

(19) 日本国特許庁(JP)

## (12) 特 許 公 報(B2)

(11) 特許番号

特許第4127071号  
(P4127071)

(45) 発行日 平成20年7月30日(2008.7.30)

(24) 登録日 平成20年5月23日(2008.5.23)

(51) Int.Cl.

G06F 13/38 (2006.01)

F 1

G06F 13/38 340C

請求項の数 11 (全 35 頁)

(21) 出願番号 特願2003-41562 (P2003-41562)  
 (22) 出願日 平成15年2月19日 (2003.2.19)  
 (62) 分割の表示 特願2002-118206 (P2002-118206)  
     分割  
     原出願日 平成14年4月19日 (2002.4.19)  
 (65) 公開番号 特開2003-316735 (P2003-316735A)  
 (43) 公開日 平成15年11月7日 (2003.11.7)  
 審査請求日 平成17年4月18日 (2005.4.18)

特許法第30条第1項適用 Design Wave  
 2002年4月号 (2002年4月1日) CQ出版株式  
 会社発行第82-88ページに発表

(73) 特許権者 000002369  
     セイコーエプソン株式会社  
     東京都新宿区西新宿2丁目4番1号  
 (74) 代理人 100090479  
     弁理士 井上 一  
 (74) 代理人 100090387  
     弁理士 布施 行夫  
 (74) 代理人 100090398  
     弁理士 大渕 美千栄  
 (72) 発明者 ▲斎▼藤 伸之  
     長野県諏訪市大和3丁目3番5号 セイコ  
     エプソン株式会社内  
 (72) 発明者 久保田 慎介  
     長野県諏訪市大和3丁目3番5号 セイコ  
     エプソン株式会社内

最終頁に続く

(54) 【発明の名称】データ転送制御装置、電子機器及びデータ転送制御方法

## (57) 【特許請求の範囲】

## 【請求項1】

U S B (Universal Serial Bus) に接続される電子機器に組み込まれ、U S B を介してデータを送受信するデータ転送制御装置であって、

エンドポイントに対応してパイプ領域をパケットバッファに確保し、パケットバッファに対してアクセス制御を行うバッファコントローラと、

パイプ領域とエンドポイントとの間でのデータ転送の転送条件情報が設定される転送条件レジスタを含むレジスタ部と、

転送条件レジスタに設定された転送条件情報に基づいて、エンドポイントに対するトランザクションを発生し、パイプ領域と、そのパイプ領域に対応するエンドポイントとの間で、データを転送する転送コントローラとを含み、

前記バッファコントローラは、

所与のデータ単位の転送が終了するとパイプ領域を解放し、

前記パイプ領域が解放された前記パケットバッファに、エンドポイントに対応してパイプ領域を再割り当てすることを特徴とするデータ転送制御装置。

## 【請求項2】

請求項1において、

前記転送条件レジスタに設定される転送条件情報が、エンドポイントとの間で転送されるデータのトータルサイズと、マックスパケットサイズと、データの転送方向とを含み、

前記転送コントローラは、

10

20

前記トータルサイズのデータを、前記マックスパケットサイズのペイロードのパケットを用いて、前記転送方向で指定される方向で、パイプ領域とそのパイプ領域に対応するエンドポイントとの間で転送することを特徴とするデータ転送制御装置。

**【請求項 3】**

請求項 1 又は 2 において、  
前記転送条件レジスタに設定される転送条件情報が、データ転送の転送種別を含み、  
前記転送コントローラは、  
パイプ領域のデータを、転送条件レジスタに設定される転送種別のデータ転送により、  
転送することを特徴とするデータ転送制御装置。

**【請求項 4】**

10

請求項 1 乃至 3 のいずれかにおいて、  
前記転送条件レジスタに設定される転送条件情報が、複数のパイプ領域間の転送比率情報を含み、  
前記転送コントローラは、  
前記転送比率情報で設定される転送比率で、パイプ領域のデータを転送することを特徴とするデータ転送制御装置。

**【請求項 5】**

20

請求項 4 において、  
前記転送比率情報が、  
パイプ領域のトランザクションの連続実行回数であることを特徴とするデータ転送制御装置。

**【請求項 6】**

請求項 1 乃至 5 のいずれかにおいて、  
前記転送条件レジスタに設定される転送条件情報が、インタラプト転送におけるトークン発行周期を含み、  
前記転送コントローラは、  
転送条件レジスタに設定される前記トークン発行周期で、インタラプト転送のトークンパケットを転送することを特徴とするデータ転送制御装置。

**【請求項 7】**

30

請求項 1 乃至 6 のいずれかにおいて、  
前記パイプ領域が、  
コントロール転送のエンドポイントに専用のパイプ領域と、任意のエンドポイントに割り当て可能な汎用のパイプ領域を含むことを特徴とするデータ転送制御装置。

**【請求項 8】**

請求項 1 乃至 7 のいずれかにおいて、  
前記バスとは異なる他のバスとパケットバッファとの間でのデータ転送を行うインターフェース回路を含み、

前記インターフェース回路、前記転送コントローラに対して、処理部がデータ転送の開始を指示した場合に、前記インターフェース回路が他のバスを介したデータ転送を行うと共に、前記転送コントローラがバスを介したデータ転送を行い、データ転送が終了した場合に、前記転送コントローラが、処理部に対して割り込みを発生することを特徴とするデータ転送制御装置。

40

**【請求項 9】**

請求項 1 乃至 8 のいずれかにおいて、  
ホストの役割として動作するホスト動作のステートと、ペリフェラルの役割として動作するペリフェラル動作のステートを含む複数のステートの制御を行うステートコントローラを含み、  
前記転送コントローラは、  
ホスト動作時において、ホストとしてのデータ転送を行うホストコントローラと、  
ペリフェラル動作時において、ペリフェラルとしてのデータ転送を行うペリフェラルコ

50

ントローラとを含み、

ホスト動作時において、パケットバッファに対してパイプ領域が確保され、前記ホストコントローラが、確保されたパイプ領域とそのパイプ領域に対応するエンドポイントとの間で、データを転送することを特徴とするデータ転送制御装置。

【請求項 10】

請求項 1 乃至 9 のいずれかのデータ転送制御装置と、

前記データ転送制御装置及びバスを介して転送されるデータの出力処理又は取り込み処理又は記憶処理を行う装置と、

前記データ転送制御装置のデータ転送を制御する処理部と、

を含むことを特徴とする電子機器。 10

【請求項 11】

USB (Universal Serial Bus) を介してデータを送受信するためのデータ転送制御方法であって、

エンドポイントに対応してパイプ領域をパケットバッファに確保し、

パイプ領域とエンドポイントとの間でのデータ転送の転送条件情報を、転送条件レジスタに設定し、

転送条件レジスタに設定された転送条件情報に基づいて、エンドポイントに対するトランザクションを発生し、パイプ領域と、そのパイプ領域に対応するエンドポイントとの間で、データを転送すると共に、

所与のデータ単位の転送が終了するとパイプ領域を解放し、 20

前記パイプ領域が解放された前記パケットバッファに、エンドポイントに対応してパイプ領域を再割り当てすることを特徴とするデータ転送制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、データ転送制御装置、電子機器及びデータ転送制御方法に関する。

【0002】

【背景技術及び発明が解決しようとする課題】

HS (High Speed) モードをサポートするUSB (Universal Serial Bus) 2.0 の市場が順調に拡大しているなか、USB Implementers Forum (USB - IF) によって、USB On - The - Go (OTG) と呼ばれるインターフェース規格が策定された。USB 2.0 を拡張する形で策定されたOTG規格 (OTG1.0) は、USBインターフェースの新たな付加価値を生む可能性を秘めており、その特質を生かしたアプリケーションの登場が待たれている。 30

【0003】

このOTGによれば、これまでUSBを介してホスト (パーソナルコンピュータ等) に接続されていたペリフェラル (周辺機器) に、ホスト機能を持たせることができる。これにより、ペリフェラル同士をUSBで接続してデータを転送することができるようになり、例えばデジタルカメラとプリンタとをダイレクトに接続して、デジタルカメラの画像を印刷することが可能になる。また、デジタルカメラやデジタルビデオカメラをストレージ装置に接続して、データを保存することが可能になる。 40

【0004】

しかしながら、OTGによりホスト機能を持たせるペリフェラルには、低性能のCPU (処理部) が組み込まれているのが一般的である。従って、ホスト機能の追加により、ペリフェラルが有するCPU (ファームウェア) の処理負荷が重くなったり、処理が複雑化すると、他の処理に支障が生じたり、機器の設計期間が長期化するなどの問題が生じる。

【0005】

本発明は、以上のような技術的課題に鑑みてなされたものであり、その目的とするところは、処理部の処理負荷の軽減等を図ることができるデータ転送制御装置、電子機器及びデータ転送制御方法を提供することにある。 50

**【0006】****【課題を解決するための手段】**

本発明は、バスを介したデータ転送のためのデータ転送制御装置であって、各エンドポイントに対応して各パイプ領域が確保され、各エンドポイントとの間で転送されるデータが各パイプ領域に記憶される複数のパイプ領域を有するパケットバッファに対して、アクセス制御を行うバッファコントローラと、各パイプ領域と各エンドポイントとの間でのデータ転送の転送条件情報が各転送条件レジスタに設定される複数の転送条件レジスタを含むレジスタ部と、転送条件レジスタに設定された転送条件情報に基づいて、エンドポイントに対するトランザクションを自動発生し、パイプ領域と、そのパイプ領域に対応するエンドポイントとの間で、データを自動転送する転送コントローラとを含むデータ転送制御装置に関係する。

10

**【0007】**

本発明では、パケットバッファに対して複数のパイプ領域（バッファ領域）が確保（allocate）される。この場合に、各パイプ領域は、バスの各エンドポイントに対応して確保される。また、各パイプ領域には、それに対応する各エンドポイントとの間で転送（送信、受信）されるデータが記憶（バッファリング）される。バッファコントローラは、このようなパイプ領域が確保されたパケットバッファ（バッファ）のアクセス制御（領域管理）を行う。

**【0008】**

そして本発明では、各パイプ領域と各エンドポイントとの間でのデータ転送の転送条件情報（エンドポイント情報、パイプ情報）が各転送条件レジスタ（パイプレジスタ）に設定される。そして、転送コントローラは、各転送条件レジスタに設定された転送条件情報に基づいて、各エンドポイントに対するトランザクションを自動発生し、各パイプ領域と各エンドポイントとの間でデータを自動転送する。これにより、データ転送制御装置の制御等を行う処理負荷の軽減等を図れる。

20

**【0009】**

また本発明では、前記転送条件レジスタに設定される転送条件情報が、エンドポイントとの間で転送されるデータのトータルサイズと、マックスパケットサイズと、データの転送方向とを含み、前記転送コントローラが、前記トータルサイズのデータを、前記マックスパケットサイズのペイロードのパケットを用いて、前記転送方向で指定される方向で、パイプ領域とそのパイプ領域に対応するエンドポイントとの間で自動転送してもよい。

30

**【0010】**

このようにすれば、転送条件レジスタに、トータルサイズ、マックスパケットサイズ、転送方向などの転送条件情報を設定するだけで、データを自動転送できるようになり、処理部の処理負荷の軽減等を図れる。

**【0011】**

また本発明では、前記転送条件レジスタに設定される転送条件情報が、データ転送の転送種別を含み、前記転送コントローラが、各パイプ領域のデータを、各転送条件レジスタに設定される転送種別のデータ転送により、自動転送してもよい。

40

**【0012】**

このようにすれば、各パイプ領域のデータを、任意の転送種別のデータ転送（例えば、アイソクロナス、インタラプトなどの周期転送、バルク、コントロールなどの非周期転送）で転送できるようになる。

**【0013】**

また本発明では、前記転送条件レジスタに設定される転送条件情報が、複数のパイプ領域間の転送比率情報を含み、前記転送コントローラが、前記転送比率情報で設定される転送比率で、各パイプ領域のデータを自動転送してもよい。

**【0014】**

このようにすれば、複数のパイプ領域のデータを、任意の転送比率で転送できるようになり、データ転送の効率的なスケジューリングが可能になる。

50

**【 0 0 1 5 】**

また本発明では、前記転送比率情報が、各パイプ領域のトランザクションの連続実行回数であってもよい。

**【 0 0 1 6 】**

このようにすれば、例えば、第 K のパイプ領域のデータ転送のトランザクションを複数回連続して実行（発生）した後に、第 K + 1 のパイプ領域のデータ転送のトランザクションを実行できるようになる。

**【 0 0 1 7 】**

また本発明では、前記転送条件レジスタに設定される転送条件情報が、インタラプト転送におけるトークン発行周期を含み、前記転送コントローラが、転送条件レジスタに設定される前記トークン発行周期で、インタラプト転送のトークンパケットを自動転送してもよい。10

**【 0 0 1 8 】**

このようにすれば、2分木構造のディスクリプタ等を用いることなく、インタラプト転送のトークンパケットを発行できるようになる。

**【 0 0 1 9 】**

また本発明では、前記パイプ領域が、コントロール転送のエンドポイントに専用のパイプ領域と、任意のエンドポイントに割り当て可能な汎用のパイプ領域を含んでもよい。

**【 0 0 2 0 】**

このようにすれば、例えば、パイプ領域とエンドポイントとの対応づけを動的に変化させることなどが可能になり、データ転送の効率的なスケジューリングが可能になる。20

**【 0 0 2 1 】**

また本発明では、前記バスとは異なる他のバスとパケットバッファとの間でのデータ転送を行うインターフェース回路を含み、前記インターフェース回路、前記転送コントローラに対して、処理部がデータ転送の開始を指示した場合に、前記インターフェース回路が他のバスを介したデータ転送を行うと共に、前記転送コントローラがバスを介したデータ転送を行い、データ転送が終了した場合に、前記転送コントローラが、処理部に対して割り込みを発生してもよい。

**【 0 0 2 2 】**

このようにすれば、処理部は、転送条件を設定し、バス及び他のバス（処理部のバス又はシステムメモリのバス等）を介したデータ転送の開始を指示した後は、割り込みが発生するまで、データ転送制御装置の制御に関わらなくて済むようになる。これにより、処理部の処理負荷の軽減等を図れる。30

**【 0 0 2 3 】**

また本発明では、ホストの役割として動作するホスト動作のステートと、ペリフェラルの役割として動作するペリフェラル動作のステートを含む複数のステートの制御を行うステートコントローラを含み、前記転送コントローラが、ホスト動作時において、ホストとしてのデータ転送を行うホストコントローラと、ペリフェラル動作時において、ペリフェラルとしてのデータ転送を行うペリフェラルコントローラとを含み、ホスト動作時において、パケットバッファに対して複数のパイプ領域が確保され、前記ホストコントローラが、確保されたパイプ領域とそのパイプ領域に対応するエンドポイントとの間で、データを自動転送してもよい。40

**【 0 0 2 4 】**

本発明によれば、例えば、ステートコントローラにより制御されるステートが、ホスト動作のステートになると、ホストコントローラにより、ホストの役割としてのデータ転送が行われる。また、ステートコントローラにより制御されるステートが、ペリフェラル動作のステートになると、ペリフェラルコントローラにより、ペリフェラルの役割としてのデータ転送が行われる。これにより、いわゆるデュアルロール・デバイス機能を実現できるようになる。

**【 0 0 2 5 】**

そして本発明では、ホスト動作時において、パケットバッファに対して複数のパイプ領域が確保され、確保されたパイプ領域とエンドポイントとの間でデータが自動転送される。これにより、デュアルロール・デバイス機能を実現できると共に、ホスト動作時における処理部の処理負荷の軽減等を図れる。

【0026】

また本発明では、USB (Universal Serial Bus)のOTG (On-The-Go)規格に準拠したデータ転送を行ってもよい。

【0027】

また本発明は、上記のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して転送されるデータの出力処理又は取り込み処理又は記憶処理を行う装置と、前記データ転送制御装置のデータ転送を制御する処理部とを含むことを特徴とする電子機器に関係する。

10

【0028】

また本発明は、バスを介したデータ転送のためのデータ転送制御方法であって、各エンドポイントとの間で転送されるデータが各パイプ領域に記憶される複数のパイプ領域を、パケットバッファに確保し、各パイプ領域と各エンドポイントとの間でのデータ転送の転送条件情報を、複数の転送条件レジスタの各転送条件レジスタに設定し、転送条件レジスタに設定された転送条件情報に基づいて、エンドポイントに対するトランザクションを自動発生し、パイプ領域と、そのパイプ領域に対応するエンドポイントとの間で、データを自動転送するデータ転送制御方法に関係する。

20

【0029】

【発明の実施の形態】

以下、本実施形態について説明する。

【0030】

なお、以下に説明する本実施形態は、特許請求の範囲に記載された本発明の内容を不当に限定するものではない。また本実施形態で説明される構成の全てが本発明の解決手段として必須であるとは限らない。

【0031】

1. OTG

まず、OTG (USB On-The-Go)について簡単に説明する。

30

【0032】

1.1 Aデバイス、Bデバイス

OTGでは、コネクタの規格として、図1(A)に示すようなMini-A プラグ、Mini-B プラグが定義されている。また、これらのMini-A プラグ、Mini-B プラグ(広義にはケーブルの第1、第2のプラグ)の両方を接続できるコネクタとして、Mini-AB レセプタクル(receptacle)が定義されている。

【0033】

そして例えば図1(B)のように、USBケーブルのMini-A プラグに電子機器Pが接続され、Mini-B プラグに電子機器Qが接続されると、電子機器PはAデバイスに設定され、電子機器QはBデバイスに設定される。一方、図1(C)に示すように、Mini-B プラグ、Mini-A プラグが電子機器P、Qに接続されると、電子機器P、Qは、各自、Bデバイス、Aデバイスに設定される。

40

【0034】

Mini-A プラグ内ではIDピンがGNDに接続されており、Mini-B プラグ内ではIDピンはフローティング状態になっている。電子機器は、内蔵するプルアップ抵抗回路を用いて、このIDピンの電圧レベルを検出することで、自身がMini-A プラグに接続されたのか、或いはMini-B プラグに接続されたのかを判断する。

【0035】

OTGでは、Aデバイス(マスター)が、電源(VBUS)を供給する側(供給元)になり、Bデバイス(スレイブ)が、電源の供給を受ける側(供給先)になる。また、Aデバ

50

イスは、デフォルトのステートがホストになり、Bデバイスは、デフォルトのステートがペリフェラル（周辺機器）になる。

【0036】

1.2 デュアルロール・デバイス

OTGでは、ホスト（簡易ホスト）としての役割とペリフェラルとしての役割の両方を持つことができるデュアルロール・デバイス（Dual-Role Device）が定義されている。

【0037】

デュアルロール・デバイスは、ホストにもペリフェラルにもなれる。そして、デュアルロール・デバイスに接続された相手が、従来のUSB規格におけるホストやペリフェラルである場合には、デュアルロール・デバイスの役割は一意に定まる。つまり、接続相手がホストであれば、デュアルロール・デバイスはペリフェラルになり、接続相手がペリフェラルであれば、デュアルロール・デバイスはホストになる。

【0038】

一方、接続相手がデュアルロール・デバイスである場合には、両方のデュアルロール・デバイスは、お互いにホストとペリフェラルの役割を交換できる。

【0039】

1.3 S R P、H N P

デュアルロール・デバイスは、図2(A)、(B)に示すようなセッション開始要求手順S R P (Session Request Protocol) やホスト交換手順H N P (Host Negotiation Protocol) の機能を持つ。

10

20

【0040】

ここでセッション開始要求手順S R Pは、BデバイスがAデバイスに対して、V B U S（電源）の供給を要求するプロトコルである。

【0041】

バスを使用しない場合にOTGでは、AデバイスはV B U Sの供給を停止できる。これにより、Aデバイスが例えば小型の携帯機器であった場合に、無駄な電力消費を防止できる。そして、AデバイスがV B U Sの供給を停止した後に、BデバイスがV B U Sを供給してもらいたい場合には、このS R Pを使用して、Aデバイスに対してV B U Sの供給の再開を要求する。

30

【0042】

図2(A)にS R Pの流れを示す。図2(A)に示すように、Bデバイスは、データ・ライン・パルシングとV B U Sパルシングを行うことで、Aデバイスに対してV B U Sの供給を要求する。そして、AデバイスによるV B U Sの供給の開始後に、Bデバイスのペリフェラル動作（peripheral operation）と、Aデバイスのホスト動作（host operation）が開始する。

【0043】

図1(A)～図1(C)で説明したように、デュアルロール・デバイス同士の接続では、Mini-Aプラグが接続された側であるAデバイスがデフォルトのホストとなり、Mini-Bプラグが接続された側であるBデバイスがデフォルトのペリフェラルになる。そして、OTGでは、プラグの抜き差しを行わなくても、ホストとペリフェラルの役割を交換できる。H N Pは、このホストとペリフェラルの役割を交換するためのプロトコルである。

40

【0044】

H N Pの流れを図2(B)に示す。デフォルトのホストとして動作するAデバイスが、バスの使用を終了すると、バスがアイドル状態になる。その後に、Bデバイスがデータ信号線D P (D+) のプルアップ抵抗を無効にすると、AデバイスがD P のプルアップ抵抗を有効にする。これにより、Aデバイスの役割はホストからペリフェラルに変わり、ペリフェラルとしての動作を開始する。また、Bデバイスの役割はペリフェラルからホストに変わり、ホストとしての動作を開始する。

【0045】

その後、Bデバイスが、バスの使用を終了し、AデバイスがD P のプルアップ抵抗を無効

50

にすると、Bデバイスが、DPのプルアップ抵抗を有効にする。これにより、Bデバイスの役割はホストからペリフェラルに戻り、ペリフェラルとしての動作を再開する。また、Aデバイスの役割はペリフェラルからホストに戻り、ホストとしての動作を再開する。

#### 【0046】

以上に説明したOTGによれば、携帯電話やデジタルカメラなどの携帯機器をUSBのホストとして動作させ、携帯機器同士をピア・ツー・ピアで接続してデータ転送を行うことが可能になる。これにより、USBインターフェースに新たな付加価値を生むことができ、これまでには存在しなかったアプリケーションを創出できる。

#### 【0047】

##### 2.OHCI

10

さて、従来のUSBでは、ホストであるパーソナルコンピュータが有するデータ転送制御装置（ホストコントローラ）は、マイクロソフト社が提唱したOHC（Open Host Controller Interface）や、UHC（Universal Host Controller Interface）といった規格に準拠していた。また、使用されるOS（Operating System）も、マイクロソフト社のWindowsやアップル社のマッキントッシュのOSなどに限定されていた。

#### 【0048】

しかしながら、OTGのターゲット・アプリケーションである小型携帯機器では、組み込まれるCPUのアーキテクチャや、使用されるOSは千差万別である。更に、パーソナルコンピュータのホストコントローラ向けに規格化されたOHCやUHCは、USBホストとしての機能をフルに実装することを前提としており、小型携帯機器への実装に最適であるとは言い難い。

20

#### 【0049】

例えば、図3（A）に、OHCで使用されるリスト構造のディスクリプタの一例を示す。

#### 【0050】

図3（A）において、エンドポイントディスクリプタED1、ED2、ED3は、リンクポインタによりリンクされており、エンドポイント1、2、3とのコミュニケーションに必要な情報が含まれている。そして、これらのED1、ED2、ED3には、転送ディスクリプタTD11～TD13、TD21、TD31～TD32がリンクポインタにより更にリンクされている。そして、これらの転送ディスクリプタには、エンドポイント1、2、3との間で転送するパケットデータに必要な情報が含まれている。

30

#### 【0051】

図3（A）のリスト構造のディスクリプタは、図3（B）のCPU610（広義には処理部）上で動作するファームウェア（ホストコントローラ・ドライバ）が作成し、システムメモリ620に書き込む。即ち、ファームウェアは、システム中のエンドポイントに対してエンドポイントディスクリプタを割り当て、エンドポイント情報等に基づいてエンドポイントディスクリプタ、転送ディスクリプタを作成する。そして、これらのディスクリプタをリンクポインタでリンクさせて、システムメモリ620に書き込む。

#### 【0052】

データ転送制御装置600（ホストコントローラ）は、システムメモリ620に書き込まれたリスト構造のディスクリプタを読み出し、エンドポイントディスクリプタや転送ディスクリプタに記述される情報に基づいて、データ転送を実行する。

40

#### 【0053】

具体的には、データ転送制御装置600（ホストコントローラ）は、ED1に基づいてエンドポイント1の情報を設定し、ED1にリンクされるTD11に基づいて、エンドポイント1との間でデータ転送を行う。次に、ED2に基づいてエンドポイント2の情報を設定し、ED2にリンクされるTD21に基づいて、エンドポイント2との間でデータ転送を行う。同様にして、データ転送制御装置600は、TD31、TD12、TD32、TD13に基づいてデータ転送を実行する。

#### 【0054】

インタラプト転送を行う場合には、CPU610上で動作するファームウェア（ホストコ

50

ントローラ・ドライバ)は、図4に示すような2分木(binary tree)構造のディスクリプタを作成する。例えば、1ms毎にインタラプト転送のポーリングを行うエンドポイントについては、そのディスクリプタを図4のプレイスホルダ(holder)700に設定する。同様に、2ms毎にポーリングを行うエンドポイントについては、そのディスクリプタをプレイスホルダ701、702に設定し、4ms毎にポーリングを行うエンドポインタについては、プレイスホルダ703、704、705、706に設定する。

#### 【0055】

そして、ポーリングを行う際には、インタラプトヘッドポインタのインデックスにしたがって、最下位層のプレイスホルダから順に2分木探索を行う。即ち、図4の経路710に示すように、まずインデックス0について最下位層から2分木探索を行う。次に、経路711に示すように、インデックス1について2分木探索を行う。同様にしてインデックス2~31について2分木探索を行う。これにより、プレイスホルダ700に対応するエンドポイントについては1ms(1フレーム)毎に、701、702に対応するエンドポイントについては2ms毎に、703~706に対応するエンドポイントについては4ms毎にインタラプト転送が行われるようになる。

#### 【0056】

以上のように、OHC1準拠のデータ転送制御装置(ホストコントローラ)では、CPU上で動作するファームウェア(ホストコントローラ・ドライバ)が、図3(A)、図4に示すような複雑な構造のディスクリプタを作成しなければならない。従って、CPUの処理負荷が非常に重い。

10

#### 【0057】

この場合、従来のUSBでは、ホストの役割が割り当てられるのはパーソナルコンピュータだけであり、このパーソナルコンピュータは高性能のCPUを有している。従って、図3(A)、図4に示すような複雑な構造のディスクリプタの作成も、余裕を持って行うことが可能であった。

20

#### 【0058】

ところが、OTGのターゲットアプリケーションである小型携帯機器(デジタルカメラ、携帯電話等)に組み込まれるCPU(embedded CPU)は、パーソナルコンピュータのCPUに比べて、性能が格段に低いのが一般的である。従って、携帯機器にOTGのホスト動作を行わせると、携帯機器に組み込まれるCPUに過大な負荷がかかり、他の処理に支障が生じたり、データ転送のパフォーマンスが低下するなどの問題が生じる。

30

#### 【0059】

##### 3. 構成例

図5に、以上のような問題を解決できる本実施形態のデータ転送制御装置の構成例を示す。なお、本実施形態のデータ転送制御装置は、図5の全ての回路ブロックを含む必要はなく、その一部の回路ブロックを省略してもよい。

#### 【0060】

トランシーバ10(以下、適宜Xcvrと呼ぶ)は、差動データ信号DP、DMを用いてUSB(広義にはバス)のデータを送受信する回路であり、USBの物理層(PHY)回路12を含む。より具体的にはトランシーバ10は、DP、DMのラインステート(J、K、SE0等)の生成、シリアル/パラレル変換、パラレル/シリアル変換、ビットスタッフィング、ビットアンスタッフィング、NRZIデコード、NRZIエンコードなどをを行う。なお、トランシーバ10をデータ転送制御装置の外部に設けるようにしてもよい。

40

#### 【0061】

OTGコントローラ20(広義にはストートコントローラ。以下、適宜OTGCと呼ぶ)は、OTGのSRP機能やHNP機能(図2(A)、(B)参照)を実現するための種々の処理を行う。即ち、OTGコントローラ20は、ホストの役割として動作するホスト動作のストートや、ペリフェラルの役割として動作するペリフェラル動作のストートなどを含む複数のストートの制御を行う。

#### 【0062】

50

より具体的には、OTG規格には、デュアルロール・デバイスのAデバイス時(図1(B)、(C)参照)のステート遷移やBデバイス時のステート遷移が定義されている。OTGコントローラ20は、これらのステート遷移を実現するためのステートマシンを含む。また、OTGコントローラ20は、USBのデータラインステートや、VBusレベルや、IDピンのステートを検出(監視)する回路を含む。そして、OTGコントローラ20が含むステートマシンは、これらの検出情報に基づいて、そのステート(例えば、ホスト、ペリフェラル、サスペンド又はアイドルなどのステート)を変化させる。この場合のステートの遷移は、ハードウェア回路により実現してもよいし、ファームウェアがステートコマンドをレジスタに設定することで実現してもよい。そして、ステートが遷移すると、OTGコントローラ20は、遷移後のステートに基づいて、VBusを制御したり、DP、DMのプルアップ抵抗/プルダウン抵抗の接続/非接続を制御する。また、ホストコントローラ50(以下、適宜HCと呼ぶ)、ペリフェラルコントローラ60(以下、適宜PCと呼ぶ)のイネーブル/ディスエーブルを制御する。

#### 【0063】

HC/PC切り替え回路30(HC/PC・コモン回路)は、トランシーバ10と、ホストコントローラ50又はペリフェラルコントローラ60との間の接続の切り替え制御を行う。また、USBのデータ(DP、DM)のラインステートの生成の指示をトランシーバ10に対して行う。なお、接続の切り替え制御は、HC/PCセレクタ32により実現され、ラインステートの生成指示は、ラインステートコントローラ34により実現される。

#### 【0064】

例えばOTGコントローラ20が、ホスト動作時(ホストステート時)にHCイネーブル信号をアクティブにすると、HC/PC切り替え回路30(HC/PCセレクタ32)は、トランシーバ10とホストコントローラ50を接続する。一方、OTGコントローラ20が、ペリフェラル動作時(ペリフェラルステート時)にPCイネーブル信号をアクティブにすると、HC/PC切り替え回路30は、トランシーバ10とペリフェラルコントローラ60を接続する。このようにすることで、ホストコントローラ50とペリフェラルコントローラ60とを排他的に動作させることができる。

#### 【0065】

転送コントローラ40は、USB(広義にはバス)を介したデータ転送を制御する回路であり、ホストコントローラ50(HC)とペリフェラルコントローラ60(PC)を含む。

#### 【0066】

ここでホストコントローラ50は、ホスト動作時(OTGコントローラ20からのHCイネーブル信号のアクティブ時)に、ホストの役割としてのデータ転送制御を行う回路である。

#### 【0067】

即ち、ホストコントローラ50は、ホスト動作時に、HC/PC切り替え回路30によりトランシーバ10に接続される。そしてホストコントローラ50は、レジスタ部70の転送条件レジスタ部72に設定された転送条件情報に基づいて、エンドポイントに対するトランザクションを自動発生する。そして、パケットバッファ100に確保(allocate)されたパイプ領域(PPIPE0~PPIPEe。以下、適宜PIPEと呼ぶ)と、そのパイプ領域に対応するエンドポイントとの間で、データ(パケット)の自動転送(処理部が介在しないハードウェア回路によるデータ転送)を行う。

#### 【0068】

より具体的にはホストコントローラ50は、複数のパイプ転送間の調停、フレームにおける時間管理、転送のスケジューリング、再送の管理などを行う。また、パイプ転送の転送条件情報(オペレーション情報)をレジスタ部70を介して管理する。また、トランザクションの管理を行ったり、パケットを生成/分解したり、サスペンド/レジューム/リセット状態生成の指示を行う。

#### 【0069】

10

20

30

40

50

一方、ペリフェラルコントローラ 60 は、ペリフェラル動作時 (OTG コントローラ 20 からの PC イネーブル信号のアクティブ時) に、ペリフェラルの役割としてのデータ転送制御を行う回路である。

#### 【0070】

即ち、ペリフェラルコントローラ 60 は、ペリフェラル動作時に、HC / PC 切り替え回路 30 によりトランシーバ 10 に接続される。そして、レジスタ部 70 の転送条件レジスタ部 72 に設定された転送条件情報に基づいて、パケットバッファ 100 に確保されたエンドポイント領域 (EP0 ~ EPe。以下適宜 EP と呼ぶ) とホストとの間でデータを転送する。

#### 【0071】

より具体的には、ペリフェラルコントローラ 60 は、エンドポイント転送の転送条件情報 (オペレーション情報) をレジスタ部 70 を介して管理する。また、トランザクションの管理を行ったり、パケットを生成 / 分解したり、リモート・ウェイクアップ信号生成の指示を行う。

#### 【0072】

なお、エンドポイントは、ユニークなアドレスを割り当てることができる、ペリフェラル (デバイス) 上のポイント (部分) である。ホストとペリフェラル (デバイス) との間でのデータ転送は、全て、このエンドポイントを経由して行われる。また、トランザクションは、トークンパケットと、オプショナルなデータパケットと、オプショナルなハンドシェークパケットにより構成される。

#### 【0073】

レジスタ部 70 は、データ転送 (パイプ転送、エンドポイント転送) 制御、バッファアクセス制御、バッファ管理、割り込み制御、ロック制御、或いは DMA 制御などを行うための種々のレジスタを含む。なお、レジスタ部 70 が含むレジスタは、RAM などのメモリにより実現してもよいし、D フリップフロップなどにより実現してもよい。また、レジスタ部 70 のレジスタは、1 つにまとめずに、各ブロック (HC、PC、OTGC、Xcvr 等) に分散して配置してもよい。

#### 【0074】

レジスタ部 70 は、転送条件レジスタ部 72 を含む。そして、この転送条件レジスタ部 72 は、ホスト動作時にパケットバッファ 100 に確保されるパイプ領域 (PIPE0 ~ PIPEe) とエンドポイントとの間でのデータ転送の転送条件情報 (転送制御情報) を記憶するレジスタを含む。これらの各転送条件レジスタは、パケットバッファ 100 の各パイプ領域に対応して設けられる。

#### 【0075】

なお、ペリフェラル動作時には、パケットバッファ 100 にはエンドポイント領域 (EP0 ~ EPe) が確保される。そして、転送条件レジスタ部 72 に設定された転送条件情報に基づいて、データ転送制御装置とホストとの間でのデータ転送が行われる。

#### 【0076】

バッファコントローラ 80 ( FIFO マネージャ ) は、パケットバッファ 100 に対するアクセス (リード / ライト) 制御や領域管理を行う。より具体的には、CPU (広義には処理部)、DMA (Direct Memory Access)、USB によるパケットバッファ 100 へのアクセス・アドレスを生成・管理する。また、CPU、DMA、USB によるパケットバッファ 100 へのアクセスの調停を行う。

#### 【0077】

例えば、ホスト動作時には、バッファコントローラ 80 は、インターフェース回路 110 (CPU 又は DMA) とパケットバッファ 100 の間のデータ転送経路と、パケットバッファ 100 とホストコントローラ 50 (USB) の間のデータ転送経路を設定 (確立) する。

#### 【0078】

一方、ペリフェラル動作時には、バッファコントローラ 80 は、インターフェース回路 1

10

20

30

40

50

10 ( C P U 又は D M A ) とパケットバッファ 100 の間のデータ転送経路と、パケットバッファ 100 とペリフェラルコントローラ 60 ( U S B ) の間のデータ転送経路を設定する。

#### 【 0079 】

パケットバッファ 100 ( F I F O 、パケットメモリ、バッファ) は、 U S B を介して転送されるデータ ( 送信データ又は受信データ ) を一時的に記憶 ( バッファリング ) するものである。このパケットバッファ 100 は、例えば R A M ( Random Access Memory ) などにより構成できる。なお、パケットバッファ 100 をデータ転送制御装置の外部に設けてもよい ( 外付けメモリにしてもよい ) 。

#### 【 0080 】

ホスト動作時には、パケットバッファ 100 はパイプ転送用の F I F O ( First-In First-Out ) として使用される。即ち、パケットバッファ 100 には、 U S B ( バス ) の各エンドポイントに対応するように、パイプ領域 P I P E 0 ~ P I P E e ( 広義にはバッファ領域 ) が確保される。また、各パイプ領域 P I P E 0 ~ P I P E e には、それに対応する各エンドポイントとの間で転送されるデータ ( 送信データ又は受信データ ) が記憶される。

10

#### 【 0081 】

一方、ペリフェラル動作時には、パケットバッファ 100 はエンドポイント転送用の F I F O として使用される。即ち、パケットバッファ 100 には、エンドポイント領域 E P 0 ~ E P e ( 広義にはバッファ領域 ) が確保される。また、各エンドポイント領域 E P 0 ~ E P e には、ホストとの間で転送されるデータ ( 送信データ又は受信データ ) が記憶される。

20

#### 【 0082 】

なお、パケットバッファ 100 に確保されるバッファ領域 ( ホスト動作時にパイプ領域に設定され、ペリフェラル動作時にエンドポイント領域に設定される領域 ) は、先に入力された情報が先に出力されるような記憶領域 ( F I F O 領域 ) に設定されている。

#### 【 0083 】

また、 P I P E 0 は、コントロール転送用のエンドポイント 0 に専用のパイプ領域であり、 P I P E a ~ P I P E e は、任意のエンドポイントに割り当て可能な汎用のパイプ領域である。

#### 【 0084 】

即ち、 U S B では、エンドポイント 0 がコントロール転送に専用のエンドポイントに設定される。従って、本実施形態のように P I P E 0 をコントロール転送に専用のパイプ領域にすることで、ユーザが混乱するのを防止できる。また、 P I P E a ~ P I P E e を、任意のエンドポイントに割り当て可能なパイプ領域にすることで、エンドポイントに対応するパイプ領域を動的に変化させることが可能になる。これにより、パイプ転送のスケジューリングの自由度を向上でき、データ転送の効率化を図れる。

30

#### 【 0085 】

なお本実施形態では、バッファ領域 ( パイプ領域又はエンドポイント領域 ) は、マックスパケットサイズ MaxPktSize ( 広義にはページサイズ ) とページ数 BufferPage により、その領域サイズ RSize が設定される ( RSize=MaxPktSize × BufferPage ) 。このようにすることで、バッファ領域の領域サイズや面数 ( ページ数 ) を任意に設定できるようになり、パケットバッファ 100 のリソースの有効利用を図れる。

40

#### 【 0086 】

インターフェース回路 110 は、 U S B とは異なる他のバスである D M A ( システムメモリ ) バスや C P U バスと、パケットバッファ 100 との間でのデータ転送を行うための回路である。このインターフェース回路 110 は、パケットバッファ 100 と外部のシステムメモリとの間で、 D M A 転送を行うための D M A ハンドラ回路 112 を含む。また、パケットバッファ 100 と外部の C P U との間で、 P I O ( Parallel I/O ) 転送を行うための C P U インターフェース回路 114 を含む。なお、 C P U ( 処理部 ) をデータ転送制御装置に内蔵させてもよい。

50

**【 0 0 8 7 】**

クロックコントローラ 120 は、内蔵 PLL 又は外部入力クロックに基づいて、データ転送制御装置の内部で使用する各種のクロックを生成する。

**【 0 0 8 8 】****4 . パイプ領域**

本実施形態では図 6 (A) に示すように、ホスト動作時に、パケットバッファ 100 にパイプ領域 PIPE0 ~ PIPEe が確保 (allocate) される。そして、この各パイプ領域 PIPE0 ~ PIPEe とペリフェラルの各エンドポイントの間で、データが転送される。

**【 0 0 8 9 】**

ここで、本実施形態のパイプ領域の「パイプ」は、USB で定義される「パイプ」（デバイス上のエンドポイントとホスト上のソフトウェアとの間の関連を表す論理的な抽象化、論理的な経路）とは若干意味合いが異なる。

10

**【 0 0 9 0 】**

本実施形態のパイプ領域は、図 6 (A) に示すように、USB (バス) に接続されるペリフェラルが有する各エンドポイントに対応して、パケットバッファ 100 上に確保される。例えば図 6 (A) において、パイプ領域 PIPEa はペリフェラル 1 のエンドポイント 1 (バルク IN) に対応し、PIPEb はペリフェラル 1 のエンドポイント 2 (バルク OUT) に対応する。また、PIPEc はペリフェラル 2 のエンドポイント 1 (バルク IN) に対応し、PIPEd はペリフェラル 2 のエンドポイント 2 (バルク OUT) に対応する。また、PIPEe はペリフェラル 3 のエンドポイント 1 (インターラプト IN) に対応する。なお、PIPE0 は、コントロール転送のエンドポイント 0 に専用のパイプ領域である。

20

**【 0 0 9 1 】**

そして図 6 (A) の例では、パイプ領域 PIPEa とペリフェラル 1 のエンドポイント 1 との間で USB のバルク IN 転送が行われ、PIPEb とペリフェラル 1 のエンドポイント 2 との間ではバルク OUT 転送が行われる。また、PIPEc とペリフェラル 2 のエンドポイント 1 との間ではバルク IN 転送が行われ、PIPEd とペリフェラル 2 のエンドポイント 2 との間ではバルク OUT 転送が行われる。また、PIPEe とペリフェラル 3 のエンドポイント 1 との間ではインターラプト IN 転送が行われる。

30

**【 0 0 9 2 】**

このように本実施形態では、パイプ領域（汎用）とそれに対応するエンドポイントとの間では、任意のデータ転送（アイソクロナス転送、バルク転送、インターラプト転送）を行うことができる。

**【 0 0 9 3 】**

そして本実施形態では、パイプ領域とそれに対応するエンドポイントの間では、所与のデータ単位（トータルサイズで指定されるデータ単位）のデータが転送される。この場合のデータ単位としては、例えば IRP (I/O リクエストパケット) により転送要求されるデータ単位又はそれを適当なサイズに分割したデータ単位などを考えることができる。エンドポイントに対するこのデータ単位のデータ転送（一連のトランザクション）を、本実施形態における「パイプ」と呼ぶことができる。そして、そのような「パイプ」のデータ（送信データ、受信データ）を記憶する領域がパイプ領域になる。

40

**【 0 0 9 4 】**

パイプ領域を用いた所与のデータ単位の転送が終了すると、そのパイプ領域は解放することができる。そして、解放されたパイプ領域は、任意のエンドポイントに割り当てることができる。このように本実施形態では、パイプ領域とエンドポイントとの対応づけを、動的に変化させることができる。

**【 0 0 9 5 】**

また本実施形態では図 6 (B) に示すように、ペリフェラル動作時には、パケットバッファ 100 にエンドポイント領域 EPO ~ EPe が確保（設定）される。そして、この各エ

50

ンドポイント領域 E P 0 ~ E P e とホスト（ホストコントローラ、システムメモリ）との間で、データが転送される。

#### 【 0 0 9 6 】

このように本実施形態では、パケットバッファ 100 のバッファ領域を、ホスト動作時にはパイプ領域に割り当て、ペリフェラル動作時にはエンドポイント領域に割り当てている。これにより、ホスト動作時とペリフェラル動作時とでパケットバッファ 100 のリソースを共用（兼用）することが可能になり、パケットバッファ 100 の使用記憶容量を節約できる。

#### 【 0 0 9 7 】

なおパイプ領域、エンドポイント領域の個数は 6 個に限定されず任意である。

10

#### 【 0 0 9 8 】

##### 5 . 転送条件レジスタ（共用レジスタ）

本実施形態では図 7 に示すように、ホスト動作時には、パイプ領域 P I P E 0 ~ P I P E e とエンドポイントとの間で行われるデータ転送の転送条件情報が、転送条件レジスタ T R E G 0 ~ T R E G e に設定される。即ち、P I P E 0 、P I P E a 、P I P E b 、P I P E c 、P I P E d 、P I P E e の転送条件情報は、各々、T R E G 0 、T R E G a 、T R E G b 、T R E G c 、T R E G d 、T R E G e に設定（記憶）される。この設定は、例えばファームウェア（C P U ）により行われる。

#### 【 0 0 9 9 】

そしてホストコントローラ 50（広義には転送コントローラ）は、転送条件レジスタ T R E G 0 ~ T R E G e に設定された転送条件情報に基づいて、エンドポイントに対するトランザクションを発生する。そして、パイプ領域とそれに対応するエンドポイントとの間で、データ（パケット）を自動転送する。

20

#### 【 0 1 0 0 】

このように本実施形態では、各パイプ領域（バッファ領域）に対応して各転送条件レジスタが設けられ、この各転送条件レジスタに設定された転送条件情報に基づいて、各パイプ領域のパイプ転送（所与のデータ単位の転送）がホストコントローラ 50 により自動的に行われる。従って、ファームウェア（ドライバ、ソフトウェア）は、転送条件レジスタに転送条件情報を設定した後は、データ転送が完了するまで、データ転送制御に関わらなくて済むようになる。そして、所与のデータ単位のパイプ転送が完了すると割り込みが発生し、転送の完了がファームウェアに伝えられる。これにより、ファームウェア（C P U ）の処理負荷を格段に低減できる。

30

#### 【 0 1 0 1 】

なお、本実施形態では図 8 に示すようにペリフェラル動作時には、エンドポイント領域 E P 0 ~ E P e とホストとの間で行われるデータ転送の転送条件情報が、転送条件レジスタ T R E G 0 ~ T R E G e に設定される。そしてペリフェラルコントローラ 60（広義には転送コントローラ）は、転送条件レジスタ T R E G 0 ~ T R E G e に設定された転送条件情報に基づいて、エンドポイント領域とホストとの間でのデータ転送を行う。

#### 【 0 1 0 2 】

このように本実施形態では、転送条件レジスタ T R E G 0 ~ T R E G e が、ホスト動作時とペリフェラル動作時とで共用（兼用）される。これにより、レジスタ部 70 のリソースを節約でき、データ転送制御装置を小規模化できる。

40

#### 【 0 1 0 3 】

図 9 に、レジスタ部 70 のレジスタ構成例を示す。なお、レジスタ部 70 のレジスタの一部を、各ブロック（O T G C 、H C 、P C 、X c v r 等）内に含ませてもよい。

#### 【 0 1 0 4 】

図 9 に示すように、レジスタ部 70 の転送条件レジスタ（T R E G 0 ~ T R E G e の各々）は、ホスト動作時（H C 、P I P E ）とペリフェラル動作時（P C 、E P ）で共用される H C / P C 共用レジスタ（共用転送条件レジスタ）を含む。また、ホスト動作時にのみ使用される H C （P I P E ）用レジスタ（ホスト用転送条件レジスタ）を含む。また、ペ

50

リフェラル動作時にのみ使用される P C ( E P ) 用レジスタ ( ペリフェラル用転送条件レジスタ ) を含む。また、パケットバッファ ( F I F O ) のアクセス制御などを行うためのレジスタであり、ホスト動作時とペリフェラル動作時で共用されるアクセス制御レジスタを含む。

#### 【 0 1 0 5 】

例えば、デュアルロール・デバイスのホスト動作時に、ホストコントローラ 5 0 ( H C ) は、 H C / P C 共用レジスタと H C 用レジスタに設定される転送条件情報に基づいて、データ ( パケット ) を転送する。

#### 【 0 1 0 6 】

一方、ペリフェラル動作時には、ペリフェラルコントローラ 6 0 ( P C ) は、 H C / P C 共用レジスタと P C 用レジスタに設定される転送条件情報に基づいて、データ ( パケット ) を転送する。  
10

#### 【 0 1 0 7 】

また、ホスト動作時、ペリフェラル動作時の両方において、バッファコントローラ 8 0 は、共用アクセス制御レジスタに基づいて、パケットバッファ 1 0 0 へのアクセス制御 ( リード / ライト・アドレスの発生、データのリード / ライト、アクセスの調停等 ) を行うことになる。

#### 【 0 1 0 8 】

図 9 の H C / P C 共用レジスタには、データの転送方向 ( I N 、 O U T 又は S E T U P 等 ) 、転送種別 ( アイソクロナス、バルク、インタラプト、コントロールなどのトランザクションの種別 ) 、エンドポイント番号 ( 各 U S B デバイスのエンドポイントに関連づけられる番号 ) 、マックスパケットサイズ ( エンドポイントが送信又は受信可能なパケットの最大ペイロードサイズ。ページサイズ ) が設定される。また、バッファ領域 ( パイプ領域、エンドポイント領域 ) のページ数 ( バッファ領域の面数 ) が設定される。また、 D M A 接続の有無 ( 図 5 の D M A ハンドラ回路 1 1 2 による D M A 転送の使用の有無 ) を指示する情報が設定される。  
20

#### 【 0 1 0 9 】

H C ( P I P E ) 用レジスタには、インタラプト転送のトークン発行周期 ( インタラプト・トランザクションを起動する周期、インターバル ) が設定される。また、トランザクションの連続実行回数 ( パイプ領域間の転送比率を設定する情報。各パイプ領域のトランザクションの連続実行回数 ) が設定される。また、ファンクションアドレス ( エンドポイントを有するファンクションの U S B アドレス ) 、転送データのトータルサイズ ( 各パイプ領域を介して転送されるデータのトータルサイズ。 I R P などのデータ単位 ) が設定される。また、自動トランザクションの開始指示 ( ホストコントローラに対する自動トランザクション処理の開始指示 ) が設定される。また、自動コントロール転送モードの指示 ( コントロール転送のセットアップステージ、データステージ、ステータスステージのトランザクションを自動発生するモードの指示 ) が設定される。  
30

#### 【 0 1 1 0 】

P C ( E P ) 用レジスタには、エンドポイントイネーブル ( エンドポイントのイネーブルやディスエーブルの指示 ) 、ハンドシェーク指定 ( 各トランザクションで行われるハンドシェークの指定 ) が設定される。  
40

#### 【 0 1 1 1 】

パケットバッファ ( F I F O ) 用の共用アクセス制御レジスタには、バッファ・ I / O ポート ( C P U により P I O 転送を行う場合の I / O ポート ) が設定される。また、バッファ・フル / エンプティ ( 各バッファ領域のフル、エンプティの通知 ) 、バッファ・残りデータサイズ ( 各バッファ領域の残りデータサイズ ) が設定される。

#### 【 0 1 1 2 】

レジスタ部 7 0 は、インタラプト系レジスタ、ブロック系レジスタ、 D M A 制御レジスタなども含む。

#### 【 0 1 1 3 】

10

20

30

40

50

インタラプト系レジスタは、割り込みのステータス（要因）を C P U に対して示すためのインタラプト・ステータスレジスタ、割り込みのイネーブル、ディスエーブル（非マスク、マスク）を設定するインタラプト・イネーブルレジスタを含む。なお、割り込みには、O T G コントローラ 2 0 系、ホストコントローラ 5 0 系、ペリフェラルコントローラ 6 0 系の割り込みがある。

#### 【 0 1 1 4 】

ブロック系レジスタは、ブロック間で共用されるブロック間共用レジスタや、各ブロック（X c v r、O T G C、H C、P C）内で使用されるブロック用レジスタを含む。

#### 【 0 1 1 5 】

ブロック間共用レジスタには、各ブロックのリセットを指示するレジスタなどがある。ブロック用レジスタには、トランシーバ 1 0（X c v r）を制御するためのレジスタや、O T G コントローラ 2 0（O T G C）のステートコマンドレジスタや、ホストコントローラ 5 0（H C）のステートコマンドレジスタや、フレーム番号を設定するレジスタなどがある。

10

#### 【 0 1 1 6 】

以上のように本実施形態では、ホスト動作時とペリフェラル動作時で共用されるレジスタ（H C / P C 共用レジスタ、共用アクセス制御レジスタ）をレジスタ部 7 0 に設けている。これにより、ホスト動作時用のレジスタとペリフェラル動作時用のレジスタを全く別個に設ける場合に比べて、レジスタ部 7 0 を小規模化できる。また、C P U 上で動作するファームウェア（ドライバ）から見た共用レジスタのアクセスアドレスを、ホスト動作時とペリフェラル動作時とで同一にできる。従って、ファームウェアは、これらの共用レジスタを同一アドレスで管理できるようになり、ファームウェア処理を簡素化できる。

20

#### 【 0 1 1 7 】

また、H C 用レジスタや、P C 用レジスタを設けることで、ホスト動作時（P I P E）の転送やペリフェラル動作時（E P）の転送に特有の転送条件を設定できる。例えば、トークン発行周期を設定することで、図 4 で説明した手法を用いることなく、ホスト動作時にインタラプト転送のトークンを所望の周期で発行することが可能になる。また、連続実行回数を設定することで、ホスト動作時にパイプ領域間の転送比率を任意に設定できる。また、トータルサイズを設定することで、ホスト動作時にパイプ領域を介して自動転送されるデータのサイズを任意に設定できる。またファームウェアは、ホスト動作時に、自動トランザクションの開始を指示したり、自動コントロール転送モードのオン／オフを指示できるようになる。

30

#### 【 0 1 1 8 】

なお、図 1 0 に、汎用の転送条件（P I P E / E P）レジスタ T R E G a ~ T R E G e (T R E G x : x = a ~ e) のレジスタマップの詳細例を示し、図 1 1 に、コントロール転送用の転送条件レジスタ T R E G 0 のレジスタマップの詳細例を示す。また、図 1 2 (A)、(B)、図 1 3 に、これらの転送条件レジスタの各ビットフィールドに設定される転送条件情報 (JoinDMA, FIFOIr, ToggleMode, AutoZeroLen, BufferPage, DirPID, TranType, EP Number, MaxPktSize 等) の概要を示す。

40

#### 【 0 1 1 9 】

例えば図 1 0 において、ホスト動作時とペリフェラル動作時とで、共用レジスタ xConfig\_0, xConfig\_1, xMaxPktSize\_H, xMaxPktSize\_L のアクセスアドレス 0x0, 0x1, 0x2, 0x3 は同一になる。またホスト動作時とペリフェラル動作時とで、これらのレジスタには同内容の情報がファームウェアにより設定されることになる。

#### 【 0 1 2 0 】

##### 6 . 自動トランザクション

図 1 4 に、ホストコントローラ 5 0 の自動トランザクション (I N、O U T) 処理時におけるファームウェア処理のフローチャート例を示す。

#### 【 0 1 2 1 】

まず、ファームウェア（処理部、ドライバ）は、図 9 等で説明した転送条件レジスタに転

50

送条件情報（パイプ情報）を設定する（ステップS1）。より具体的には、転送データのトータルサイズ、マックスパケットサイズ（MaxPktSize）、ページ数（BufferPage）、転送方向（IN、OUT又はSETUP）、転送種別（アイソクロナス、バルク、コントロール、インターラプト）、エンドポイント番号、パイプ領域のトランザクションの連続実行回数（転送比率）、インターラプト転送のトークン発行周期などを、転送条件レジスタに設定する。

#### 【0122】

次に、外部のシステムメモリとパケットバッファ100の間に転送経路を設定する（ステップS2）。即ち図5のDMAハンドラ回路112を介したDMA転送経路を設定する。

#### 【0123】

次に、ファームウェアは、DMA転送の開始指示を行う（ステップS3）。即ち、図9のDMA制御レジスタのDMA転送開始指示ビットをアクティブにする。なお、CPUによる転送では、図9のバッファ・I/Oポートにアクセスすることで、パケットバッファ100にアクセスすることが可能になる。

#### 【0124】

次に、ファームウェアは、自動トランザクションの開始指示を行う（ステップS4）。即ち、図9のHC用レジスタ（パイプレジスタ）の自動トランザクション開始指示ビットをアクティブにする。これにより、ホストコントローラ50による、自動トランザクション処理、パケット処理（パケットの生成、分解）、スケジューリング処理が行われる。即ち、ホストコントローラ50は、トータルサイズで指定されるデータを、マックスパケットサイズのペイロードのパケットを用いて、転送方向で指定される方向（IN、OUT）で、自動転送する。

#### 【0125】

なお、ステップS3、S4の処理の順序は問わず、自動トランザクション開始指示の後にDMA転送の開始指示を行ってもよい。

#### 【0126】

次に、ファームウェアは、パイプ転送の完了を知らせる割り込みが発生するのを待つ（ステップS5）。そして、割り込みが発生すると、ファームウェアは、図9のインターラプト系レジスタの割り込みステータス（要因）を調べる。そして、処理が正常完了又はエラー終了する（ステップS6）。

#### 【0127】

このように本実施形態によれば、ファームウェアは、各パイプ領域毎に転送条件情報を設定し（ステップS1）、DMA転送開始の指示（ステップS3）と自動トランザクション開始の指示（ステップS4）を行うだけで、その後のデータ転送処理はホストコントローラ50のハードウェア回路により自動的に行われるようになる。従って、図3（A）、（B）、図4で説明したOHC1準拠の手法に比べて、ファームウェアの処理負荷が軽減され、低性能のCPUが組み込まれる携帯機器に最適なデータ転送制御装置を提供できる。

#### 【0128】

図15、図16に、ホストコントローラ50による自動トランザクション処理時の信号波形例を示す。なお、これらの図において、「H P」は「ホストからペリフェラルにパケットが転送されること」を表し、「P H」は「ペリフェラルからホストにパケットが転送されること」を表す。

#### 【0129】

図15は、INトランザクションの場合（転送種別がINの場合）の信号波形例である。

#### 【0130】

図14のステップS4で、ファームウェアが自動トランザクションの開始指示を行うと、図15のC1に示すように、PipeXTranGo（PipeXについてのファームウェアからの転送要求信号）がアクティブになる。これにより、そのPipeX（X = 0 ~ e）についての、ホストコントローラ50による自動トランザクション処理が開始する。

#### 【0131】

10

20

30

40

50

そして、C 2 に示すようにPipeTranGo（ホストコントローラ 50 内のHCシーケンス管理回路からの転送要求信号）がアクティブになると、C 3 に示すように、ホストコントローラ 50 がIN token パケットを生成して、USB を介してペリフェラルに転送する。そして、C 4 に示すようにペリフェラルからホストコントローラ 50 にINデータパケットが転送されると、ホストコントローラ 50 は、C 5 に示すようにハンドシェークパケット（ACK）を生成して、ペリフェラルに転送する。これにより、C 6 に示すようにTranCmpACKがアクティブになる。

#### 【0132】

同様にして、C 7 に示すようにPipeTranGoがアクティブになると、C 8、C 9、C 10 に示すパケット転送が行われ、C 11 に示すようにTranCmpACKがアクティブになる。そして、C 12 に示すようにPipeXTranComp（ファームウェアへのIRP のデータ単位の転送終了通知信号）がアクティブになる。このPipeXTranCompによる割り込みにより、ファームウェアは、そのパイプについての転送が完了したことを知ることができる。10

#### 【0133】

なお、PipeXTranCompがアクティブになると、C 13 に示すようにPipeXTranGoが非アクティブになり、そのパイプが非転送状態であることが示されるようになる。

#### 【0134】

図16は、OUTトランザクションの場合（転送種別がOUTの場合）の信号波形例である。20

#### 【0135】

ファームウェアが自動トランザクションの開始指示を行うと、E 1 に示すようにPipeXTranGoがアクティブになり、E 2 に示すようにPipeTranGoがアクティブになる。すると、ホストコントローラ 50 は、E 3 に示すようにOUT token パケットをペリフェラルに転送し、E 4 に示すようにOUTデータパケットを転送する。そして、E 5 に示すようにペリフェラルからハンドシェークパケット（ACK）が返ってくると、E 6 に示すようにTranCmpACKがアクティブになる。20

#### 【0136】

同様にして、E 7 に示すようにPipeTranGoがアクティブになると、E 8、E 9、E 10 に示すパケット転送が行われ、E 11 に示すようにTranCmpACKがアクティブになる。そして、E 12 に示すようにPipeXTranCompがアクティブになる。このPipeXTranCompによる割り込みにより、ファームウェアは、そのパイプについての転送が完了したことを知ることができる。なお、PipeXTranCompがアクティブになると、E 13 に示すようにPipeXTranGoが非アクティブになる。30

#### 【0137】

### 7. トランザクションの連続実行回数

本実施形態では、トランザクション（パイプ転送）の連続実行回数（広義には複数のパイプ領域間の転送比率情報）を転送条件レジスタに設定できる。具体的には図9で説明したように、この連続実行回数はHC用の転送条件レジスタに設定される。

#### 【0138】

例えば図17(A)では、パイプ領域PIPE a、PIPE b、PIPE c のトランザクションの連続実行回数は「1」に設定されている。この場合には、PIPE a、PIPE b、PIPE c、PIPE a、PIPE b、PIPE c ……というように、同一の転送比率でパイプ領域のデータの自動転送が行われる。40

#### 【0139】

これに対して図17(B)では、PIPE a、PIPE c のトランザクションの連続実行回数が、各々、「3」、「2」に設定されている。この場合に本実施形態では、PIPE a、PIPE a、PIPE a、PIPE a、PIPE b、PIPE b、PIPE c、PIPE c、PIPE a、PIPE a、PIPE a ……というように、パイプ領域のデータの自動転送が行われる。即ち、PIPE a、PIPE b、PIPE c の転送比率（パイプ転送の相対的な割合）が3対1対2に設定される。50

**【0140】**

これにより、例えば、PIPE a、PIPE c の転送データのトータルサイズが PIPE b よりも大きい場合に、PIPE a、PIPE c のデータを効率良く転送でき、効率的なスケジューリングが可能になる。

**【0141】**

なお、図17(B)の場合に、PIPE a や PIPE c のデータを連続して自動転送しないようにしてもよい。即ち、例えば、PIPE a、PIPE b、PIPE c、PIPE a、PIPE c、PIPE a … というような順序で、データを自動転送してもよい。このようにしても、PIPE a、PIPE b、PIPE c の転送比率を 3 対 1 対 2 に設定できる。

10

**【0142】****8. インタラプト転送のトークン発行周期**

USB には 4 種類の転送手法が定義されており、その一つであるインタラプト転送は、ホストが周期的にトークンを発行することで、ペリフェラル(デバイス)のポーリングを行うという転送である。

**【0143】**

そして、従来の OHCI 準拠のデータ転送制御装置では、インタラプト転送のトークン発行周期を設定するために、図4に示すような 2 分木構造のディスクリプタをファームウェアが作成する必要があった。このため、ファームウェアの処理負荷が重くなるという問題があった。

20

**【0144】**

そこで本実施形態では、インタラプト転送のトークン発行周期を例えばフレームオーダ(m s)で直接に、転送条件レジスタに設定できるようにしている。具体的には図9で説明したように、このトークン発行周期は HC 用の転送条件レジスタに設定される。

**【0145】**

そして、ファームウェア(処理部)が、転送条件レジスタにトークン発行周期を設定すると、ホストコントローラ50(転送コントローラ)は、設定されたトークン発行周期で、インタラプト転送のトークンパケットを自動生成して、ペリフェラルに転送する。

**【0146】**

即ち、本実施形態では、あるエンドポイントに対する転送を、1つの転送オーダであるパイプとして管理している。そして、ファームウェアはインタラプト転送を行うパイプを決定し、そのパイプの転送条件レジスタに、トークン発行周期を直接に数値で設定する。すると、ホストコントローラ50は、そのパイプ(パイプ領域)においては、その発行周期でインタラプト転送のトークンを発行するようになる。

30

**【0147】**

例えば図18(A)では、トークン発行周期レジスタ(転送条件レジスタの1つ)に、「1」が設定されている。すると、この場合には、1フレームのインターバルで、ホストコントローラ50によりインタラプト転送のトークンパケットが自動転送される。また、図18(B)、(C)では、トークン発行周期レジスタに、「2」、「4」が設定されており、この場合には、2フレーム、4フレームのインターバルでトークンパケットが自動転送される。

40

**【0148】**

このようにすることで、インタラプト転送を実現するために図4に示すような2分木構造のディスクリプタを作成する必要がなくなり、ファームウェアの処理負荷を軽減できる。

**【0149】**

特に、OHCI では、USB に多数のエンドポイントが接続されている場合にも対応できるように、図4に示すような2分木構造のディスクリプタを採用している。

**【0150】**

ところが、OTG では、多くの場合、ピア・ツー・ピアの接続が予定されており、図4に示すような2分木構造のディスクリプタを用いることは、処理の無駄になる。

50

**【0151】**

本実施形態では、このような処理の無駄に着目すると共に、パイプ領域毎に転送条件レジスタが用意されることに着目して、トークン発行周期を各転送条件レジスタに直接設定できるようにしたところに特徴がある。

**【0152】****9. 各ブロックの詳細な構成例**

次に各ブロックの詳細な構成例について説明する。

**【0153】****9.1 OTGコントローラ**

図19に、OTGコントローラ20の構成例を示す。 10

**【0154】**

OTGコントローラ20は、OTGレジスタ部22を含む。このOTGレジスタ部22は、OTGコントローラ20のモニタレジスタや制御レジスタを含む。またファームウェア(CPU)により書き込まれるOTGステートコマンドをデコードする回路を含む。

**【0155】**

またOTGコントローラ20はOTG制御回路23を含む。そして、このOTG制御回路23は、OTGステートの管理を行うOTG管理回路24、IDピンの電圧レベルを検出するID検出回路25、VBUSの電圧レベルを検出するVBUS検出回路26、DP及びDMのラインステートを検出するラインステート検出回路27を含む。 20

**【0156】**

またOTGコントローラ20は、OTGステートの遷移判断条件の1つである時間を計測するタイマ28を含む。

**【0157】**

OTGステートを遷移させるために検出すべき情報は、ID、VBUSの電圧レベル、DP/DMのラインステートである。本実施形態のOTGコントローラ20は、これらの情報を検出し、モニタレジスタを介してファームウェア(CPU)に伝える。

**【0158】**

ファームウェアは、これらの検出情報に基づいて自身のステートを遷移させると共に、次に遷移すべきステートを、OTGステートコマンドを用いてOTGコントローラ20に伝える。 30

**【0159】**

OTGコントローラ20は、OTGステートコマンドをデコードし、そのデコード結果に基づいて、VBUSのドライブ制御、プルアップ/プルダウン抵抗の接続制御等を行い、図2(A)、(B)で説明したSRPやHNPを実現する。

**【0160】**

このように本実施形態では、ステート毎のOTG制御はOTGコントローラ20が担当し、ファームウェアはステートの遷移管理に専念できる。この結果、全てのステート制御をファームウェアで実現する場合に比べて、ファームウェア(CPU)の処理負荷を軽減できると共に、効率的なファームウェア開発が可能になる。

**【0161】**

なお、OTGのステート遷移の判断を、ファームウェアが行わずに、ハードウェア回路が行うようにしてもよい。或いは、OTGコントローラ20のほとんど全ての処理(例えばVBUS制御、プルアップ/プルダウン抵抗制御、ID検出、VBUS検出、ラインステート検出以外の処理)をファームウェア(ソフトウェア)により実現してもよい。

**【0162】****9.2 ホストコントローラ、ペリフェラルコントローラ**

図20(A)に、ホストコントローラ50の構成例を示す。

**【0163】**

ホストコントローラ50はHCシーケンス管理回路52を含む。このHCシーケンス管理回路52は、パイプ転送(パイプ領域を用いたデータ転送)の調停、時間管理、パイプ転 50

送のスケジューリング、再送管理などを行う。

**【0164】**

より具体的にはH C シーケンス管理回路5 2は、フレーム番号のカウントや、S O F (Start-Of-Frame) パケットの送信指示を行う。また、アイソクロナス転送を各フレームの先頭で優先的に実行するための処理を行ったり、インタラプト転送をアイソクロナス転送の次に優先的に取り扱うための処理を行う。また、パイプ転送の順序にしたがって各パイプ転送を指示する処理を行う。また、トランザクションの連続実行回数を管理したり、残りフレーム時間の確認処理を行う。また、ペリフェラルから返ってきたハンドシェークパケット(ACK、NAK)に対する処理を行う。また、トランザクション実行時のエラー処理を行う。

10

**【0165】**

ホストコントローラ5 0はターゲットパイプ管理回路5 4を含む。このターゲットパイプ管理回路5 4は、レジスタ部7 0の転送条件レジスタに設定された転送条件情報のハンドリング処理などを行う。

**【0166】**

より具体的にはターゲットパイプ管理回路5 4は、転送条件情報の選択処理や、割り込み信号の生成処理を行う。また自動トランザクションの開始が指示された場合に、そのパイプ領域の転送データのトータルサイズをロードする。そして、残り転送データサイズのカウント(デクリメント)処理を行う。また、バッファコントローラ8 0へのデータの送受信の際にバッファ(FIFO)領域の状態を確認する処理を行う。また、トランザクション管理回路5 6への転送指示を行う。また、予期しないショートパケットの受信の判断処理や、マックスパケットサイズ以上のパケットの受信の判断処理を行う。また、零長パケットを自動転送するモードが設定されている場合には、最後の零長パケットの送信をトランザクション管理回路5 6に指示する。また、自動コントロール転送モードでのシーケンス管理を行う。

20

**【0167】**

ホストコントローラ5 0はトランザクション管理回路5 6を含む。このトランザクション管理回路5 6は、転送パケットの種類や転送順序の管理(トランザクションのシーケンス管理)を行う。また、タイムアウトの監視処理を行う。また、トランザクション終了の通知処理を行う。

30

**【0168】**

ホストコントローラ5 0はパケットハンドラ回路5 8を含む。このパケットハンドラ回路5 8は、パケットの生成、分解処理を行う。また、P I DのチェックやC R Cのデコード、エンコードを行う。また、バッファ領域のパケットのペイロードのリード、ライト処理や、S O Fパケットの送信処理を行う。また、送受信データのカウント処理を行う。

**【0169】**

図20(B)にペリフェラルコントローラ6 0の構成例を示す。

**【0170】**

ペリフェラルコントローラ6 0は、トランザクション管理回路6 2、パケットハンドラ回路6 4を含む。これらのトランザクション管理回路6 2、パケットハンドラ回路6 4は、ホストコントローラ5 0のトランザクション管理回路5 6、パケットハンドラ回路5 8とほぼ同様の処理を行う。

40

**【0171】**

9.3 バッファコントローラ

図21にバッファコントローラ8 0の構成例を示す。

**【0172】**

バッファコントローラ8 0は領域確保(allocation)回路8 2を含む。この領域確保回路8 2は、パケットバッファ1 0 0に、バッファ領域(ホスト動作時にパイプ領域に設定され、ペリフェラル動作時にエンドポイント領域に設定される領域)を確保する回路である。

50

**【 0 1 7 3 】**

領域確保回路 8 2 は領域計算回路 8 3 を含む。この領域計算回路 8 3 は、マックスパケットサイズ(広義にはページサイズ)やページ数に基づいて、バッファ領域の領域サイズ、スタートアドレス、エンドアドレスなどを計算する回路である。

**【 0 1 7 4 】**

例えば図 2 2 ( A ) に示すバッファ領域 P I P E 0 / E P 0 、 P I P E a / E P a 、 P I P E b / E P b 、 P I P E c / E P c では、マックスパケットサイズ( MaxPktSize )が、各々、 3 2 、 6 4 、 6 4 バイトに設定され、ページ数( BufferPage )が、各々、 1 、 1 、 3 、 2 ページに設定されている。領域計算回路 8 3 は、これらのマックスパケットサイズ、ページ数などに基づいて、バッファ領域 P I P E 0 / E P 0 ~ P I P E c / E P c の領域サイズ、スタートアドレス、エンドアドレスを計算する。例えば図 2 2 ( A ) において、 P I P E 0 / E P 0 、 P I P E a / E P a 、 P I P E b / E P b 、 P I P E c / E P c の領域サイズは、各々、 3 2 (= 3 2 × 1 ) 、 6 4 (= 6 4 × 1 ) 、 1 9 2 (= 6 4 × 3 ) 、 1 2 8 (= 6 4 × 2 ) バイトと計算されることになる。  
10

**【 0 1 7 5 】**

ポインタ割り当て回路 8 4 は、各バッファ領域の書き込みポインタ W P ( W P 0 、 W P a 、 W P b 、 W P c ) 、読み出しポインタ R P ( R P 0 、 R P a 、 R P b 、 R P c ) を、 D M A 用ポインタ、 C P U 用ポインタ、 U S B 用ポインタに割り当てる回路である。

**【 0 1 7 6 】**

例えば図 2 2 ( B ) に示すように、データ送信時( D M A 又は C P U からパケットバッファ 1 0 0 を介して U S B 側にデータが転送される時)であり、且つ、 D M A 転送使用時には、そのバッファ領域の書き込みポインタ W P は D M A ( D M A アクセス)用のポインタに割り当てられ、読み出しポインタ R P は U S B ( U S B アクセス)用のポインタに割り当たられる。また、データ送信時であり且つ C P U ( P I O ) 転送使用時には、そのバッファ領域の書き込みポインタ W P は C P U ( C P U アクセス)用のポインタに割り当てられ、読み出しポインタ R P は U S B 用のポインタに割り当たられる。  
20

**【 0 1 7 7 】**

一方、図 2 2 ( C ) に示すように、データ受信時( U S B からパケットバッファ 1 0 0 を介して D M A 又は C P U 側にデータが転送される時)であり、且つ、 D M A 転送使用時には、そのバッファ領域の書き込みポインタ W P は U S B 用ポインタに割り当てられ、読み出しポインタ R P は D M A 用ポインタに割り当たられる。また、データ受信時であり且つ C P U 転送使用時には、そのバッファ領域の書き込みポインタ W P は U S B 用ポインタに割り当てられ、読み出しポインタ R P は C P U 用ポインタに割り当たられる。  
30

**【 0 1 7 8 】**

なお、各バッファ領域の書き込みポインタ W P 、読み出しポインタ R P のポインタ情報(位置情報)は、レジスタ部 7 0 の各転送条件レジスタ( P I P E / E P レジスタ)に保持される。

**【 0 1 7 9 】**

ポインタ管理回路 8 6 は、ポインタの更新を行いながら、パケットバッファ 1 0 0 にアクセスするための実アドレスを生成する回路である。  
40

**【 0 1 8 0 】**

ポインタ管理回路 8 6 は、 C P U 用アドレス生成回路 8 7 、 D M A 用アドレス生成回路 8 8 、 U S B 用アドレス生成回路 8 9 を含む。これらの生成回路 8 7 、 8 8 、 8 9 は、各々、ポインタ割り当て回路 8 4 により割り当てられた C P U 用ポインタ、 D M A 用ポインタ、 U S B 用ポインタに基づいて、 C P U 用アドレス、 D M A 用アドレス、 U S B 用アドレスを生成する。また、 C P U ( C P U インターフェース回路)、 D M A ( D M A ハンドラ回路)からのアクセス毎に、或いは U S B ( H C 又は P C )のトランザクション終了( A C K 、 N A K などのハンドシェーク送受信)毎に、ポインタを更新する処理を行う。なお、更新後のポインタの情報は、領域確保回路 8 2 を介してレジスタ部 7 0 の各転送条件レジスタに書き戻される。  
50

**【0181】**

バッファ管理回路90は、パケットバッファ100へのアクセスを管理する回路である。

**【0182】**

バッファ管理回路90はバッファインターフェース回路92を含む。このバッファインターフェース回路92は、ポイント管理回路86からのCPU用アドレス、DMA用アドレス、USB用アドレスなどを受け、パケットバッファ100へのデータの入出力や、アドレス、出カイネーブル、ライトイネーブル、リードイネーブルなどの出力を行う。

**【0183】**

バッファ管理回路90は調停回路93を含む。この調停回路93は、CPU(CPUインターフェース回路)、DMA(DMAハンドラ回路)、USB(ホストコントローラ又はペリフェラルコントローラ)からのアクセスを調停する回路である。この調停結果に基づいて、CPU用アドレス、DMA用アドレス、USB用アドレスのいずれかが、パケットバッファ100のアクセス・アドレスとして出力され、CPU、DMA又はUSBとパケットバッファ100との間のデータ転送経路が設定される。10

**【0184】**

H C / P C セレクタ94は、バッファ管理回路90(バッファコントローラ80)とホストコントローラ50(HC)又はペリフェラルコントローラ60(PC)との間の接続の切り替え制御を行う。例えばホスト動作時には、ホストコントローラ50とバッファ管理回路90を接続し、ペリフェラル動作時には、ペリフェラルコントローラ60とバッファ管理回路90を接続する。なお、この接続の切り替え制御は、OTGコントローラ20(OTGC)からのH C / P C イネーブル信号に基づいて行う。20

**【0185】****10. フームウェアの処理**

次に、ファームウェア(処理部)の処理の詳細例について説明する。

**【0186】**

図23は、ホスト動作時におけるファームウェア処理のフローチャートである。

**【0187】**

まず、パイプ(バッファ)領域を既に確保しているか否かを確認し(ステップS11)、確保している場合にはパイプ領域のデータクリア指示[FIFOClr]を行う(ステップS12)。30

**【0188】**

次に、エンドポイント(転送条件)情報を転送条件レジスタにセットする(ステップS13)。即ち、エンドポイント番号[EPNumber]、ファンクションアドレス[FuncAddr]、IN/OUT/SETUPなどの転送方向[DirPID]、アイソクロナス、バルク、コントロール、インタラプトなどの転送種別[TranType]、マックスパケットサイズ[MaxPktSize]などをセットする。

**【0189】**

次に、転送種別[TranType]を判断し(ステップS14)、転送種別がアイソクロナスの場合にはステップS18に移行する。転送種別がインタラプト転送の場合にはトークン発行周期[Interval]を指定すると共にトグルモード[ToggleMode]を指定する(ステップS15、S16)。転送種別が、アイソクロナスでもインタラプトでもない場合(バルク、コントロールの場合)には、HCのスケジューリング[連続実行回数:Continuity]を指定する(ステップS17)。40

**【0190】**

次に、トグルビットの初期値[Toggle]をセットし、転送データのトータルサイズ[TotalSize]をセットする(ステップS18、S19)。なお、アイソクロナス転送の場合はトグルビットの初期値のセットは不要となる。また、ステップS13～S19の設定順序は任意である。

**【0191】**

次に、パイプ(バッファ)領域のページ数[BufferPage]をセットし(ステップS20)50

、パイプ領域確保の指示 [SetAllocation] を行う（ステップ S 2 1）。

**【0192】**

次に、DMAを用いるか否かを判断し、用いる場合にはDMAバスの接続指示 [JoinDMA] を行う（ステップ S 2 2、S 2 3）。また、自動トランザクションの開始指示 [TranGo] を行う（ステップ S 2 4）。

**【0193】**

そして、割り込みが発生するのを待ち（ステップ S 2 5）、正常完了、STALL返答、ハンドシェーク待ちのタイムアウト処理などを行う（ステップ S 2 6）。

**【0194】**

図24は、ペリフェラル動作時におけるファームウェア処理のフローチャートである。

10

**【0195】**

まず、エンドポイント（バッファ）領域のデータクリア指示 [FIFOClr] を行う（ステップ S 3 1）。そして、トグルビット初期値 [Toggle] をセットする（ステップ S 3 2）。

**【0196】**

次にエンドポイント（転送条件）情報をセットする（ステップ S 3 3）。即ち、エンドポイント番号 [EPNumber]、転送方向 [DirPID]、転送種別 [TranType]、マックスパケットサイズ [MaxPktSize]などをセットする。

**【0197】**

次に、エンドポイントイネーブル指示 [EnEndPoint] を行う（ステップ S 3 4）。そして、転送種別 [TranType] を判断し、インタラプト転送の場合にはトグルモード [ToggleMode] を指定する（ステップ S 3 5、S 3 6）。

20

**【0198】**

次に、エンドポイント（バッファ）領域のページ数 [BufferPage] をセットし（ステップ S 3 7）、エンドポイント領域確保の指示 [SetAllocation] を行う（ステップ S 3 8）。

。

**【0199】**

次に、DMAを用いるか否かを判断し、DMAを用いる場合にはDMAバスの接続指示 [JoinDMA] を行う（ステップ S 3 9、S 4 0）。

**【0200】**

次に、ホストからのトーケン受信の待ち状態になる（ステップ S 4 1）。そして、割り込みが発生するのを待ち（ステップ S 4 2）、正常完了（ACK受信）、NAK返信、STALL返信、ハンドシェーク待ちのタイムアウト処理などを行う（ステップ S 4 3）。

30

**【0201】**

### 11. 電子機器

次に、本実施形態のデータ転送制御装置を含む電子機器の例について説明する。

**【0202】**

例えば図25（A）に電子機器の1つであるプリンタの内部ブロック図を示し、図26（A）にその外観図を示す。CPU510（処理部）はシステム全体の制御などを行う。操作部511はプリンタをユーザが操作するためのものである。ROM516には、制御プログラム、フォントなどが格納され、RAM517（システムメモリ）はCPU510のワーク領域として機能する。DMAC518は、CPU510を介さずにデータ転送を行うためのDMAコントローラである。表示パネル519はプリンタの動作状態をユーザに知らせるためのものである。

40

**【0203】**

USBを介してパーソナルコンピュータ、デジタルカメラ、デジタルビデオカメラなどの他の機器から送られてきたシリアルの印刷データ（印字データ、画像データ）は、データ転送制御装置500によりパラレルの印刷データに変換される。そして、変換後のパラレル印刷データは、CPU510又はDMAC518により、印刷処理部（プリンタエンジン）512に送られる。そして、印刷処理部512においてパラレル印刷データに対して所与の処理が施され、プリントヘッダなどからなる印刷部（データの出力処理を行う装置

50

) 514により紙に印刷されて出力される。

**【0204】**

図25(B)に電子機器の1つであるデジタルカメラの内部ブロック図を示し、図26(B)にその外観図を示す。CPU520はシステム全体の制御などを行う。操作部521(シャッターボタン、操作ボタン等)はデジタルカメラをユーザが操作するためのものである。ROM526には制御プログラムなどが格納され、RAM527はCPU520のワーク領域として機能する。DMAC528はDMAコントローラである。

**【0205】**

CCD、レンズなどからなる撮像部(データの取り込み処理を行う装置)522により画像が撮像され、撮像された画像のデータは画像処理部524により処理される。そして、処理後の画像データは、CPU520又はDMAC528によりデータ転送制御装置500に送られる。データ転送制御装置500は、このパラレルの画像データをシリアルデータに変換し、USBを介してプリンタ、ストレージ装置、パーソナルコンピュータなどの他の機器に送信する。10

**【0206】**

図25(C)に電子機器の1つであるCD-RWドライブ(ストレージ装置)の内部ブロック図を示し、図26(C)にその外観図を示す。CPU530はシステム全体の制御などを行う。操作部531はCD-RWをユーザが操作するためのものである。ROM536には制御プログラムなどが格納され、RAM537はCPU530のワーク領域として機能する。DMAC538はDMAコントローラである。20

**【0207】**

レーザ、モータ、光学系などからなる読み取り&書き込み部(データの取り込み処理を行う装置又はデータの記憶処理を行うための装置)533によりCD-RW532から読み取られたデータは、信号処理部534に入力され、エラー訂正処理などの所与の信号処理が施される。そして、信号処理が施されたデータが、CPU530又はDMAC538によりデータ転送制御装置500に送られる。データ転送制御装置500は、このパラレルのデータをシリアルデータに変換し、USBを介して他の機器に送信する。

**【0208】**

一方、USBを介して他の機器から送られてきたシリアルのデータは、データ転送制御装置500によりパラレルのデータに変換される。そして、このパラレルデータは、CPU530又はDMAC538により信号処理部534に送られる。そして、信号処理部534においてこのパラレルデータに対して所与の信号処理が施され、読み取り&書き込み部533によりCD-RW532に記憶される。30

**【0209】**

なお、図25(A)、(B)、(C)において、CPU510、520、530の他に、データ転送制御装置500でのデータ転送制御のためのCPUを別に設けるようにしてもよい。

**【0210】**

本実施形態のデータ転送制御装置を電子機器に用いれば、OTG機能を有する電子機器を実現できる。即ち、電子機器にホストとしての役割を持たせたり、デバイスとしての役割を持たせることが可能になり、これまでに存在しなかったアプリケーションを創出できる。40

**【0211】**

また本実施形態のデータ転送制御装置を電子機器に用いれば、電子機器に組み込まれるCPU(処理部)の処理負荷が軽減され、安価なCPUを用いることが可能になる。また、CPUが、データ転送制御処理以外の他の処理を余裕を持って行うことが可能になり、電子機器の性能向上や低コスト化を図れる。また、CPU上で動作するファームウェアのプログラムを簡素化でき、電子機器の開発期間の短縮化を図れる。

**【0212】**

なお本実施形態のデータ転送制御装置を適用できる電子機器としては、上記以外にも例え50

ば、種々の光ディスクドライブ（CD-ROM、DVD）、光磁気ディスクドライブ（MO）、ハードディスクドライブ、デジタルビデオカメラ、携帯電話、スキヤナ、TV、VTR、オーディオ機器、電話機、プロジェクタ、パーソナルコンピュータ、電子手帳、或いはワードプロセッサなど種々のものを考えることができる。

【0213】

なお、本発明は本実施形態に限定されず、本発明の要旨の範囲内で種々の変形実施が可能である。

【0214】

例えば、本発明のデータ転送制御装置の構成は、図5等で説明した構成に限定されるものではなく、種々の変形実施が可能である。

10

【0215】

また、データ転送制御装置の各ブロック（HC、PC、OTGC等）の構成も、本実施形態で説明したものに限定されず、種々の変形実施が可能である。

【0216】

また、転送条件レジスタに設定される転送条件情報も、本実施形態で説明した情報に限定されるものではない。

【0217】

また、明細書中の記載において広義な用語（ストートコントローラ、処理部、転送コントローラ、バス、転送比率情報、バッファ領域等）として引用された用語（OTGコントローラ、CPU・ファームウェア、ホストコントローラ・ペリフェラルコントローラ、USB、連続実行回数、パイプ領域・エンドポイント領域等）は、明細書中の他の記載においても広義な用語に置き換えることができる。

20

【0218】

また、本発明のうち従属請求項に係る発明においては、従属先の請求項の構成要件の一部を省略する構成とすることもできる。また、本発明の1の独立請求項に係る発明の要部を、他の独立請求項に従属させることもできる。

【0219】

また、本実施形態ではUSBのOTG規格への適用例を説明したが、本発明が適用されるのはOTG規格に限定されない。例えばOTG規格と同様の思想に基づく規格やOTG規格を発展させた規格におけるデータ転送にも本発明は適用できる。

30

【図面の簡単な説明】

【図1】 図1(A)、(B)、(C)は、USBのOTG規格について説明するための図である。

【図2】 図2(A)、(B)は、SRPやHNPの手順について説明するための図である。

【図3】 図3(A)、(B)は、OHCIのリスト構造のディスクリプタなどについて説明するための図である。

【図4】 OHCIの2分木構造のディスクリプタについて説明するための図である。

【図5】 本実施形態のデータ転送制御装置の構成例を示す図である。

【図6】 図6(A)、(B)は、パイプ領域、エンドポイント領域について説明するための図である。

40

【図7】 データ転送制御装置のホスト時の動作について説明するための図である。

【図8】 データ転送制御装置のペリフェラル時の動作について説明するための図である。

【図9】 レジスタ部について説明するための図である。

【図10】 汎用転送条件レジスタのレジスタマップの詳細例である。

【図11】 コントロール転送用転送条件レジスタのレジスタマップの詳細例である。

【図12】 図12(A)、(B)は、転送条件レジスタの各ビットフィールドに設定される転送条件情報の概要について示す図である。

【図13】 転送条件レジスタの各ビットフィールドに設定される転送条件情報の概要に

50

について示す図である。

【図14】 フームウェアの処理例を説明するためのフローチャートである。

【図15】 INトランザクションでの自動トランザクション処理の信号波形例である。

【図16】 OUTトランザクションでの自動トランザクション処理の信号波形例である。

【図17】 図17(A)、(B)は、トランザクションの連続実行回数を設定する手法について説明するための図である。

【図18】 図18(A)、(B)、(C)は、インターラプト転送のトークン発行周期を設定する手法について説明するための図である。

【図19】 OTGコントローラの詳細な構成例を示す図である。 10

【図20】 図20(A)、(B)は、ホストコントローラ、ペリフェラルコントローラの詳細な構成例を示す図である。

【図21】 バッファコントローラの詳細な構成例を示す図である。

【図22】 図22(A)、(B)、(C)は、領域確保手法やポイント割り当て手法について説明するための図である。

【図23】 ホスト動作時におけるファームウェアの詳細な処理例を示すフローチャートである。

【図24】 ペリフェラル動作時におけるファームウェアの詳細な処理例を示すフローチャートである。

【図25】 図25(A)、(B)、(C)は、種々の電子機器の内部ブロック図の例である。 20

【図26】 図26(A)、(B)、(C)は、種々の電子機器の外観図の例である。

#### 【符号の説明】

P I P E 0 ~ P I P E e パイプ(バッファ)領域、

E P 0 ~ E P e エンドポイント(バッファ)領域、

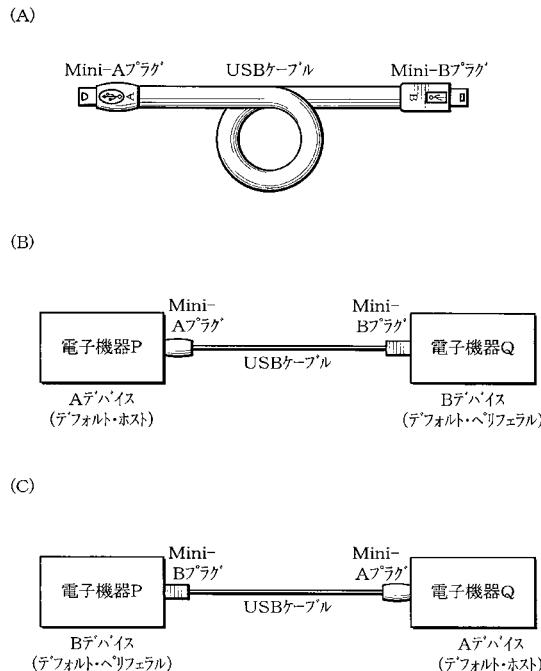
T R E G 0 ~ T R E G e 転送条件レジスタ(共用レジスタ)、

1 0 トランシーバ、 1 2 物理層回路、 2 0 OTGコントローラ(ステートコントローラ)、 3 0 H C / P C 切り替え回路、 3 2 H C / P C セレクタ、 3 4 ラインステートコントローラ、 4 0 転送コントローラ、

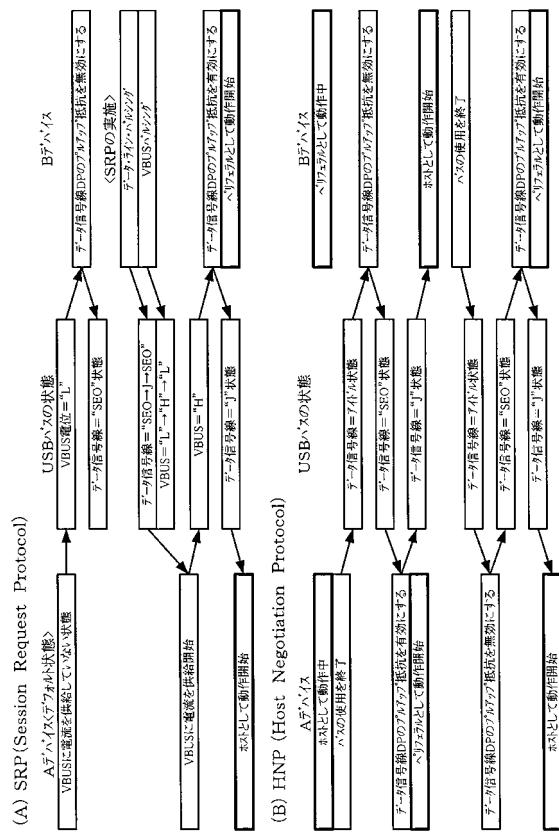
5 0 ホストコントローラ、 6 0 ペリフェラルコントローラ、 7 0 レジスタ部、 30  
7 2 転送条件レジスタ部(共用レジスタ)、 8 0 バッファコントローラ、 1 0

0 パケットバッファ(FIFO、RAM)、 1 1 0 インターフェース回路、 1 1  
2 DMAハンドラ回路、 1 1 4 CPUインターフェース回路、 1 2 0 クロックコントローラ

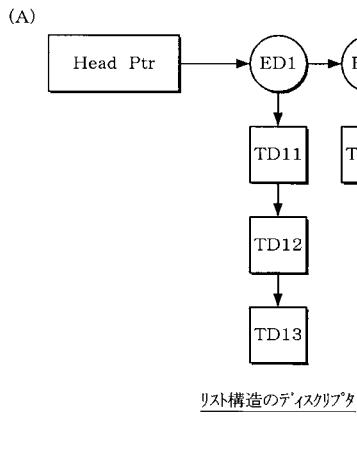
【 図 1 】



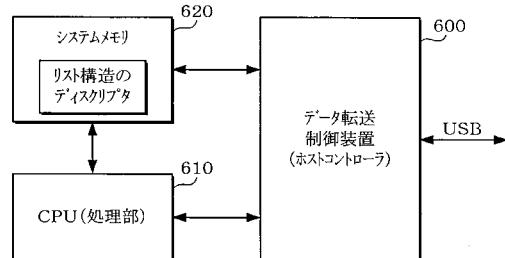
【 四 2 】



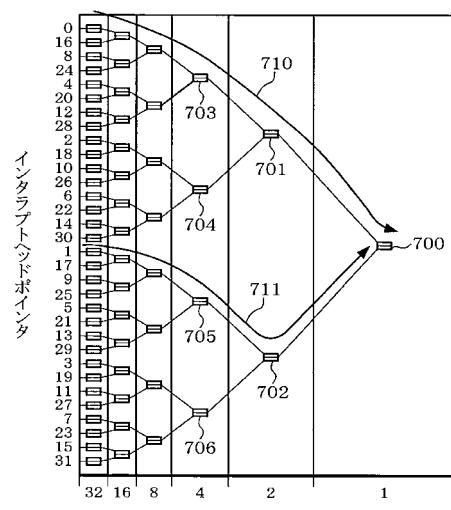
【 図 3 】



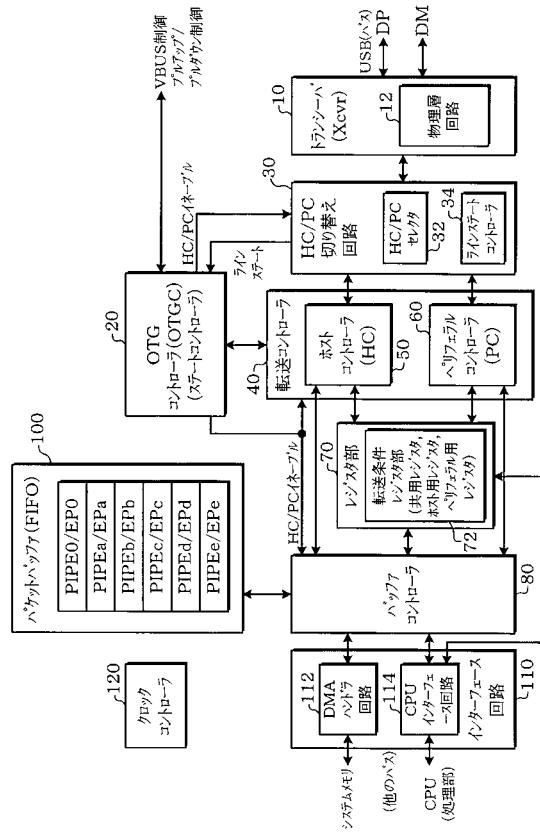
(B) ホスト(パソコン等)



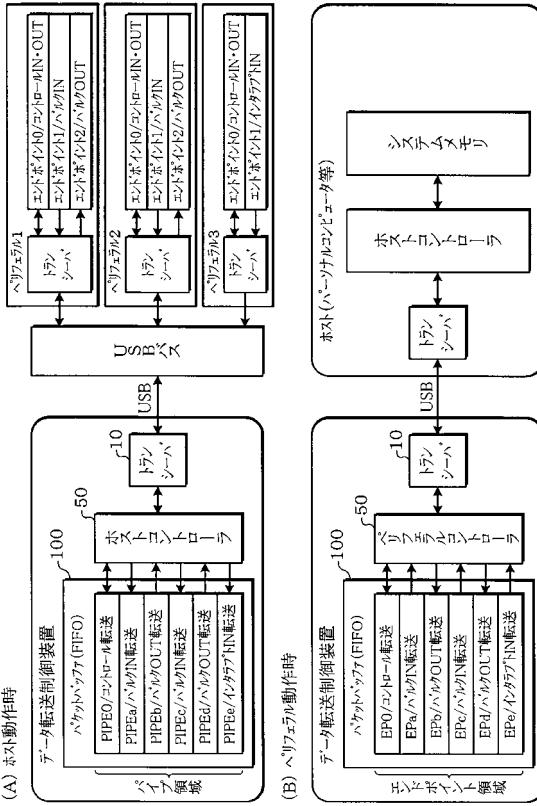
【 义 4 】



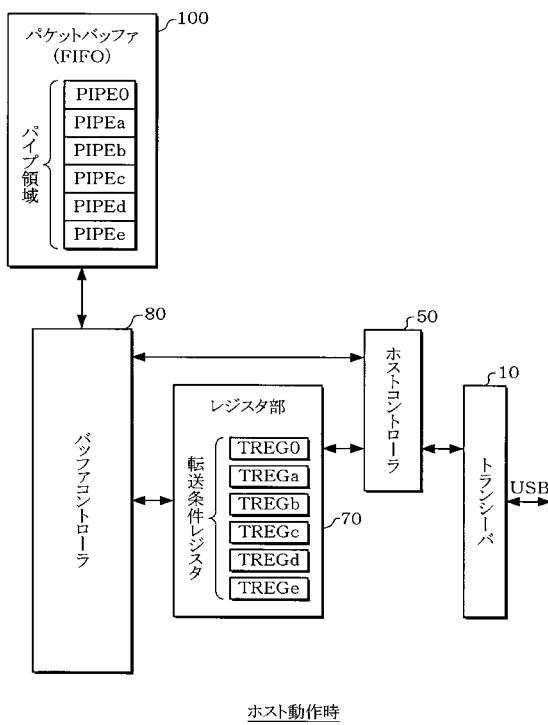
【図5】



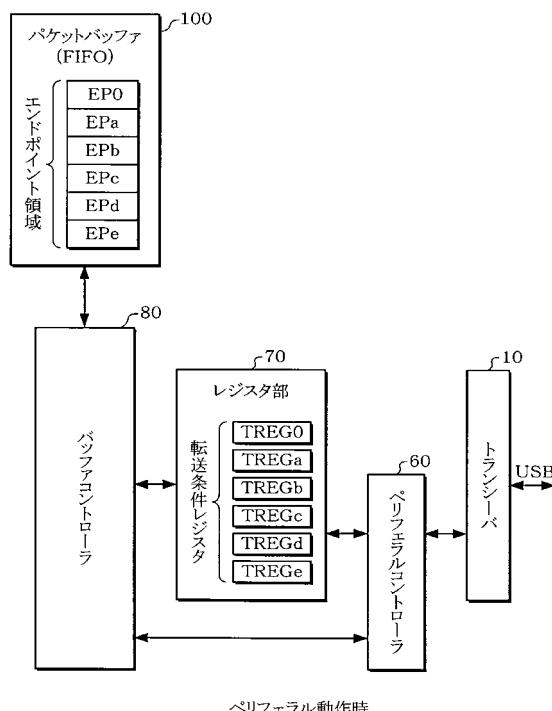
【図6】



【図7】



【図8】



【図9】

転送条件用レジスタ (PIPE1用制御レジスタ) (O,a-e)	HC/PC(PIPE/EP) 共用レジスタ	ホスト動作時PIPEと ヘリカル動作時PIPEで 共用されるレジスタ	データの転送方向/転送種別、 エンドポイント番号、マスク、ケットサイズ、 ページ数、DMA接続の有無等
	HC(PIPE)用レジスタ	ホスト動作時PIPEにのみ 使用されるレジスタ	ヘリカル動作時PIPEと ヘリカル動作時PIPEにのみ 使用されるレジスタ
	PC(EP)用レジスタ	ヘリカル動作時PIPEに のみ使用されるレジスタ	エンドポイントリスト、 ヘッドジャギング指定等
	バクトバク(PIPE)用 共用アセサブルレジスタ	バクトバク(PIPE)用の 制御を行なうためのレジスタ	バクトバク(PIPE)用ヘッダ/ヘンダブリーフ、 ヘッダ/ヘンダブリーフ等
シンクアラート用レジスタ	ヘリカル+スケーリング ドロップ用レジスタ	郵便込みのステータス(原因)を ドロップ	OTGC系、HC(Pipe)系、PC(EP)系等
	ヘリカル+ノーブルンダ ドロップ用レジスタ	ドロップのみのイネーブル、 ディエアーカーを設定するレジスタ	OTGC系、HC(Pipe)系、PC(EP)系等
アロカク系レジスタ	アロカク間用レジスタ アロカク用レジスタ	アロカク間で共用されるレジスタ 各フレグ内で共用されるレジスタ	アロカクのセカンド指示等 Xor前削除、OTGスヌーズ、トライプ、 HCRスタートマーク、トライプ、ヘンダ等
DMA制御レジスタ	DMA転送に関する 設定を行なうレジスタ	DMA転送に関する 設定を行なうレジスタ	DMA転送開始指示、 DMA転送データのヘッダ等

【 図 1 1 】

【図10】

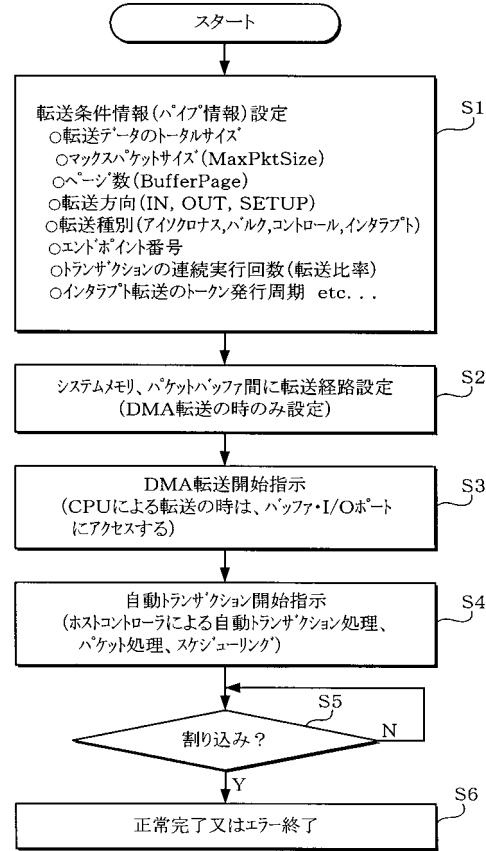
【 図 1 2 】

(A) レジスター		概要	
JoinPPI	PIPPortの接続の有無。	DMA/S總線端子のクリア。(ポートを元に戻す)	
FIFOClr	1バッファ(PIPE EP)領域のデータシーケンスモードを設定。	1バッファ(PIPE EP)領域のデータシーケンスモードを設定。	
ToggleMode	1バッファ(PIPE EP)領域のデータシーケンスモードを設定。	1バッファ(PIPE EP)領域のデータシーケンスモードを設定。	
AutoZeroLen	1バッファ(PIPE EP)領域のデータシーケンスモードを設定。	1バッファ(PIPE EP)領域のデータシーケンスモードを設定。	
BuffToPage	バッファ(PIPE EP)領域のページ数。	バッファ(PIPE EP)領域のページ数。	
DirPd	転送方向(レクシヨン)選別(アインクロス、ハリク、コントロール、インタラクト)。	転送方向(レクシヨン)選別(アインクロス、ハリク、コントロール、インタラクト)。	
TransType	エンティティ番号。	エンティティ番号。	
EPNumber	マックスシケンササイズ。	マックスシケンササイズ。	
xConfig_0	トランザクションの遷移実行回数。	トランザクションの遷移実行回数。	
xConfig_1	トランザクションの遷移実行回数。	トランザクションの遷移実行回数。	
MaxPktSize_H	トランザクションの遷移実行回数。	トランザクションの遷移実行回数。	
xMaxPktSize_L	トランザクションの遷移実行回数。	トランザクションの遷移実行回数。	
PIPEAntival	Continuity	トランザクションの遷移実行回数。	
PIPEControl	Toogle	トランザクションの遷移実行回数。	
PIPEFuncAddr	Trans	トランザクションの遷移実行回数。	
PIPETotalSize_H	ToGo	トランザクションの遷移実行回数。	
PIPETotalSize_M		トランザクションの遷移実行回数。	
PIPETotalSize_L		トランザクションの遷移実行回数。	
xPIPEFuncCPU_H		トランザクションの遷移実行回数。	
xPIPEFuncCPU_L		トランザクションの遷移実行回数。	
xPIPEFuncCPU_M		トランザクションの遷移実行回数。	
xPIPEFuncCPU_S		トランザクションの遷移実行回数。	
xFIFOEmpty	バッファ(PIPE EP)領域のEmitを示す。	バッファ(PIPE EP)領域のEmitを示す。	
xFIFOFull	バッファ(PIPE EP)領域のFillを示す。	バッファ(PIPE EP)領域のFillを示す。	
xFIFOFlow	上記2つの最も近いフレーム。	上記2つの最も近いフレーム。	
xFIFOByteAccess	番號ハイバーデータ上記2つでワードアクセスする場合に、最終データbyteのアクセスを有効とする。	番號ハイバーデータ上記2つでワードアクセスする場合に、最終データbyteのアクセスを有効とする。	
xFIFODataRemain	バッファ(PIPE EP)領域のデータ残量。	バッファ(PIPE EP)領域のデータ残量。	
xFIFOControl_1			
(B) レジスター		概要	
DataStageDr	データステージがある場合に、その転送方向を指定。	データステージの有無を指定。	
PEPCTLAutoNode	データステージの有無を指定。	データステージの有無を指定。	
ERCTLAuto	コンソール転送自動モードのON/OFF。	コンソール転送自動モードのON/OFF。	

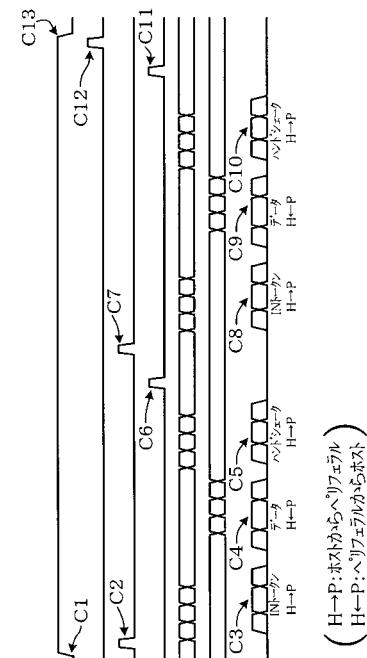
【図13】



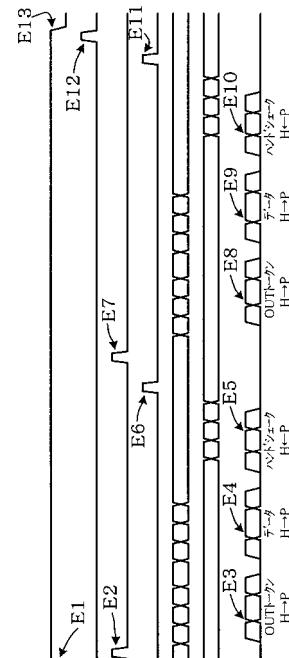
【図14】



【 図 15 】



【 図 1 6 】



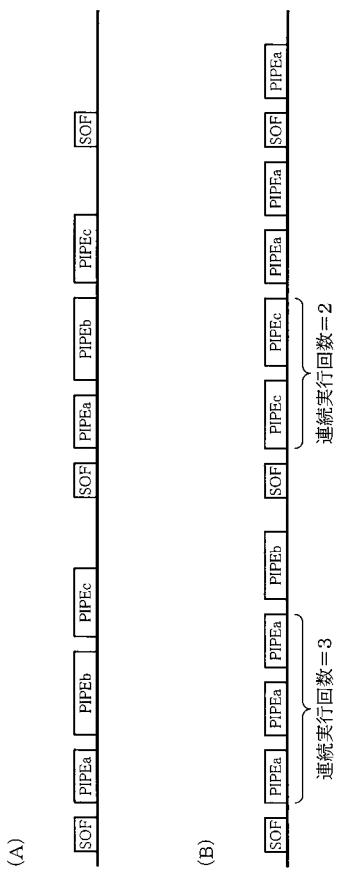
N トランザクション

UNITトランザクション

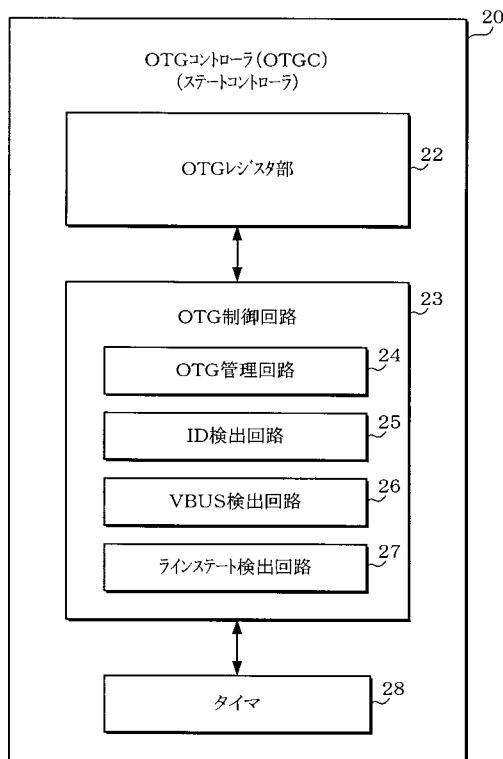
```

sequenceDiagram
    participant TG as TranGo
    participant TC as TranComp
    TG->>TC: Init
    TG-->>TG: TrnCmp
    TC-->>TG: TrnCmpACK
    TG-->>TG: TrnCmpNACK
    TC-->>TG: TrnCmpNACK
    TG-->>TG: TrnCmp
    
```

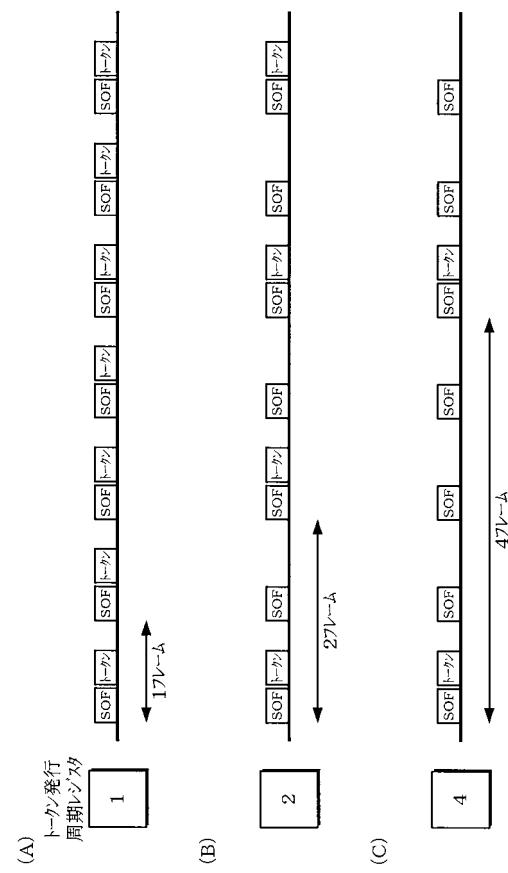
【図 17】



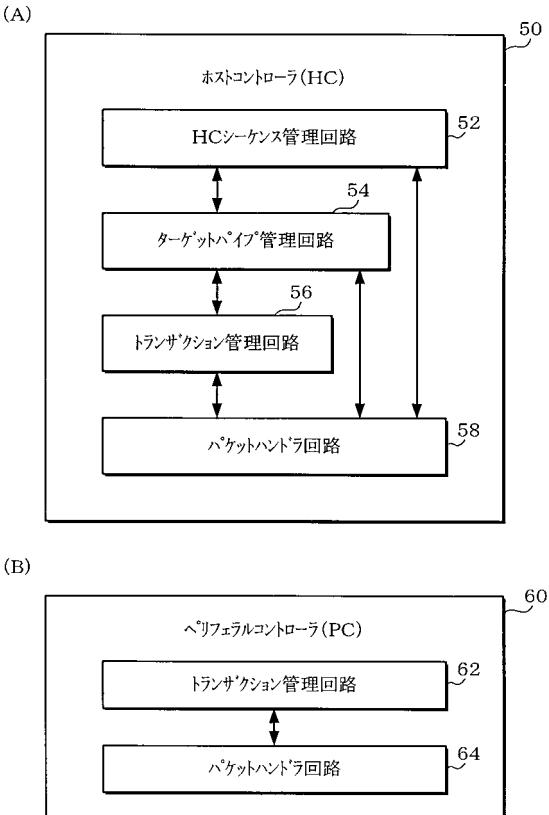
【図19】



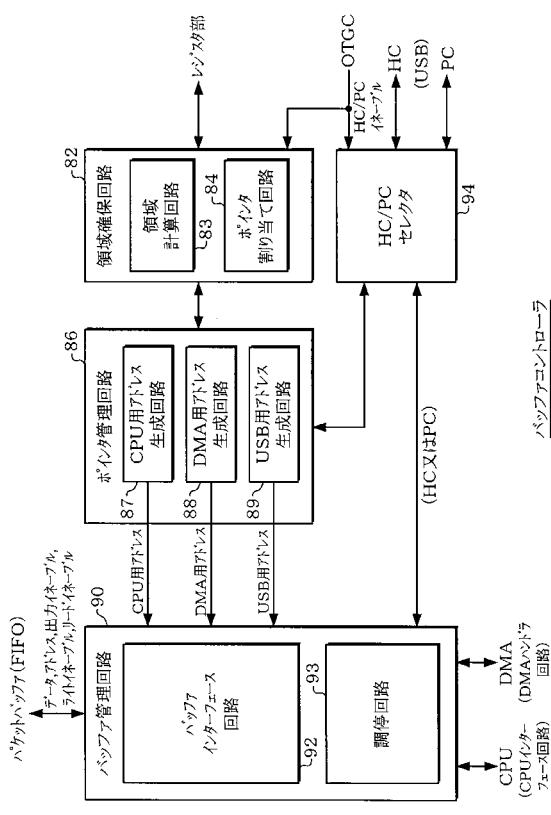
【 図 1 8 】



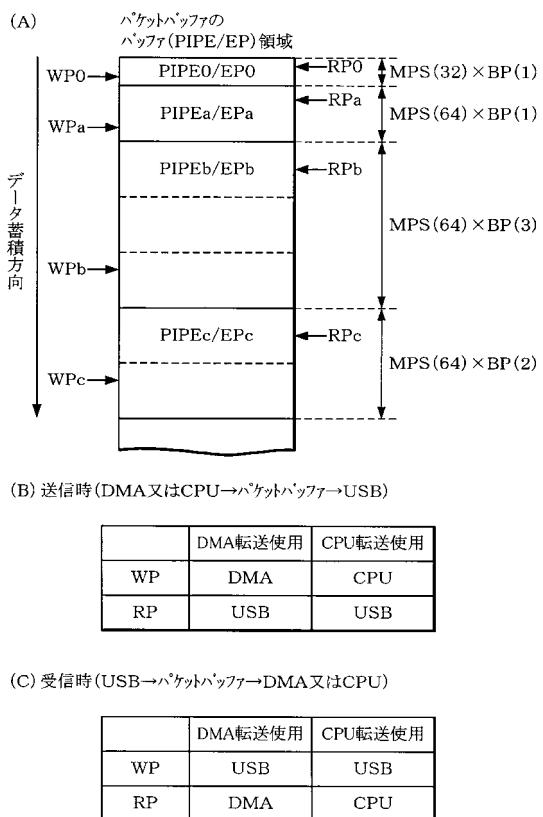
【図20】



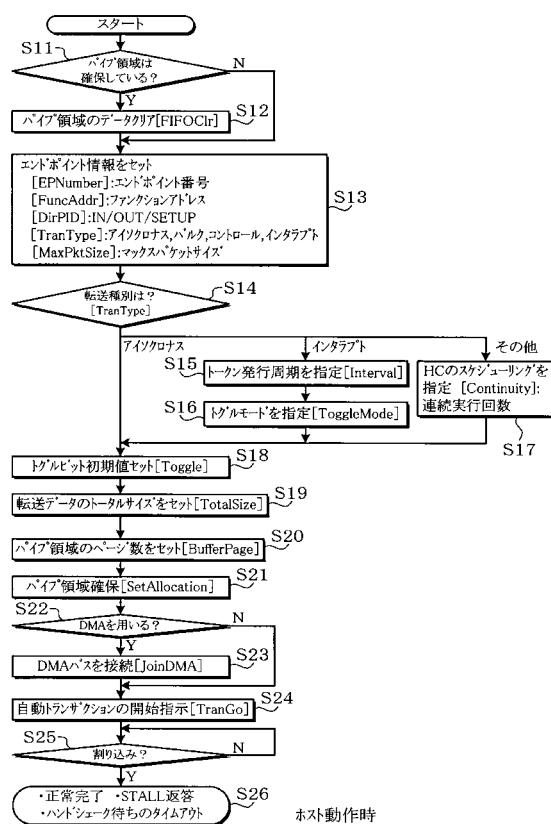
【図21】



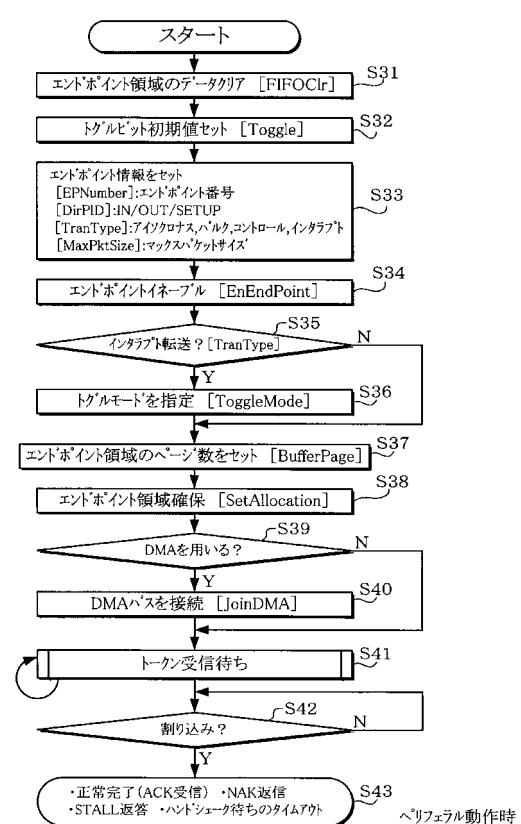
【図22】



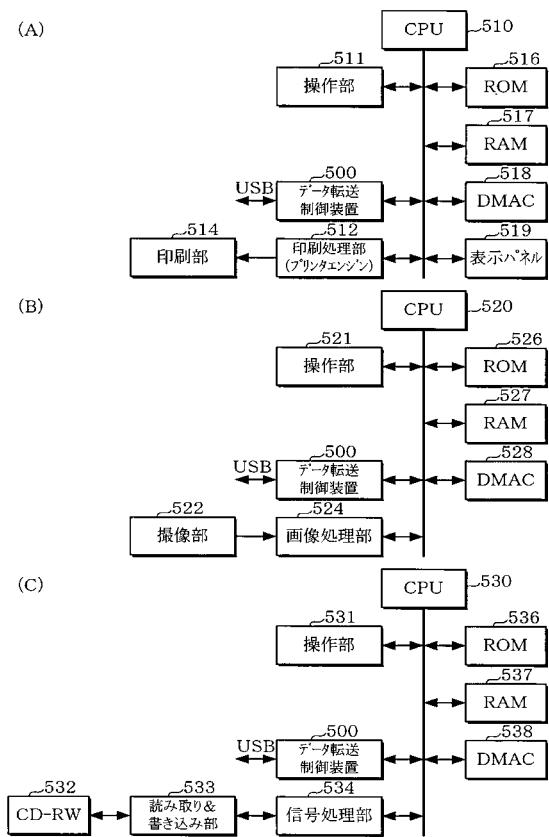
【図23】



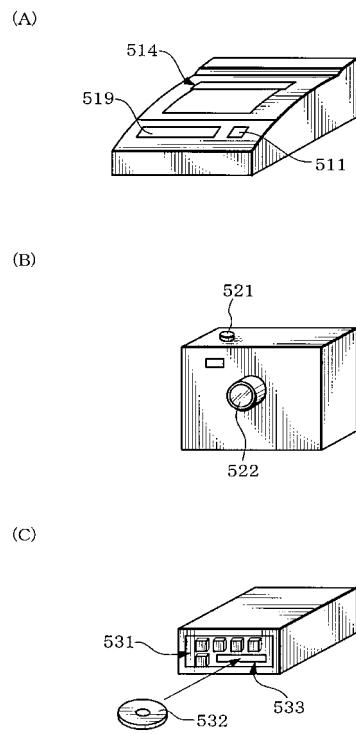
【図24】



【図25】



【図26】



---

フロントページの続き

(72)発明者 下野 洋昭  
長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内

(72)発明者 松田 邦昭  
長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内

審査官 石井 茂和

(56)参考文献 特開2002-055936(JP,A)  
特開2002-091717(JP,A)  
特開2002-009849(JP,A)  
特開2002-116998(JP,A)  
特開平10-301898(JP,A)  
米国特許第6105097(US,A)

(58)調査した分野(Int.Cl., DB名)

G06F 13/38  
WPI(DIALOG)  
JSTPlus(JDreamII)