケットモジュールとドライバレベルソフトウェアが同期していることを確実にするのに使用される。このアドレスから読み取りが行われると、上位の 3 2 ビットが 0 で埋められた 6 4 ビットの値が、戻される。

40

【 0 1 0 0 】

パラメータレジスタは、ファームウェアソケットモジュールの中のプログラミング可能なパラメータを設定するのに使用される 6 4 ビットのレジスタである。以下の表 7 に示される通り、これらのパラメータのいくつかは、読み取り / 書き込みであるのに対して、他のパラメータは、読み取り専用である。

50

【表 7】

表7:

ビット	パラメータ	読み取り/書き込み
5-0	最大PCI-Xバーストサイズ	読み取り/書き込み
54-6	予約済み	読み取り/書き込み
55	バス速度有効	読み取り専用
63-56	PCI-Xバス速度	読み取り専用

10

## 【0101】

最大PCI-Xバーストサイズパラメータは、ファームウェアソケットモジュールが、PCI-Xバス112上で実行しようと試みる最大サイズトランザクションを設定するのに使用される。この最大トランザクションサイズは、パラメータレジスタの中の6ビットの値掛ける128であるが、このフィールドを0に設定することにより、最大トランザクションサイズが4096（PCI-X規格によって許される最大トランザクションサイズ）に設定される。

## 【0102】

パラメータレジスタのビット55は、このレジスタの上位の8ビットにおけるPCI-Xバス速度が有効であるかどうかを示すのに使用される（「1」は、有効を示し、「0」は、無効を示す）。

20

## 【0103】

パラメータレジスタの最上位の8ビットは、PCI-Xバスの計算速度をMHz単位で示す。バス速度有効ビット（ビット55）も、この値が有効であるか否かを確認するのに読み取られなければならない。PCI-Xバス速度値が、有効ではない場合、ドライバレベルソフトウェアは、短い時間、待ち、この値を再び読み取らなければならない。

## 【0104】

前述した通り、ファームウェアソケットモジュールとドライバレベルソフトウェア204の間の通信は、3つの記述子セット、すなわち、1つの入力記述子プールと、2つの出力記述子プール（1つは、コマンドに関し、他方は、データに関する）とを使用して行われる。各プールは、記述子バッファ（カーネルバッファ）についての情報を有するフラグおよびフィールドを含む独自の記述子セットを有する。

30

## 【0105】

各記述子は、好ましくは、以下の表8に示される通り、それぞれの8バイトの4つの部分に分割される32バイト長である。すなわち、

【表 8】

表8:

バイト数	フィールド
8	フラグ
8	バイトカウント
8	物理的地址
8	仮想アドレス

40

## 【0106】

フラグフィールドは、以下の表9に示される通り、記述子バッファについての情報を含む64ビットのフィールドである。

【表 9】

表9:

ビット	フラグ名
0	ビジー
1	コマンド/データ
2	Data-to-Kernel
3	Data-to-Card
4	レディ
5	完了
6	割り込み
7-63	予約済み

10

## 【0107】

ビジービットは、ドライバレベルソフトウェア204によって、記述子バッファが使用中であることを示すのに設定される。

## 【0108】

コマンド/データビットは、バッファが、コマンド情報またはデータを含むかどうかをファームウェアソケットモジュールに告げるのに使用される。バッファが、コマンド情報を含む場合、このビットは、設定されなければならない、含まない場合、このビットは、クリアされなければならない。ファームウェアソケットモジュールは、Data-to-Card入力記述子バッファ上で、このビットを読み取り、そのバッファの中のすべてのデータに適切にマークを付ける（コマンドまたはデータとして）。

20

## 【0109】

Data-to-Kernelビットは、バッファの中のデータが、ドライバレベルソフトウェア204（カーネル）を宛先としているか否かを示す。このビットが設定されている場合、データは、カーネルを宛先として、ファームウェアソケットモジュールから出力される。ファームウェアソケットモジュールが、このビットを設定することを担う。このビット、またはData-to-Cardビットが常に設定されなければならないが、両方が設定されることは決してない。

30

## 【0110】

Data-to-Cardビットは、バッファの中のデータが、ファームウェアソケットモジュール（カード上に配備された）を宛先としているかどうかを示す。このビットが設定されている場合、データは、ドライバレベルソフトウェア204からファームウェアソケットモジュールに入力される。ドライバレベルソフトウェア204は、このビットを設定することを担う。このビット、またはData-to-Kernelビットが常に設定されなければならないが、両方が設定されることは決してない。

## 【0111】

レディビットは、ドライバレベルソフトウェア204によって、バッファが、ファームウェアソケットモジュールによって使用される準備ができていることを示すのに設定される。ファームウェアソケットモジュールは、そのバッファを処理する際、このビットをクリアしなければならない。

40

## 【0112】

完了ビットは、ファームウェアソケットモジュールによって、ファームウェアソケットモジュールが、そのバッファを処理することを完了したことをドライバレベルソフトウェア204に示すのに設定される。ドライバレベルソフトウェア204は、そのバッファをクリーンアップすることを終わると、このビットをクリアしなければならない。

## 【0113】

割り込みビットは、ドライバレベルソフトウェアによって、ファームウェアソケットモ

50

ジュールが、このバッファを処理した後、Data - to - Card 記述子バッファ完了割り込みを使用して割り込みを行うよう、ファームウェアソケットモジュールに命令するのに設定される。

【 0 1 1 4 】

バイトカウントフィールドは、バッファに関するバイトカウントを含む 6 4 ビットのフィールドである。このバイトカウントの意味は、入力記述子バッファに関するのと、出力記述子バッファに関するのとで異なる。Data - to - Card 入力記述子バッファの場合、このバイトカウントフィールドは、バッファの中に含まれるデータのバイト数である。ファームウェアソケットモジュールは、入ってくるバイトカウントに関して 2 0 ビットだけしか使用せず、このため、ファームウェアソケットモジュールが扱うことができる最大の Data - to - Card バッファは、( 1 M B - 1 ) バイトである。このバイトカウント値は、ドライバレベルソフトウェア 2 0 4 によって設定されなければならない。Data - to - Kernel 出力記述子バッファ、および Command - to - Kernel 出力記述子バッファの場合、このバイトカウントは、ドライバレベルソフトウェア 2 0 4 によって、バッファの中でどれだけのバイトが、書き込みのために利用可能であることを示すように最初に設定される。ファームウェアソケットモジュールが、バッファ上で完了ビットを設定する際、ファームウェアソケットモジュールは、バッファに書き込まれる実際のバイト数で、このバイトカウントの更新も行わなければならない。このバイトカウントも、最大で ( 1 M B - 1 ) バイトである。

【 0 1 1 5 】

物理的アドレスフィールドは、メインメモリの中のカーネルバッファの物理的アドレスを含む 6 4 ビットのフィールドである。このフィールドは、ドライバレベルソフトウェア 2 0 4 によって設定される。好ましくは、ファームウェアソケットモジュールは、4 8 ビットで表されることが可能な物理的アドレスだけを扱う。

【 0 1 1 6 】

仮想アドレスフィールドは、カーネルバッファの仮想アドレスを含む 6 4 ビットのフィールドであり、ドライバレベルソフトウェア 2 0 4 だけによって使用される。

【 0 1 1 7 】

動作の際、ファームウェアソケットモジュールは、ファームウェアソケットモジュールのドアベルレジスタ上で実行信号を受け取ると、バッファ 5 0 0 の中の Data - to - Card 入力記述子を読み取り、準備ができていないバッファから読み取ろうと試みるまで、またはファームウェアソケットモジュールによって保持される出力 F I F O の中にコマンド情報またはデータが存在するまで、読み取り続ける。

【 0 1 1 8 】

ファームウェアソケットモジュールは、好ましくは、出力のためのコマンド情報に最高の優先順位を与える。出力のためのコマンド情報が存在する場合、ファームウェアソケットモジュールは、その情報を、利用可能な Command - to - Kernel 出力記述子バッファ 5 0 6 に出力する。そのようなバッファ 5 0 6 が、利用できない場合、ファームウェアソケットモジュールは、Command - to - Kernel 出力記述子バッファ満杯割り込みを発行する。すると、ファームウェアソケットは、ドアベルレジスタの中の実行ビットに、ドライバレベルソフトウェア 2 0 4 によって、バッファ 5 0 6 が利用可能にされたことを示すように書き込みが行われるまで、バッファ 5 0 6 にコマンド情報を再び出力しようとししない。

【 0 1 1 9 】

ファームウェアソケットモジュールは、好ましくは、出力のためのデータに、2 番目に高い優先順位を与える。ファームウェアソケットモジュールの出力 F I F O の中のデータの量が、最大 P C I - X パーストサイズ以上である場合、ファームウェアソケットモジュールは、利用可能な Data - to - Kernel 出力記述子バッファ 5 0 4 にデータを出力する。次の Data - to - Kernel バッファ 5 0 4 が、利用できない場合、ファームウェアソケットモジュールは、Data - to - Kernel 出力記述子バッファ

満杯割り込みを通知する。すると、ファームウェアソケットモジュールは、ドアベルレジスタの中の実行ビットに、ドライバレベルソフトウェア204によって、バッファ504が利用可能にされたことを示すように書き込みが行われるまで、バッファ504にデータを再び出力しようとしな

【0120】

ファームウェアソケットモジュールは、好ましくは、入ってくるデータまたはコマンドに最低の優先順位を与える。次のData-to-Card入力記述子バッファ500が、利用できない場合、ファームウェアソケットモジュールは、Data-to-Card入力記述子バッファ空割り込みを通知する。ファームウェアソケットモジュールは、ドアベルレジスタの中の実行ビットに、ドライバレベルソフトウェア204によって書き込みが行われて、バッファ500が満たされたことを示すまで、次のバッファ500を読み取るうとしな

10

【0121】

また、ファームウェアソケットモジュールは、好ましくは、ネクストポインタが、カウントに到達するまで、バッファ500、504、および506を順に処理することを続け、到達した時点で、ネクストポインタは、第1のバッファにリセットされる。

【0122】

図6aは、インバウンドのデータおよびコマンドが、システム100によってどのように処理されることが可能であるかの例を示す。図6aの入力記述子ブールバッファ500は、複数の記述子を格納し、これらの記述子は、コマンドおよびターゲットデータが格納されたメモリ空間502内のアドレスに対するポインタを含む。この例において、ポインタ1は、データストリームの中で「Smith」という語に関する検索を実行するよう、FAMチェーン130におけるFAM#3を構成するコマンドをポイントする。他のコマンドに対するポインタは、図6aにおいて特定される。また、バッファ500は、システム100を通して処理されるべきターゲットデータをポイントする記述子も格納する（例えば、ポインタ4参照）。前述した通り、ドライバレベルソフトウェア204は、これらの記述子でバッファ500をポピュレートして、ファームウェアソケットモジュールに送られ、FAMチェーンを通して伝搬されるべきコマンドおよびデータの順序を定義する。

20

【0123】

図6b - 図6oは、図6aの例によって定義される順序を使用する、そのような伝搬を示す。図示を簡単にするため、SDコマンドおよびEDコマンドは、バッファ500の中のターゲットデータの境界において省略されている。見て取ることができる通り、コマンドとターゲットデータの間の定義された順序は、これらのコマンドおよびデータが、FPGAを通して処理される中で、ファームウェアソケットモジュールおよびFAMチェーンによって保たれる。入力記述子バッファの中でコマンドおよびターゲットデータを適切に順序付けることにより、ソフトウェアは、強力な高速データ処理動作を実現することができる。例えば、ポインタ1 - 5に対応するバッファ500の中の記述子は、アドレスA + 1からCまでメモリの中に格納された暗号化されたターゲットデータに対して解読動作を実行し、その後、その解読されたデータの中で、「Smith」という語の存在の検索を実行するよう効果がある。また、アドレスP + 1からWまでメモリの中に格納されたデータが、暗号化された後に圧縮されているものと想定すると、ポインタ6以下に対応するバッファ500の中の記述子は、そのターゲットデータに対して伸張動作を実行し、その後、その伸張されたターゲットデータに対して解読動作を実行し、その後、解読され、伸張されたターゲットデータの中で、「Jones」という語の検索を実行するよう効果がある。また、実際には、コマンドセットに従って処理されるターゲットデータの長さは、図6a - 図6oに示される例よりも相当に大きい可能性が高く、例えば、ターゲットデータの長さは、所望されるデータ処理動作の範囲に応じて、メガバイト単位、ギガバイト単位、さらにテラバイト単位（またはそれより大きい単位）のオーダーでさえある可能性があることも注目に値する。

30

40

【0124】

50

## ソフトウェアライブラリインタフェース 2 1 0

F A Mのアプリケーション使用をサポートするソフトウェア A P Iは、好ましくは、E x e g y F p g aというラベルが付けられることが可能なクラスなどの、クラスで実現される。このクラスは、好ましくは、以下の公開メソッドを含む。

## 【数 1】

```
bool ExegyFpga::SendCommand ( const char * TwoChars,
                               const int ModuleID,
                               const unsigned int Parameter,
                               const bool WaitForIt,
                               const bool Sync )
```

10

## 【0 1 2 5】

S e n d C o m m a n dメソッドは、コマンドチャネルを介してファームウェアソケットモジュールおよびF A Mに所与のコマンドを送る。引数「T w o C h a r s」は、コマンドフィールド（16ビット）に入るバイトのペア（通常、A S C I I文字であるが、そうである必要はない）である。誤りフィールドが、クリアされる（0に設定される）。引数「S y n c」は、同期フィールド（1ビット）に入る。引数「M o d u l e I d」は、モジュールIDフィールド（6ビット）に入る。引数「P a r a m e t e r」は、パラメータフィールド（32ビット）に入る。引数「W a i t F o r I t」は、コマンドがファームウェアから戻るまで、メソッド呼び出しがブロックすべき（W a i t F o r I tが、1である場合）か、またはブロックすべきでないか（W a i t F o r I tが、0である場合）を示すフラグである。メソッドは、誤りが生じると、間違いを戻す。

20

## 【数 2】

```
size_t ExegyFpga::Write ( size_t bytes, char * buffer )
```

## 【0 1 2 6】

W r i t eメソッドは、所与のバッファからデータチャネルにデータを送る。メソッドは、転送されたバイト数、または「e r r n o」の中で誤りを戻す。

## 【数 3】

```
size_t ExegyFpga::ReadCmd ( size_t bytes, char * buffer )
```

30

## 【0 1 2 7】

R e a d C m dメソッドは、C o m m a n d - t o - K e r n e l出力記述子バッファからコマンドを読み取り、これらのコマンドをユーザバッファに送る。このメソッドは、バッファの中に入れられたバイトを戻す。

## 【数 4】

```
size_t ExegyFpga::Read( size_t bytes, char * buffer )
```

## 【0 1 2 8】

R e a dメソッドは、D a t a - t o - K e r n e l出力記述子バッファからデータを読み取り、これらのデータをユーザバッファに送る。このメソッドは、バッファに入れられたバイト、または「e r r n o」を戻す。同期マーク（すなわち、コマンド同期ビットの結果であるN u l lバッファ）に遭遇すると、E N O D A T Aが、戻される。

40

## 【0 1 2 9】

ソフトウェアライブラリインタフェースの例示的な用法が、以下に示される

【数 5】

```

Card = ExegyFpga::Open(...);

Card->SendCommand(...);           //send a command to FAM chain
Card->Write(...);                  //send test data
Card->ReadCmd(...);                //ensure command made it
Card->Read(...);                   //read test data
Card->Close();

```

10

【0130】

本発明を、本発明の好ましい実施形態に関連して以上に説明してきたが、本発明の範囲にやはり含まれる様々な変形が、好ましい実施形態に行われることが可能である。本発明のそのような変形は、本明細書の教示を検討することで、認識可能となろう。このため、本発明の完全な範囲は、特許請求の範囲、および特許請求の範囲の法的均等物によってだけ規定されるべきである。

【図面の簡単な説明】

【0131】

【図1】本発明を実施する好ましいシステムを示すブロック図である。

【図2】好ましいシステムのためのソフトウェアおよびファームウェアの配備に関する例示的なフレームワークを示す図である。

20

【図3a】本発明によるデータ処理タスクを実行するようにコンピュータシステムに実装するための好ましいプリント回路基板を示すブロック図である。

【図3b】本発明によるデータ処理タスクを実行するようにコンピュータシステムに実装するための代替のプリント回路基板を示すブロック図である。

【図4】ファームウェアソケットモジュールとファームウェアアプリケーションモジュールとの間の好ましい信号層インタフェースを示す図である。

【図5】ファームウェアソケットモジュールと、デバイスドライバによって管理されるメモリ空間との間の好ましいハードウェア/ソフトウェアインタフェースを示す図である。

【図6a】デバイスドライバによって管理される入力記述子プールバッファ - およびカーネルメモリ空間の実施例を示す図である。

30

【図6b】図6aの入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

【図6c】図6aの入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

【図6d】図6aの入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

40

【図6e】図6aの入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

【図6f】図6aの入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通さ

50

れるかを示す図である。

【図 6 g】図 6 a の入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

【図 6 h】図 6 a の入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

【図 6 i】図 6 a の入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

10

【図 6 j】図 6 a の入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

【図 6 k】図 6 a の入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

20

【図 6 l】図 6 a の入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

【図 6 m】図 6 a の入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

【図 6 n】図 6 a の入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

30

【図 6 o】図 6 a の入力記述子プールバッファ実施例によって規定されるコマンドとターゲットデータの順序に従って、コマンドおよびターゲットデータが、ファームウェアソケットモジュールおよびファームウェアアプリケーションモジュールをどのように順に通されるかを示す図である。

【図 7】ファームウェアソケットモジュールと、異なるデータ処理動作を実行するようにそれぞれが構成された、複数のパイプライン化されたファームウェアアプリケーションモジュールとを有して構成された例示的な F P G A を示す図である。

40

【図 8】パイプラインのファームウェアアプリケーションモジュールが、どのように複数の F P G A にまたがって配備されることが可能であるかの例を示す図である。



【図 1】

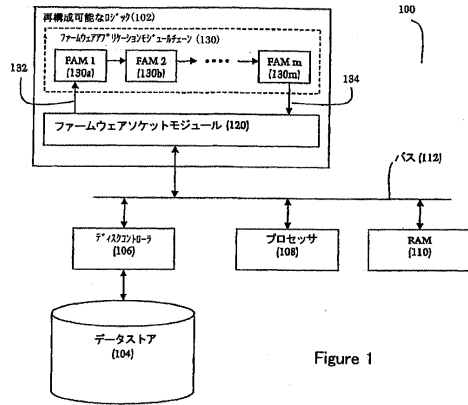


Figure 1

【図 2】

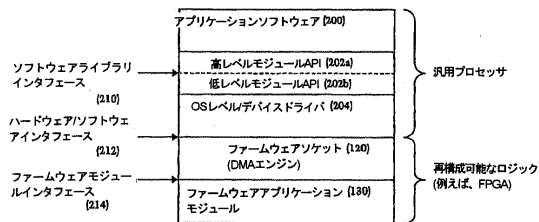


Figure 2

【図 3 a】

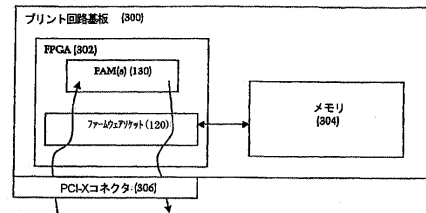


Figure 3a

【図 3 b】

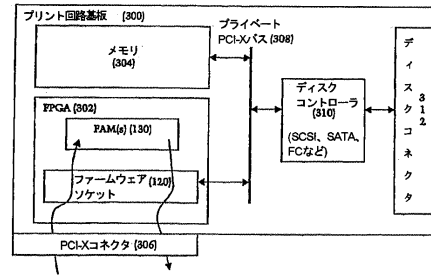


Figure 3b

【図 4】

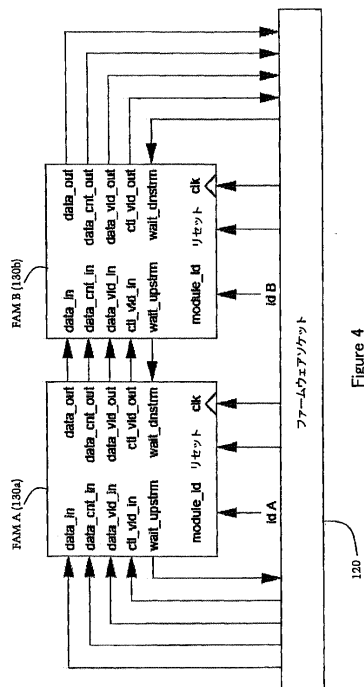


Figure 4

【図 5】

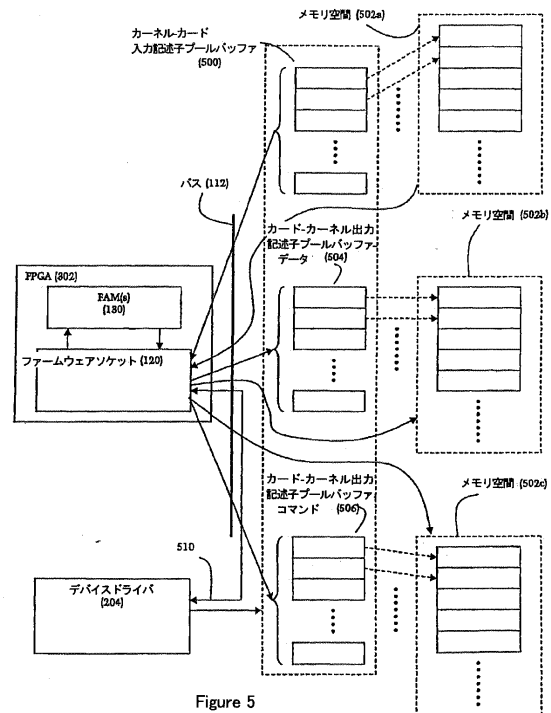


Figure 5

【図 6 a】

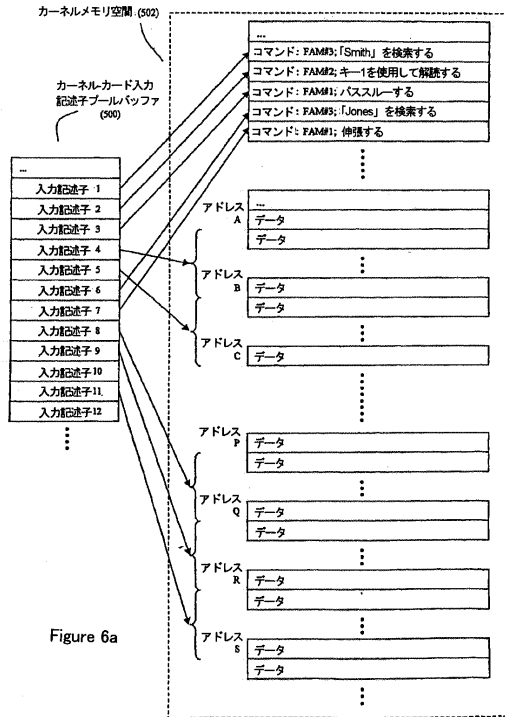


Figure 6a

【図 6 b】

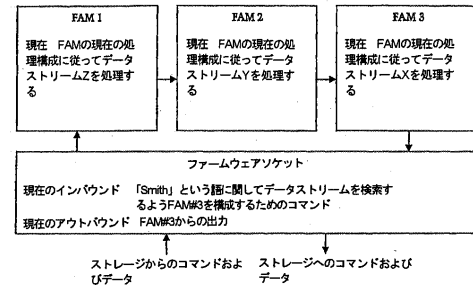


Figure 6b

【図 6 c】

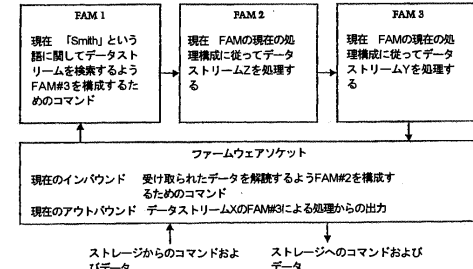


Figure 6c

【図 6 d】

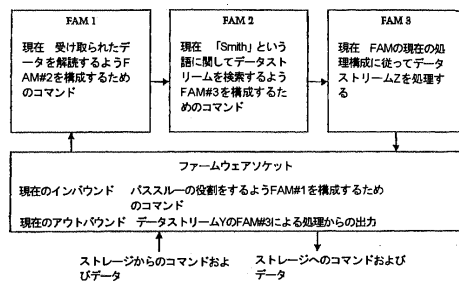


Figure 6d

【図 6 f】

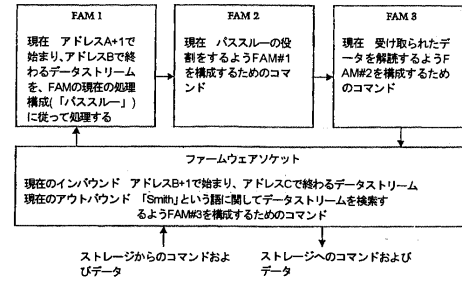


Figure 6f

【図 6 e】

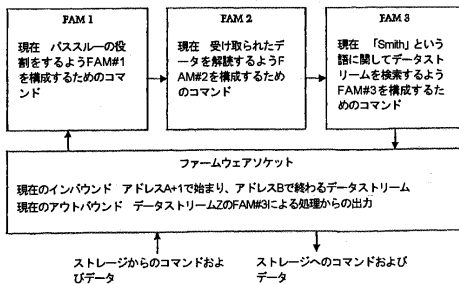


Figure 6e

【図 6 g】

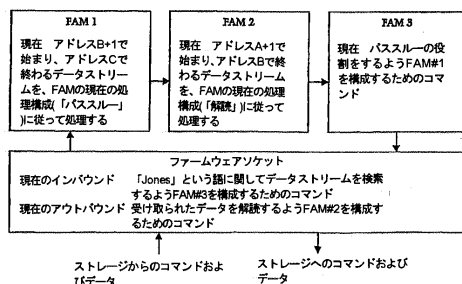


Figure 6g

【 図 6 i 】

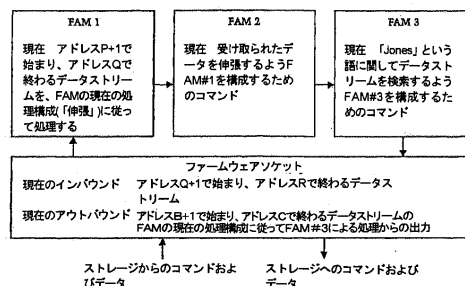


Figure 6j

【 ㊦ 6 k 】

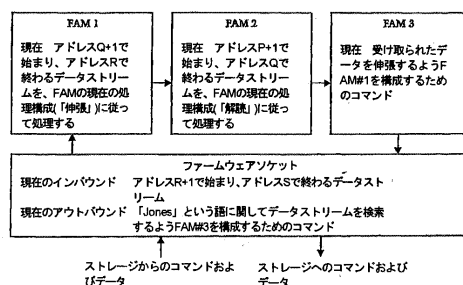


Figure 6k

【 図 6 n 】

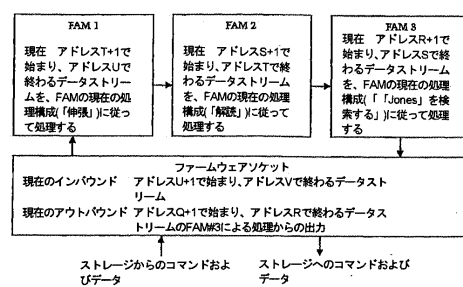


Figure 6n

【 図 6 0 】

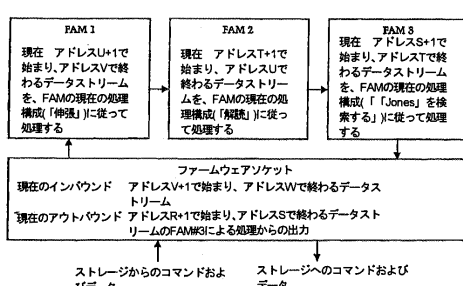


Figure 6o

【図 7】

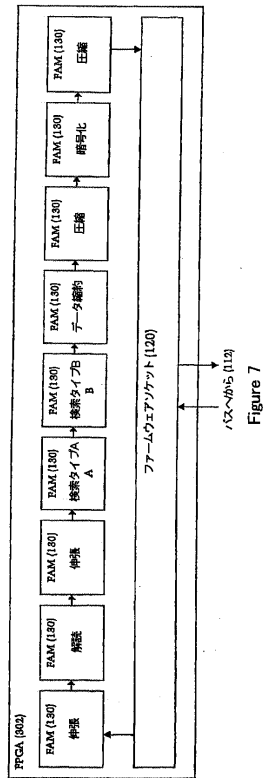


Figure 7

【図 8】

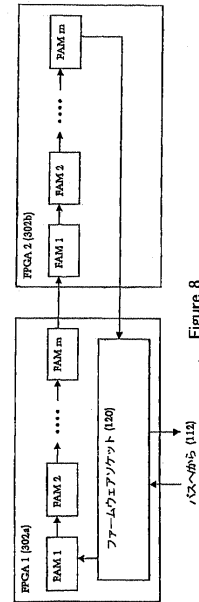


Figure 8

## フロントページの続き

- (74)代理人 100140523  
弁理士 渡邊 千尋
- (74)代理人 100119253  
弁理士 金山 賢教
- (74)代理人 100103920  
弁理士 大崎 勝真
- (74)代理人 100124855  
弁理士 坪倉 道明
- (72)発明者 チャンパーライン, ロジヤー・デイ  
アメリカ合衆国、ミズーリ・６３１４１、セント・ルイス、ノートル・ダム・ドライブ・６４
- (72)発明者 シヤンズ, イー・エフ・バークレ  
アメリカ合衆国、ミズーリ・６３１２２、カークウッド、メドウリッジ・レーン・６０３
- (72)発明者 ブロディ, ベンジャミン・シー  
アメリカ合衆国、ミズーリ・６３１３０、ユニバーシティ・シティ、リーランド・アベニュー・１  
・ダブリュ・７２６
- (72)発明者 ヘンリクス, マイケル  
アメリカ合衆国、ミズーリ・６３１３９、セント・ルイス、サウスウエスト・アベニュー・５７０  
８
- (72)発明者 ホワイト, ジエイソン・アール  
アメリカ合衆国、ミズーリ・６３０２１、マンチエスター、ドッグウッド・クリーク・ドライブ・  
９０２、アパートメント・１

審査官 中野 裕二

- (56)参考文献 国際公開第２００４／０４２５６０(WO, A1)  
国際公開第２００４／０４２５６１(WO, A1)  
国際公開第２００４／０４２５７４(WO, A1)  
国際公開第２００４／０４２５６２(WO, A1)  
国際公開第２００４／０４２５６９(WO, A1)  
特表２００６－５１８０５７(JP, A)  
特表２００６－５１５９４１(JP, A)  
特表２００６－５１８４９５(JP, A)  
特表２００６－５１８０５８(JP, A)  
特表２００６－５１８０５６(JP, A)  
特表２００３－５３２２２４(JP, A)  
特開平０５－０９４４０６(JP, A)  
米国特許第０５４２１０２８(US, A)  
米国特許出願公開第２００３／００９７４８１(US, A1)

- (58)調査した分野(Int.Cl., DB名)  
G 0 6 F 9 / 3 8  
G 0 6 F 1 5 / 8 0