

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2005-504392

(P2005-504392A)

(43) 公表日 平成17年2月10日(2005.2.10)

(51) Int. Cl. ⁷	F I	テーマコード (参考)
G06F 13/38	G06F 13/38 350	5B077
H04L 12/40	G06F 13/38 320Z	5K032
	H04L 12/40 Z	

審査請求 未請求 予備審査請求 未請求 (全 45 頁)

(21) 出願番号 特願2003-533133 (P2003-533133)
 (86) (22) 出願日 平成14年9月16日 (2002. 9. 16)
 (85) 翻訳文提出日 平成16年3月23日 (2004. 3. 23)
 (86) 国際出願番号 PCT/IB2002/003812
 (87) 国際公開番号 W02003/029996
 (87) 国際公開日 平成15年4月10日 (2003. 4. 10)
 (31) 優先権主張番号 PCT/SG01/00194
 (32) 優先日 平成13年9月27日 (2001. 9. 27)
 (33) 優先権主張国 シンガポール (SG)
 (81) 指定国 EP (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), CN, JP, KR

(71) 出願人 590000248
 コーニンクレッカ フィリップス エレクトロニクス エヌ ヴィ
 Koninklijke Philips Electronics N. V.
 オランダ国 5621 ペーアー アインドーフェン フルーネヴァウツウェeg 1
 Groenewoudseweg 1, 5621 BA Eindhoven, The Netherlands
 (74) 代理人 100092048
 弁理士 沢田 雅男

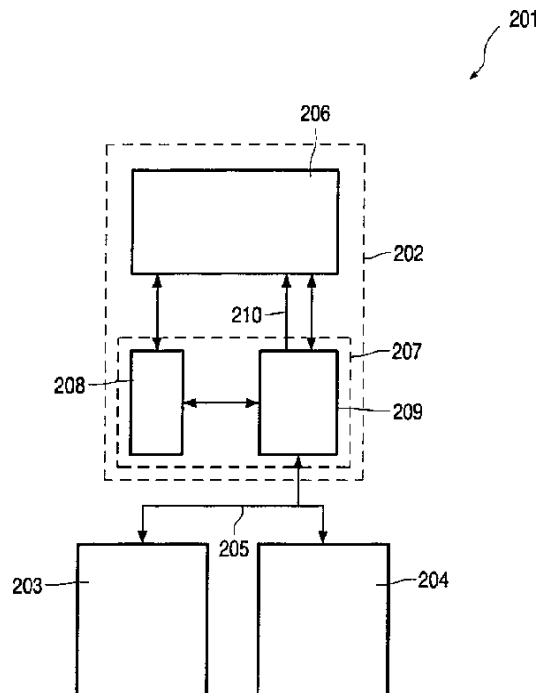
最終頁に続く

(54) 【発明の名称】 バス・システムおよびバス・インターフェース

(57) 【要約】

【課題】 バス・システムの第1の端末のプロセッサから、バス・システムの第2の端末まで、またはその逆で、データを輸送するために必要なオーバーヘッドの量を減らすこと。

【解決手段】 本発明は、データおよび制御信号を転送するためのバスによって結合されている第1の端末 (202) と第2の端末 (203、204) とを備えているバス・システムに関する。前記バスは、当該第1の端末 (202) が当該第2の端末 (203、204) に繰り返しリクエストを送信するプロトコルによって作動すると着想されている。第1の端末 (202) は、プロセッサ (206) と、バス・インターフェース (207) と、当該バス・インターフェースおよび当該インタラプト可能なプロセッサに結合されるバッファ (208) とを含み、前記プロセッサ (206) は、前記第1の端末 (202) によって送信されるリクエストに対するリクエスト特性を生成するように動作可能である。さらに、プロセッサ (206) は、当該データ項目を処理するために動作可能である。バッファ (208) は、当該リクエスト特性を格納するために動作可能である。バス



【特許請求の範囲】**【請求項 1】**

データおよび制御信号を転送するためのバスによって結合されている第1の端末と第2の端末とを備え、当該バスが、当該第1の端末が当該第2の端末に繰り返しリクエストを送信すると考えられているプロトコルによって作動するバス・システムにおいて、当該第1の端末は、プロセッサと、バス・インターフェースと、当該プロセッサ及び当該バス・インターフェースに結合されるバッファを備え、

当該プロセッサは、当該リクエスト及び当該データ項目を処理するためのリクエスト特性を生成するために動作可能であり、

当該バッファは、当該リクエスト特性を格納するために動作可能であり、

10

当該バス・インターフェースは、当該格納されているリクエスト特性から当該リクエストを生成し、かつ当該リクエストを繰り返し送信するために動作可能であることを特徴とするバス・システム。

【請求項 2】

当該リクエスト特性は、当該リクエストのポーリング・レートを備えることを特徴とする請求項1に記載のバス・システム。

【請求項 3】

当該リクエスト特性は、さらに当該リクエストのペイロードを備えることを特徴とする請求項2に記載のバス・システム。

【請求項 4】

20

当該第2の端末は、当該リクエストに回答し、当該プロセッサは、インタラプト可能なプロセッサであり、当該バス・インターフェースは、当該第2の端末の選択された回答の受信によって、当該プロセッサにインタラプトするために動作可能であることを特徴とする請求項1に記載のバス・システム。

【請求項 5】

当該バス・システムは、USBシステムであることを特徴とする請求項1に記載のバス・システム。

【請求項 6】

バスのための接続と、プロセッサからリクエスト特性を受信するための入力とを備えるバス・インターフェースにおいて、当該リクエスト特性を格納するために動作可能なバッファと、前記格納されているリクエスト特性からのリクエストを生成し、かつ前記接続に当該リクエストを繰り返し送信するために動作可能なコントローラとを備えることを特徴とするバス・インターフェース。

30

【請求項 7】

当該リクエスト特性は、当該リクエストのポーリング・レートを備えることを特徴とする請求項6に記載のバス・インターフェース。

【請求項 8】

当該リクエスト特性は、さらに当該リクエストのペイロードを備えることを特徴とする請求項7に記載のバス・インターフェース

【請求項 9】

40

当該バス・インターフェースは、さらにインタラプト可能なプロセッサにインタラプト信号を送信するためのインタラプト出力を備え、かつ当該コントローラは、当該接続からの回答を受信し、かつインタラプト出力にインタラプト信号を送信することによって選択された回答の受信によって当該プロセッサをインタラプトするために動作可能であることを特徴とする請求項6に記載のバス・インターフェース。

【請求項 10】

当該バス・インターフェースは、USBインターフェースとして動作するように動作可能であることを特徴とする請求項6に記載のバス・インターフェース。

【発明の詳細な説明】**【技術分野】**

50

【0001】

本発明は、データおよび制御信号を転送するためのバスによって結合されている第1の端末と第2の端末とを備え、当該バスが、当該第1の端末が当該第2の端末に繰り返しリクエストを送信すると考えられているプロトコルによって作動するバス・システムに関する。

【0002】

また、本発明は、バスのための接続と、プロセッサからリクエスト特性を受信するための入力とを備えるバス・インターフェースに関する。

【背景技術】

【0003】

このようなバス・システムは、USB規格に開示されている。USB規格は、コンパックコンピュータ社、ヒューレット・パカード社、インテル社、ルーセント・テクノロジー社、マイクロソフト社、日本電気株式会社、およびコーニンクレッカ フィリップス エレクトロニクス エヌ ヴィ社を含む一群の会社によって提案された標準規格である。本発明の完全な理解に適切なUSBの各種の態様については後述する。USBに関する更なる背景は、参照文献として本明細書に含まれる、USB規格（リビジョン2.0）から得ることができる。

10

【0004】

USBは、パーソナルコンピュータと多種多様な周辺機器（例えば、ウェブカメラ、プリンタ、スキャナ、マイクおよびキーボード）との間のデータ交換をサポートするように設計されている標準バスである。従って、USBは、RS-232Cシリアルポート、パラレル・ポートおよびPS/2インターフェースなどの従来のインターフェースに取って代わるものである。例えば、USB On-The-Goに記載されているように、USBの更なる開発は、他の装置をパーソナルコンピュータに置き換えるバス・システムを照準として定めている。例えば、このような他の装置には、デジタル・スチルカメラ、ビデオカメラ、パーソナル携帯情報機器または携帯電話機を用いることができるであろう。

20

【0005】

USBシステムのようなバス・システムの場合、電子性バス・ステーションは、バスを介して互いに接続されている。通常、バス・ステーションは、データを処理または生成するため、およびデータのフロー制御を行うために、プロセッサまたは演算システムを備えている。USB端末は、ホスト（例えば、パーソナルコンピュータ）およびデバイス（周辺機器）として規定されている。一般に、いかなるUSBシステムにおいても、ホストは一つしか存在しない。ホストのUSBインターフェースは、ホスト・コントローラとも呼ばれる。ホスト・コントローラは、ハードウェア、ファームウェアまたはソフトウェアの組合せで、実施することができる。USBデバイスは、追加的な取付点を提供するハブ、またはシステムに可能出力を提供するファンクションとして規定されている。これらのデバイスは、階層的なスター・トポロジーでホストに接続されている。ハブは、USBデバイスが取付けられるまたは取り外されることを示す。ホストは、新しく取り付けられたUSBデバイスが、ハブであるかファンクションであるかを決定し、かつそのUSBデバイスに固有のUSBアドレスを割り当てる。全てのUSBデバイスは、固有のUSBアドレスによってアクセスされる。加えて、各々のデバイスは、ホストが通信できる単数または複数のエンドポイントをサポートする。以下の説明は、ファンクションとして規定されるUSBデバイスに限るものである。

30

40

【0006】

USBは、ポーリングされるバスである。ホスト・コントローラにより、あらゆるデータ転送が開始される。大部分のバス・トランザクションには、3つのパケットが関係する。ホスト・コントローラが、スケジューリングに基づいて、トランザクションのタイプおよび指示と、USBデバイスアドレスと、エンドポイント番号とが記述されているUSBパケットを送信することによって、各々のトランザクションが開始される。このパケットは、「トークン・パケット」と呼ばれる。アドレス指定されたUSBデバイスは、適正なアドレス・フィールドを復号することによってそれ自体を選択する。所定のトランザクションでは、ホストからデバイスまで、またはデバイスからホストまでのいずれかで、データ転送が行わ

50

れる。データ転送の方向は、トークン・パケットにおいて指定されている。このとき、トランザクションの転送元は、データ・パケットを送信する、または転送するデータがないことを示す。一般に、転送先は、データ転送が成功したか否かを示すハンドシェイク・パケットによって応答する。

【0007】

USBアーキテクチャには、次の3つの基本タイプのデータ転送が包含される。(1)アイソクロナス転送：ホストとデバイスとの間の周期的で、連続的な通信で、通常、予めネゴシエートされた待ち時間でUSBバンド幅の予めネゴシエートされた量を占有する時間関連情報に用いられる。(2)インタラプト転送：低周波数で、通信待ち時間が固定されている通信で、通常、人間の認知可能なエコーまたはフィードバック応答特性を有する文字または座標などのアシンクロナス・データに用いられる。(3)バルク転送：非周期性で、大きいパケットのバースト通信で、通常、いかなる有効なバンド域幅も用いることができ、かつバンド域幅が有効になるまで遅延させることが出来るデータに用いられる。

【0008】

バス・ステーションの双方のタイプ、つまり、ホストとデバイスは、通常、データの処理または生成のいずれかを行うため、および/またはデータのフロー制御を行うためのプロセッサまたはコンピューティングシステムを備えている。上述したUSBシステムの簡単な実施は、ホストとして機能するバス・ステーションのプロセッサと、デバイスとして機能するバス・ステーションのプロセッサとの双方にかなりのオーバーヘッドをかけることになる。このことは、特に、インタラプト転送の場合に言える。バス・システムがポーリングを行なうために、ホストは定期的に関係するデバイス(例えば、キーボード)に問い合わせを行うことが必要となる。

【発明の開示】

【課題を解決するための手段】

【0009】

本発明の目的は、とりわけ、バス・システムの第1の端末のプロセッサから、バス・システムの第2の端末まで、またはその逆で、データを輸送するために必要なオーバーヘッドの量を減らすことである。

【0010】

このために、本発明は、冒頭のパラグラフにおいて規定されたバス・システムにおいて、当該第1の端末は、プロセッサと、バス・インターフェースと、当該インタラプト可能なプロセッサ及び当該バス・インターフェースに結合されるバッファを含み、当該プロセッサは、当該リクエスト及び当該データ項目を取り扱うためのリクエスト特性を生成するために動作可能であり、当該バッファは、当該リクエスト特性を格納するために動作可能であり、当該バス・インターフェースは、当該格納されているリクエスト特性から当該リクエストを生成し、かつ当該リクエストを繰り返し送信するために動作可能であることを特徴とするバス・システムを提供する。

【0011】

作動中、第1の端末内のプロセッサは、第2の端末に情報を提供すること、または第2の端末から情報を得ることが必要となる。このために、第1の端末は、第2の端末と通信することが必要となる。USBシステムのようなポーリングされたバス・システムでは、USBシステムのホストとなる第1の端末によって、通信が開始される。USBシステムの一デバイスである第2の端末が、不規則にデータを受信または提供することしか出来ない場合には、情報の転送が成功する前に、第1の端末は第2の端末に何度もポーリングを行うことが必要となる事態がしばしば生じる。換言すれば、これらの状況の下では、第1の端末は、情報の転送が成功する前に、相対的に頻繁に、同じリクエストを第2の端末へ再送することが必要となる。

【0012】

従来処理法では、当該プロセッサが、送信が必要である当該リクエストを毎回準備する

10

20

30

40

50

ことが必要になる。その後、当該バス・インターフェースが、当該バスを介して当該第2の端末に、実際に、当該リクエストを送信する。本発明のバス・システムでは、プロセッサが、リクエストの特性を生成または決定する。バス・インターフェースの可能出力を強化することによって、当該バス・インターフェースは、実際に、プロセッサによって提供される当該特性からリクエストを生成し、かつ要求される限り何回でも、当該第2の端末にリクエストを送信することができる。

【0013】

さらに、本発明のバス・システムは、プロセッサによって提供されるリクエスト特性を格納するために用いられる第1の端末内にバッファを備えている。バッファは、プロセッサとバス・インターフェースとの双方に対してアクセス可能であることが好ましい。このようにして、プロセッサの通信オーバーヘッドは、減らすことができる。

10

【0014】

本発明のバス・システムの実施例は、当該リクエスト特性に、当該リクエストのポーリング・レートと、規定の場合の当該リクエストのペイロードとを備えていることを特徴とする。リクエスト特性にポーリング・レートを含むことにより、より多くのことに適応することが可能となる。プロセッサは、バス・インターフェースのポーリング・レートを調整するために用いることができる第2の端末の特徴的な情報を有していてもよい。たとえば、USBシステムは、フレームと称される1ミリ秒またはマイクロフレームと称される125マイクロ秒のいずれかの時間基準を確立する。ポーリング・レート情報に、リクエスト特性が含まれない場合、バス・インターフェースは、おそらく、(マイクロ)フレームごとに一度、最大ポーリング・レートでリクエストを送信するという、固定されたポーリング・レートをを用いるであろう。リクエスト特性には、ポーリング・レート情報が含まれているので、バス・インターフェースは、可能な限りいつでも低いポーリング・レートをを用い、要求されたときに高いポーリング・レートをを用いることによって、ポーリング・レートを、必要に応じて調整する。これにより、バス負荷、および第2の端末に対する通信オーバーヘッドが、減少する。

20

【0015】

第1の端末が、データを受信するために、第2の端末に対してリクエストを送信する場合、そのリクエストそのものが、必ずしもペイロード(すなわち、データ項目)を含むわけではない。プロセッサによって提供されるリクエスト特性内にペイロード情報を含むことは、第1の端末内のプロセッサからの妨害なしに、第1の端末から第2の端末まで、データを繰り返し送信することができるという効果をもたらす。これにより、プロセッサの通信オーバーヘッドが、更に減少する。

30

【0016】

本発明のバス・システムの更なる実施例では、当該第2の端末は、当該リクエストに回答し、当該プロセッサは、インタラプト可能なプロセッサであり、当該バス・インターフェースは、当該第2の端末の選択された応答の受信によって、当該プロセッサにインタラプトするために動作可能であるであることを特徴とする。更なる動作が要求されない場合には、当該プロセッサをインタラプトする必要はない。したがって、通信が予想通りに完了した場合、または第2の端末によって何ら新たな情報・新たなデータ項目が提供されない場合のいずれかでは、第2の端末との通信結果の報告を戻す必要がない。例えば、USBシステムでは、第2の端末は、データ項目肯定応答信号・否定応答信号を有するメッセージを送信すること、信号が停止されていることを示す信号を送信することによって応答することができる。さらに、第2の端末が、信号の送信に対して応答しない場合、または、第1の端末と第2の端末との間の通信のエラーに起因して、第1の端末で信号を受信しない場合には、これを応答と解釈することもできる。この選択された応答には、第2の端末から応答信号が含まれない場合がある。したがって、プロセッサによる更なる動作を要求する応答が受信された場合のみ、プロセッサをインタラプトさせれば、プロセッサの通信オーバーヘッドは減少する。このために、バス・インターフェースは、受信された応答の中から、そのような応答を選択する。このような状況を処理できるように、プロセッサをプログラ

40

50

ムすることによって、プロセッサは、必要に応じて更なる動作を決定することができる。

【0017】

本発明のバス・インターフェースは、バスのための接続と、インタラプト可能なプロセッサからリクエスト特性を受信するための入力とを備え、当該リクエスト特性を格納するために動作可能なバッファと、前記格納されているリクエスト特性からのリクエストを生成し、かつ前記接続に当該リクエストを繰り返し送信するために動作可能なコントローラとを備えることを特徴とする。

【発明を実施するための最良の形態】

【0018】

これらおよびその他の本発明の態様は、以下に記述されている実施例から明らかになり、かつ、以下に記述されている実施例を参照して説明される。各図において、同一の部分は、同一の参照符号によって示される。

10

【0019】

図1は、USB規格のレビジョン2.0において特定されている、インタラプト・トランザクションフォーマットを示すダイアグラムである。USBの中で、全ての通信は、ホストから開始される。全てのUSBデバイスは、固有のUSBアドレスによってアクセスされる。加えて、各々のデバイスは、ホストと通信することができる単数または複数のエンドポイントをサポートしている。

【0020】

USBは、ファンクショナルデータ、およびUSBホストとUSBデバイスとの間の交換制御をサポートする。USBデータ転送は、ホスト・ソフトウェアとUSBデバイス上の特定のエンドポイントとの間で生じる。USBアーキテクチャには、以下の3つ基本タイプの転送が包含される。

20

【0021】

- アイソクロナス転送、つまり、アイソクロナス・データまたは予めネゴシエートされた待ち時間のUSBバンド幅の予めネゴシエートされた量を占有するストリーミングリアルタイム・データの転送。

【0022】

- インタラプト転送、つまり、アシンクロナス双方向データ（例えば人間の認知可能なエコーまたはフィードバック応答特性を有する指標または座標）の転送。

30

【0023】

- バルク転送、つまり、相対的に大きくバースト的な数量で生成または消費され、かつ広く動的な許容範囲および伝送制約を有するアシンクロナス・ブロック転送データの転送。

【0024】

進行中のトランザクションがない場合、USBシステムは、アイドルモード101になる。データ転送を備えているバス・トランザクションでは、通常、セットアップステージ102と、データステージ103と、ステータスステージ104という、3つのステージを備えている。

【0025】

データ転送を含むバス・トランザクションは、通常、3つの異なるパケット、例えば、トークン・パケット105、106、データ・パケット107、108およびハンドシェイク・パケット109、110、111、112、115、116の伝送に関係する。

40

【0026】

インタラプト・トランザクションは、イン転送またはアウト転送から構成することができる。イン転送の場合、USBホストは、入力トークン105を送信することによってトランザクションを開始する。USBデバイスは、入力トークン105を受信すると、データ107、否定応答信号（NAK）111、またはUSBデバイスが命令（STALL）112を完了することができないことを示す信号を戻すことができる。USBデバイスは、USBホストへ戻すための新たなインタラプト情報が、エンドポイントにない場合（すなわち、インタラプトの待機中でない場合）、データステージ103で、NAKハンドシェイク111を戻す。USBデバイスは、命令を完了す

50

ることができない場合には、STALLハンドシェイク112を戻す。USBデバイスは、インタラプトの待機中、データ・パケット107としてインタラプト情報を戻す。USBホストは、データ・パケット107が不正に受信された場合、データ107がエラーなく受信された場合、またはデータ・パケット107がライン113によって示される、ハンドシェイクなしに戻る場合には、データ・パケット107の受け取りに回答して、いずれの場合にも、ACKハンドシェイク109（肯定応答信号）を発行する。さらに、USBホストは、入力トークン105に対するUSBデバイスからの応答が、全く受信できない場合には、ACK109を発行しない。この状況は、ライン114によって示される。イン転送の完了の後、USBシステムは、アイドルモード101に戻る。

【0027】

アウト転送の場合、USBホストは、出力トークン106を送信することによってトランザクションを開始する。次いで、USBホストは、データ・パケット108を送信する。USBデバイスは、データ・パケット108がエラーなく受信された場合、ステータスステージ104で、ACKハンドシェイク110、デバイスがデータを処理することができないことを示すNAKハンドシェイク115、または、USBデバイスがストール中であることを示しているSTALLハンドシェイク116を発行することができる。USBデバイスは、データ・パケット108が不正に受信された場合には、ライン117によって示される、ハンドシェイクなしに戻る。さらに、USBデバイスは、ライン118によって状況が示される、USBホストからデータを受信しない場合に、ACKハンドシェイク110を発行しない。アウト転送の完了の後、USBシステムは、アイドルモードに101に戻る。

【0028】

全てのパケットは、シンクロナス（SYNC）フィールドから開始される。これは入力回路で着信データとローカル・クロックとを同調させるために用いられる。すべてのUSBパケットは、SYNCフィールドの直後に、パケット識別子（PID）がある。PIDは、4ビットのチェックフィールドと、その後の4ビットのIDフィールドとから構成されている。PIDは、パケット（例えば、トークン・パケット105、106、データ・パケット107、108およびハンドシェイク・パケット109、110、111、112、115、116）のタイプ、およびパケットのフォーマットに加えて、そのパケットに適用される変異検出のタイプを示す。PIDの4ビットのチェックフィールドにより、確実に信頼性の高い復号が行える。

【0029】

トークン・パケット105、106は、トークンPIDの値に従って、データ・パケット107、108の転送元または転送先であるUSBアドレスによって、目標USBデバイスを指定するADDRフィールドを備えている。USBデバイスは、ADDRフィールドの完全な復号を実行しなければならない。

【0030】

データ・パケット107、108は、ゼロからNバイトまで変動する場合があります、整数バイトとなるべきデータ・フィールドを備えている。各々のバイト内のデータ・ビットは、最初に最上位ビットからシフトされる。データ・パケットのデータ・フィールドは、リクエストがなされつつあることについてUSBデバイスに付加的な情報を与えるフィールドに分けられる。あるフィールド（命令コード）は、USBデバイス空間からの読み出しのセットアップ、または、USBデバイス空間への書き込みのセットアップを行うなど、ターゲットとなるUSBデバイスが何をすべきかを伝える。読み出し空間命令は、USBデバイスの所定空間の所定位置からの情報をリクエストするために用いられる。この命令は、読み出しのセットアップを行うために用いられる。この後に、データを送信するというUSBデバイスのホストからのリクエストが続く。その後、USBデバイスは、既に指定されている空間から読み出したデータを送信する。書き込み空間は、読み出し空間と全く同じ規定を使用する。しかしながら、データ・パケットには、ホストによってUSBデバイスに転送されるデータが含まれる。

【0031】

ハンドシェイク・パケット109、110、111、112、115、116は、PIDのみから構成されてい

10

20

30

40

50

る。上述したように、ハンドシェイク・パケットは、データ転送のステータスを報告するために用いられ、データの受信が成功したことを示す値、CRCの不成功を示す値、フロー制御を示す値および特定の障害状態を示す値を返すことができる。

【0032】

図2は、本発明のバス・システムを示すブロック図である。バス・システム201は、USBホストとしての役割を果たす第1の端末202と、USBデバイスとしての役割を果たす第2の端末203および204とを備えている。端末202、203、204は、バス接続205によって結合されている。第1の端末202は、更に詳細に示されている。第1の端末202は、インタラプト可能なプロセッサ206と、バス・インターフェース207とを備える。バス・インターフェース207は、バッファ208とコントローラ209とを備える。バッファ208が、バス・インターフェース207の一部であることは、必須ではない。バッファ208を、バス・インターフェース207の外側であって、第1の端末202内に配置して、代替りの装置としてもよい。バッファ208は、プロセッサ206とコントローラ209との双方によってアクセスされる。コントローラ209は、バス接続205に結合されている。さらに、コントローラ209は、プロセッサ206に対するインタラプト接続210を有している。

10

【0033】

図3Aは、データ伝送のためのトランザクションを示すメッセージ・ダイアグラムである。メッセージ・ダイアグラムには、イン転送からなるインタラプト・トランザクションのための図1のシステムの動作を図示している。上から下へ時間が経過している。プロセッサ206とバス・インターフェース207との間の情報交換は、第1のライン301と第2のライン302との間の矢印によって表されている。バス・インターフェースと第2の端末203、204との間の情報交換は、第2のライン302と第3ライン303との間の矢印によって表されている。

20

【0034】

プロセッサ206は、第2の端末（USBデバイス）203、204のうちの1つから、データを要求する。このため、バス・インターフェース207は、データに対するリクエストを、バス接続205に繰り返し発行しなければならない。プロセッサ206に対する通信オーバーヘッドを減らすために、プロセッサ206は、リクエスト特性を一度提供するのみで充分とすべきである。アドレス指定がされたUSBデバイスが、第1の端末へデータを戻す場合、バス・インターフェース207は、インタラプト接続210によってプロセッサ206にインタラプトを行い、プロセッサ206にデータを提供しなければならない。エラーが発生する場合には、プロセッサ206に、通知を行う必要もある。これらの状況では、バス・インターフェース207は、プロセッサ206にも、インタラプトをかけなければならない。他の状況では、バス・インターフェース207は、プロセッサ206にインタラプトすべきでないので、プロセッサ206からの動作は要求されない。

30

【0035】

イン転送304では、プロセッサ206は、メッセージ309によって、リクエスト特性をバス・インターフェース207に送信する。ここで、リクエスト特性は、バッファ208に格納される。バス・インターフェース207は、バッファ208から読み出される、メッセージ309内のリクエスト特性に基づき、バス接続205上に入力トークン310を発行することによってイン転送を開始する。アドレス指定がされたUSBデバイスである第2の端末は、データ項目を有するメッセージ311を送信することによって応答する。バス・インターフェース207は、このメッセージ311の受信で、新たなデータが有効なことをプロセッサ206へ知らせるために、プロセッサ206に対してインタラプト接続210を介して、インタラプト信号312を、送信する。さらに、バス・インターフェース207は、アドレス指定がされたUSBデバイスに、ACK信号313を発行する。これにより、イン転送304が完了する。

40

【0036】

イン転送305では、バス・インターフェース207は、同一のリクエスト特性を有するイン転送304に続き、新たな情報がプロセッサ206から要求されないときに、バッファ208から再びリクエスト特性を読み出す。バス・インターフェース207は、バス接続205上に入力トークン314を発行することによってイン転送を開始する。アドレス指定がされたUSBデバイス

50

は、データ項目を有するメッセージ315を送信することによって応答する。バス接続205上のエラーに起因して、第2のメッセージ315は、バス・インターフェース207によって不正に受信される。したがって、バス・インターフェース207は、ACKハンドシェイクを発行しない。その代わりに、バス・インターフェース207は、エラーが発生したことをプロセッサ206に知らせるために、プロセッサ206に対してインタラプト接続210を介してインタラプト信号316を送信する。これにより、イン転送305が完了する。

【0037】

イン転送306では、バス・インターフェース207は、同一のリクエスト特性を有するイン転送304に続いて、バッファ208から再びリクエスト特性を読み出すが、プロセッサ206からは新たな情報は何ら要求されない。バス・インターフェース207は、バス接続205上に入力トークン317を発行することによってイン転送を開始する。バス接続205のエラーに起因して、入力トークン317が、アドレス指定がされたUSBデバイスで、正しく受信されない。このため、アドレス指定がされたUSBデバイスは、入力トークン317に応答しない。USBデバイスによる応答がない場合、バス・インターフェース207は、エラーが発生したことをプロセッサ206に知らせるために、プロセッサ206に対してインタラプト接続210を介してインタラプト信号318を送信する。これにより、イン転送306が完了する。

10

【0038】

イン転送307では、バス・インターフェース207は、同一のリクエスト特性を有するイン転送304に続き、バッファ208から再びリクエスト特性を読み出すが、プロセッサ206から新たな情報は何ら要求されない。バス・インターフェース207は、バス接続205上に入力トークン319を発行することによってイン転送を開始する。アドレス指定がされたUSBデバイスは、有効なデータがないことを示す、NAKハンドシェイク320によって応答する。これにより、イン転送307が完了する。

20

【0039】

イン転送308では、バス・インターフェース207は、同一のリクエスト特性を有するイン転送304に続き、バッファ208から再びリクエスト特性を読み出すが、プロセッサ206から新たな情報は何ら要求されない。バス・インターフェース207は、バス接続205上に入力トークン321を発行することによってイン転送を開始する。アドレス指定がされたUSBデバイスは、命令を完了することができないことを示す、STALLハンドシェイク322によって応答する。バス・インターフェース207は、STALLハンドシェイク322の受信で、プロセッサ206にインタラプト接続210を介してインタラプト信号323を送信する。これにより、イン転送308が完了する。

30

【0040】

イン転送304、305、306、307および308が、バス・システムの機能を明らかにするためのみに例示されていて、かつバス・システムの機能が、ここに示されているイン転送にも、イン転送が示される順序にも限定されないことは、明らかであろう。

【0041】

図3Bは、データ伝送のためのトランザクションを示すメッセージ・ダイアグラムである。メッセージ・ダイアグラムには、アウト転送からなるインタラプト・トランザクションのための図1のシステムの動作を図示する。上から下へ時間が経過している。プロセッサ206とバス・インターフェース207との間の情報交換は、第1のライン301と第2のライン302との間の矢印によって表されている。バス・インターフェースと第2の端末203、204との間の情報交換は、第2のライン302と第3ライン303との間の矢印によって表されている。

40

【0042】

プロセッサ206は、第2の端末（USBデバイス）203、204のうちの1つへデータを転送することが必要となる。このため、バス・インターフェース207は、バス接続205に受信データのリクエストを繰り返し発行しなければならない。プロセッサ206に対するオーバーヘッド通信を減らすために、プロセッサ206は、リクエスト特性を一度提供すれば充分とすべきである。アドレス指定されたUSBデバイスがデータを受け入れる場合、バス・インターフェース207は、インタラプト接続210によってプロセッサ206にインタラプトを行い、プロ

50

セッサ206に知らせなければならない。エラーが発生する場合にも、プロセッサ206に通知を行う必要がある。このような状況では、バス・インターフェース207は、プロセッサ206にもインタラプトをかけなければならない。他の状況では、バス・インターフェース207は、プロセッサ206にインタラプトを行ってはならないので、プロセッサ206からの動作は要求されない。

【0043】

アウト転送324では、プロセッサ206は、メッセージ329によって、リクエスト特性をバス・インターフェース207に送信する。ここで、リクエスト特性はバッファ208に格納される。バス・インターフェース207は、バッファ208から読み出される、メッセージ329内のリクエスト特性に基づき、バス接続205上に出力トークン330を発行することによってアウト転送を開始する。続いて、バス・インターフェース207は、データ・パケット331を発行する。アドレス指定がされたUSBデバイスである第2の端末は、ACKハンドシェーク332を送信することによって応答する。バス・インターフェース207は、ACKハンドシェーク332の受信で、データ転送が成功したことをプロセッサ206へ知らせるために、プロセッサ206に対してインタラプト接続210を介して、インタラプト信号333を送信する。別の装置では、プロセッサ206は、インタラプトされない場合、データ伝送が成功したと仮定するため、バス・インターフェース207は、プロセッサ206にインタラプト信号333を送信しない。

10

【0044】

アウト転送325では、バス・インターフェース207は、同一のリクエスト特性を有するアウト転送324に続き、バッファ208から再びリクエスト特性を読み出すが、プロセッサ206から新たな情報は何ら要求されない。バス・インターフェース207は、バス接続205上に出力トークン334を発行することによってアウト転送を開始する。バス・インターフェースは、続いて、データ・パケット335を発行する。アドレス指定がされたUSBデバイスは、転送されたデータを処理するための用意ができていないことを示すために、NAKハンドシェーク336を発行することによって応答する。これにより、アウト転送325が完了する。

20

【0045】

アウト転送326では、バス・インターフェース207は、同一のリクエスト特性を有するアウト転送324に続き、バッファ208から再びリクエスト特性を読み出すが、プロセッサ206から新たな情報は何ら要求されない。バス・インターフェース207は、バス接続205上に出力トークン337を発行することによってアウト転送を開始する。バス・インターフェース207は、続いて、データ・パケット338を発行する。それに応じて、USBデバイスは、アウト転送を処理することができないことを示すために、STALLハンドシェーク339を送信する。バス・インターフェース207は、STALLハンドシェーク339の受け取りによって、アドレス指定がされたUSBデバイスのステータスをプロセッサ206に知らせるために、インタラプト接続210を介してインタラプト信号340をプロセッサ206に送信する。これにより、アウト転送326が完了する。

30

【0046】

アウト転送327では、バス・インターフェース207は、同一のリクエスト特性を有するアウト転送324に続き、バッファ208から再びリクエスト特性を読み出すが、プロセッサ206から新たな情報は何ら要求されない。バス・インターフェース207は、バス接続205上に出力トークン341を発行することによってアウト転送を開始する。バス・インターフェースは、続いて、データ・パケット342を発行する。バス接続205のエラーに起因して、データ・パケット342が、アドレス指定がされたUSBデバイスで、正確に受信されない。このため、アドレス指定がされたUSBデバイスは、応答しない。USBデバイスによる応答がない場合、バス・インターフェース207は、プロセッサ206にその処理を通知するために、インタラプト接続210を介してインタラプト信号343を送信する。これにより、アウト転送327が完了する。

40

【0047】

アウト転送328では、バス・インターフェース207は、同一のリクエスト特性を有するアウト転送324に続き、バッファ208から再びリクエスト特性を読み出すが、プロセッサ206か

50

ら新たな情報は何ら要求されない。バス・インターフェース207は、バス接続205上に出カトークン344を発行することによってアウト転送を開始する。エラー条件の発生に起因して、バス・インターフェース207は、順次データ・パケットの発行を開始せず、または既に開始されている順次データ・パケットの発行を完了させない。このため、バス・インターフェース207は、プロセッサ206にその処理を通知するために、インタラプト接続210を介してインタラプト信号345を送信する。これにより、アウト転送328が完了する。

【0048】

アウト転送324、325、326、327および328が、バス・システムの機能を明らかにするためのみに例示されていて、かつバス・システムの機能が、ここに示されているアウト転送にも、アウト転送が示される順序にも限定されないことは、明らかであろう。

10

【0049】

図4は、本発明のバス・インターフェースを示すブロック図である。図4に示されるブロック図は、バス・インターフェース集積回路のブロック図である。バス・インターフェース集積回路401は、二つの役割を有している。それは、USBデバイスの機能を果たしているバス・ステーションの部分と、USBホストの機能を果たしているバス・ステーションの部分である。このことは、特にUSB On-The-Go (OTG) 装置に有効である。例えば、このようなUSB OTG装置は、デジタル・スチルカメラ、デジタルビデオ・カメラ、パーソナル携帯情報機器またはオシロスコープとすることができる。そして、それらは、特定の動作モードで、USBを介して、例えば、大容量記憶装置（例えば、ハードディスク装置、光レコーダまたは半導体のレコーダ）、プリンタまたは携帯電話機に接続される。この動作モードでは、USB OTG装置は、USBホストとして機能を果たさなければならない。他の動作モードでは、USB OTGは、USBを介して、例えば、パーソナルコンピュータに接続される。この動作モードでは、USB OTG装置は、USBデバイスとして機能を果たさなければならない。USB OTG装置のバス・インターフェース集積回路が作動するモードは、USB OTG装置内のプロセッサによって決定され、かつ制御されるソフトウェアで実現すればよい。

20

【0050】

USB OTG装置が携帯用の場合、例えば、バス・トラフィックが低い、あるいはバス・トラフィックが全くない期間には、バス・インターフェース集積回路が特定の機能ブロックをシャットダウンする、または、低いクロック・スピードで作動する、低消費電力モードが用意されていることが好ましい。このような低い消費電力モードは、特定あるいは全ての機能ブロックの動作に戻すために、またはバス・インターフェース集積回路のクロック・スピードを増加させるために、USB OTG装置のプロセッサによって制御される、いわゆるリモート・ウェイク・アップ特性を備えていることによって強化することができる。例えば、プロセッサが、バス・トラフィックの増加を予期する場合、リモート・ウェイク・アップ特性を活性化すればよい。

30

【0051】

バス・インターフェース集積回路401は、バス・インターフェース集積回路401がUSBホストとして作動する場合には、プロセッサとの接続402、USBに対して接続するための接続403を備えている。そして、バス・インターフェース集積回路401がUSBデバイスとして作動する場合には、USBに対して接続するための接続404を備えている。実際に実施する場合には、内部マルチプレクサが、バス信号の内部ルーティングの処理を行うときに、接続403および404が、同じ入出力端子を共有することがある点に注意する。このような実施は、接続403および404が、同時に用いられないことが保証される場合には好ましい。

40

【0052】

バス・インターフェース集積回路401は、さらにホスト・デバイス・マルチプレクサ (HMD) 405、タイミング・ブロック (TB) 406、ホストインターフェース (HI) 407、デバイス・インターフェース (DI) 408、ホスト・コントローラ (HC) 409、デバイス・コントローラ (DC) 410、ホスト・バッファ (HB) 411、デバイス・バッファ (DB) 412、ホスト・トランシーバ (HT) 413およびデバイス・トランシーバ (DT) 414を備えている。

【0053】

50

ホスト・デバイス・マルチプレクサ405は、プロセッサとの接続402に接続されている。プロセッサの制御の下で、ホスト・デバイス・マルチプレクサ405が、プロセッサによって、バス・インターフェース集積回路401の通信の内部ルーティングを決定することが好ましい。バス・インターフェース集積回路401がUSBホストとして作動する場合には、ホスト・デバイス・マルチプレクサ405は、ホストインターフェース407を介して、ホスト・コントローラ409を、プロセッサに連結する。バス・インターフェース集積回路401がUSBデバイスとして作動する場合には、ホスト・デバイス・マルチプレクサ405は、デバイス・インターフェース408を介して、デバイス・コントローラ410を、プロセッサに連結する。ホスト・コントローラ409は、ホスト・トランシーバ413を介してバス接続403にアクセスする。デバイス・コントローラ410は、デバイス・トランシーバ414を介してバス接続404にアクセスする。

10

【0054】

ホスト・バッファ411は、ホスト・デバイス・マルチプレクサ405を介したプロセッサと、ホスト・コントローラ409との双方によってアクセス可能である。ホスト・バッファ411は、例えば、プロセッサによって発行されたリクエスト特性を格納するため、かつホスト・コントローラ409とUSBデバイスとの間の通信結果を格納するために用いることができる。

【0055】

デバイス・バッファ412は、ホスト・デバイス・マルチプレクサ405を介したプロセッサと、デバイス・コントローラ410との双方によってアクセス可能である。デバイス・バッファ412は、例えば、USBホストに送信するために、プロセッサによって生成されるデータをバッファリングするために用いることができる。

20

【0056】

バス・インターフェース集積回路と広範囲にわたるプロセッサとの間の提携をサポートするために、ホスト・バッファ411およびデバイス・バッファ412は、直接メモリアクセス（DMA）モードのプロセッサと並列入出力モードのプロセッサとの双方によって、アクセス可能であることが好ましい。

【0057】

バス・インターフェース集積回路401内のバッファとプロセッサとの間の通信のいくつかの手段をサポートすることが好ましい。これにより、プロセッサの選択がより広がる結果になる。DMA動作のサポートは、例えば、より精巧なコンピュータ・システムによって行うことができる。バス・インターフェースによって、さらにDMA動作をサポートする場合には、プロセッサの通信オーバーヘッドを更に減らすことができる。一方、あまり精巧でないバス・ステーションでは、DMAコントローラのサポートがない全ての通信を、プロセッサが処理することになる。このような状況では、バス・インターフェースが、並列入出力モードで、バス・インターフェースとプロセッサとのデータ交換をサポートすることが必要となる。

30

【0058】

上述の実施例は、バス・インターフェース集積回路に関するものであるが、当業者にとって、別のブロック、またはダイアグラムのブロックのグループを、別の集積回路および個別素子で具体化できることは明らかである。このように、ブロック図によって記載されているバス・インターフェースに関する別の実施例では、各種のブロックの機能を実行している複数の集積回路および個別素子を備えていることができる。

40

【0059】

バス・システムの要求に従い、集積回路としてバス・インターフェースを実現することが好ましい。集積回路の素子数の減少と、これによる領域の最小化とを可能にする機能の統合化により、バス・インターフェースを備えているバス・ステーションの物理的な大きさは減少するという結果になる。さらに、バス・インターフェースを集積回路として実現することによって、消費電力を減少させることができる。

【0060】

バス・システムの要求に従い、極力、標準的に構築されているブロックを用いて、複数の

50

集積回路および個別電子部品を有するバス・インターフェースを実現することが好ましい。これにより、バス・インターフェースの原価を低減させることができる。

【0061】

図5は、先入れ先出し (first-in, first-out : FIFO) 方法で構成されるバッファ501の模式的な概要である。この方法は、ホスト・バッファ411を効率よく体系づけるものである。FIFOバッファの上部は、矢印502によって示されている。FIFOバッファの下部は、矢印503によって示されている。バッファは、例えば、USBトランザクションまたはそれ以上に特にUSBインタラプト・トランザクションを開始するためのいくつかのリクエスト504、505、506のリクエスト特性を格納する。概要としては、最初に発行されるリクエスト特性は、504によって示され、2番目に発行されるリクエスト特性は、505によって示され、最後に発行されるリクエスト特性は、506によって示される。領域507は、残りのリクエスト特性を示す。それと共に、領域508は、FIFOバッファの残りの未使用空間を示す。リクエスト特性は、2つの部分に分けられる。それは、すなわち、転送記述子またはヘッダ509の部分と、規定の場合のペイロード・データ510の部分とである。転送記述子509は、ホスト・コントローラ409によって処理されるトランザクションの特性が記述されている。ペイロード・データ510は、ホスト・コントローラ409によって処理されるトランザクションのデータ・パケットに含まれるべきデータである。転送記述子509は、トランザクションの結果を格納するために、ホスト・コントローラ409によっても用いられる。したがって、プロセッサおよびホスト・コントローラ409は、FIFOバッファ501にアクセスすることになる。

10

20

【0062】

先入れ先出しモードのバッファを準備することが好ましい。このように、プロセッサとコントローラとの双方が、バッファに格納されているデータに、相対的にシンプルに、アクセスすることができる。

【0063】

リクエスト特性および第2の端末との通信結果を格納するためにバッファを用いることが好ましい。リクエスト特性とそれらのリクエスト結果とを格納するために同じバッファが用いられるので、バッファまたは少なくともバッファの指定された区間にリード・アクセスするプロセッサを与えることによって、効率のよい通信手段が確立される。従って、リクエスト特性を書き込む場合に、アクセスのタイプを書き込みモードから読み出しモードへ変更するのみで、プロセッサは、同一の手段によりバッファにアクセスすることができる。

30

【0064】

プロセッサによって提供される情報に基づいて、バス・インターフェースに正確なポーリング・インターバルを決定させることが好ましい。可能な限りポーリング・レートの数を制限することにより、別のポーリング・レートの別のリクエストを処理する際のバス・インターフェースの作業をシンプルにする。例えば、プロセッサが、1、2、3、および128ミリ秒の間のいかなるポーリング・レートを選択できる場合でも、バス・インターフェースは、選択されたポーリング・レートを、以下の実ポーリング・レート、すなわち、1、2、4、8、16、32、64、または128ミリ秒の1つにマッピングする。こうして、別のポーリング・レートの処理の見込みの数は、128から8に低減する。バス・インターフェースが、プロセッサによって選択されたポーリング・レートに最も近い実ポーリング・レートが選択されるポーリング・レートをマッピングすることが好ましい。

40

【0065】

下記の表1は、転送記述子のフォーマットを示す。バイト (Byte) 0、バイト 1、...、バイト7として示されている8バイトから構成されているフォーマットが示されている。すべてのバイトは、0、1、...、7として列挙されている8ビットから構成されている。

【0066】

【表1】

-	7	6	5	4	3	2	1	0
Byte 0	実バイト (7:0)							
Byte 1	完了コード				アクティブ	トグル	実バイト (9:8)	
Byte 2	最大パケットサイズ (7:0)							
Byte 3	エンドポイント番号 (3:0)			最後	スピード	最大パケットサイズ (9:8)		
Byte 4	全バイト (7:0)							
Byte 5	0	0	応答	応答	方向トークン (1:0)		全バイト (9:8)	
Byte 6	0	デバイスアドレス						
Byte 7	インターラプトポーリング・インターバル (4:0)							

10

下記の表2によって、上の表1のパラメータの意味を明らかにする。

【 0 0 6 7 】

【表 2】

名称	長さ (ビット)	記述
実バイト	10	転送されたバイト数を含む
完了 コード	4	関係する転送識別子に対してエラーの発生を備えている最後のトランザクションがどのように完了されたかを示す
アクティブ	1	トランザクションが実行されなければならないかを示す
トグル	1	データPID値を生成または比較するために用いられる。トグルは、各々の伝送が成功した後またはデータ・パケットの受信の後更新される。
最大 パケットサイズ	10	単一のデータ・パケットの関係するエンドポイントに送信される、または関係するエンドポイントから受信されるバイト最大数を示す
応答	1	未使用
スピード	1	関係するエンドポイントのスピードを示す
全バイト	10	転送されるバイト全数を示す
方向 トークン	2	関係する転送記述子がイン転送またはアウト転送を備えているか否かを示す
デバイス アドレス	7	アドレス指定すべきUSBデバイスのアドレスを示す
インタラプト・ ポーリング・ インターバル	4	インタラプト転送のポーリング・インターバルを示す

20

30

40

50

【図面の簡単な説明】

【0068】

【図1】USB規格のリビジョン2.0において指定されている、インタラプト・トランザクションフォーマットを示しているダイアグラムである。

【図2】本発明のバス・システムを示すブロック図である。

【図3A】データ伝送のためのトランザクションを示すメッセージ・ダイアグラムである。

【図3B】データ伝送のためのトランザクションを示すメッセージ・ダイアグラムである。

10

【図4】本発明のバス・インターフェースを示すブロック図である。

【図5】先入れ先出し方法で構成されるバッファの模式的な概要を示す図である。

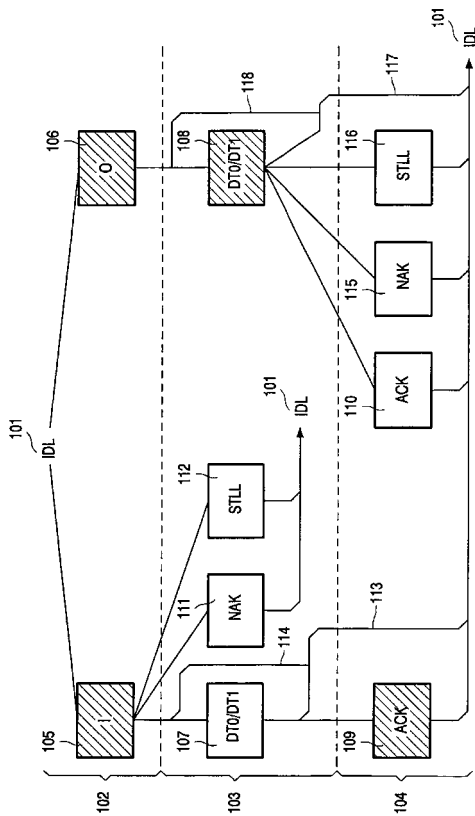
【符号の説明】

【0069】

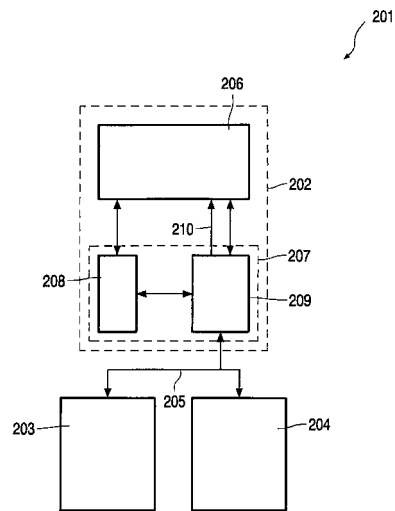
101	アイドルモード	
102	セットアップステージ	
103	データステージ	
104	ステータスステージ	
105、106	トークン・パケット	
107、108	データ・パケット	20
109、110、111、112、115、116	ハンドシェイク・パケット	
113、114、117、118	ライン	
201	バス・システム	
202	第1の端末	
203、204	第2の端末	
205	バス接続	
206	プロセッサ	
207	バス・インターフェース	
208	バッファ	
209	コントローラ	30
210	インタラプト接続	
301	第1のライン	
302	第2のライン	
303	第3のライン	
304、305、306、307、308	イン転送	
309、311、315、329	メッセージ	
310、314、317、319、321	入力トークン	
312、316、318、323、333、340、343、345	インタラプト信号	
313	ACK信号	
320、336	NAKハンドシェーク	40
322、339	STALLハンドシェーク	
324、325、326、327、328	アウト転送	
330、334、337、341、344	出力トークン	
331、335、338、342	データ・パケット	
332	ACKハンドシェーク	
401	バス・インターフェース集積回路	
402、403、404	接続	
405	ホスト・デバイス・マルチプレクサ	
406	タイミング・ブロック	
407	ホストインターフェース	50

- 408 デバイス・インターフェース
- 409 ホスト・コントローラ
- 410 デバイス・コントローラ
- 411 ホスト・バッファ
- 412 デバイス・バッファ
- 413 ホスト・トランシーバ
- 414 デバイス・トランシーバ
- 501 バッファ
- 502、503 矢印
- 504、505、506 リクエスト
- 507、508 領域
- 509 転送記述子
- 510 ペイロード・データ

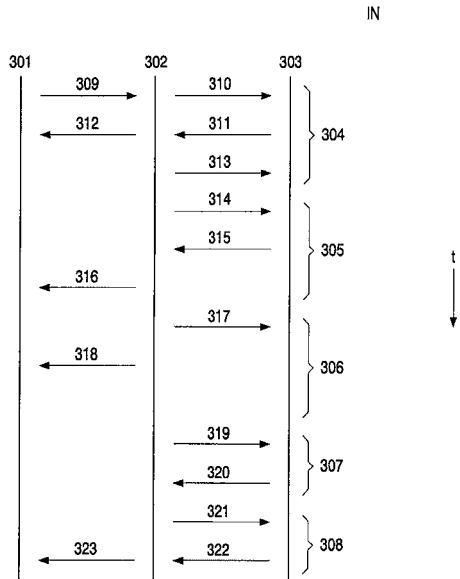
【図1】



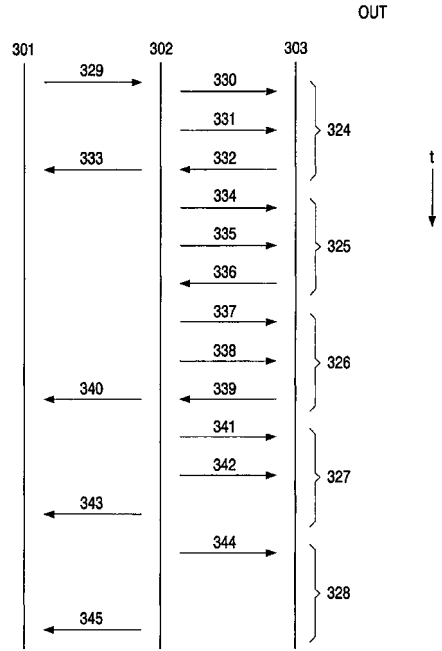
【図2】



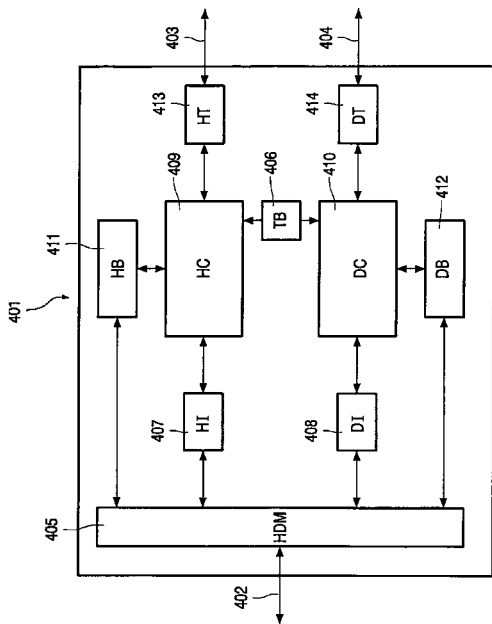
【 図 3 A 】



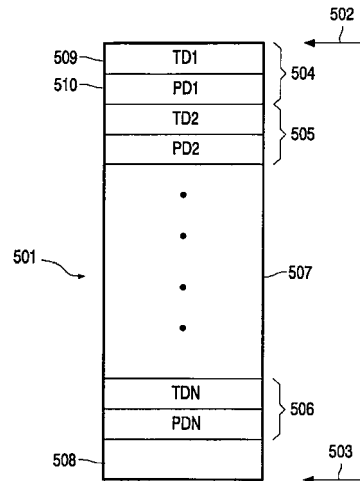
【 図 3 B 】



【 図 4 】



【 図 5 】



【国際公開パンフレット】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
10 April 2003 (10.04.2003)

PCT

(10) International Publication Number
WO 03/029996 A1

- (51) International Patent Classification: G06F 13/24, 13/42
- (72) Inventors: CHANG, Yeow, K.; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL); ZHANG, Zhenyu; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).
- (21) International Application Number: PCT/HB02/03812
- (74) Agent: DUIJVESTIJN, Adriaans, J.; International Octrooibureau B.V., Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).
- (22) International Filing Date: 16 September 2002 (16.09.2002)
- (81) Designated States (national): CN, JP, KR.
- (25) Filing Language: English
- (84) Designated States (regional): European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR).
- (26) Publication Language: English
- (30) Priority Data: PCT/SG01/00194 27 September 2001 (27.09.2001) SG
- Published: with international search report

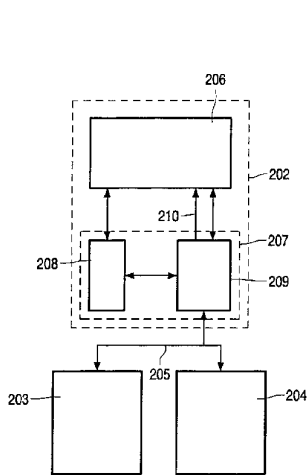
(71) Applicant: KONINKLIJKE PHILIPS ELECTRONICS N.V. [NL/NL]; Groenevoudseweg 1, NL-5621 BA Eindhoven (NL).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: BUS SYSTEM AND BUS INTERFACE FOR CONNECTION TO A BUS



WO 03/029996 A1



(57) Abstract: The invention relates to a bus system comprising a first station (202) and a second station (203, 204) coupled by a bus for transferring data and control signals. The bus is conceived to operate according to a protocol in which the first station (202) repeatedly sends requests to the second station (203, 204). The first station (202) comprises a processor (206), a bus interface (207), and a buffer (208) coupled to said interruptible processor and said bus interface, the processor (206) being operable to generate request properties for the requests to be sent by the first station (202). Furthermore, the processor (206) is operable to handle said data items. The buffer (208) is operable to store said request properties. The bus interface (207) is operable to generate said requests from said stored request properties and to repeatedly send said requests.

WO 03/029996

PCT/IB02/03812

1

Bus system and bus interface for connection to a bus

The invention relates to a bus system comprising a first station and a second station coupled by a bus for transferring data and control signals, said bus being conceived to operate according to a protocol in which said first station repeatedly sends requests to said second station.

5 The invention also relates to a bus interface, comprising a connection for a bus and an input for receiving request properties from a processor.

10 Such a bus system is disclosed in the USB specification. The USB specification is a standard promulgated by a group of companies including Compaq Computer Corporation, Hewlett-Packard Company, Intel Corporation, Lucent Technologies Inc, Microsoft Corporation, NEC Corporation, and Koninklijke Philips Electronics N.V. Described below are various aspects of the USB relevant to a complete understanding of the present invention. Further background concerning the USB may be obtained from USB
15 Specification, Revision 2.0, which is included herein by reference.

The USB is a standard bus designed to support data exchange between a personal computer and a wide variety of peripherals such as webcams, printers, scanners, microphones and keyboards. The USB thus replaces existing interfaces such as the RS-232C serial ports, parallel ports, and PS/2 interfaces. Further developments of the USB, for
20 example, as in USB On-The-Go, aim at bus systems where another apparatus replaces the personal computer. Such an apparatus could be, for example, a digital still camera, a video camera, a personal digital assistant, or a cellular phone.

In a bus system like a USB system, electronic bus stations are connected to one another via a bus. Usually a bus station comprises a processor or computing system to
25 process or generate data and to control the flow of data. USB stations are defined as hosts (e.g. personal computers) and as devices (peripherals). In general there is only one host on any USB system. A USB interface at the host is also referred to as a host controller. The host controller may be implemented in a combination of hardware, firmware, or software. USB devices are defined as hubs which provide additional attachment points or functions which

WO 03/029996

PCT/IB02/03812

2

provide capabilities to the system. The devices are connected to the host via a tiered-star topology. Hubs indicate the attachment or removal of a USB device. The host determines if a newly attached USB device is a hub or a function and assigns a unique USB address to the USB device. All USB devices are accessed by a unique USB address. Each device

5 additionally supports one or more endpoints with which the host may communicate. The following description will be limited to USB devices defined as functions.

The USB is a polled bus. The host controller initiates all data transfers. Most bus transactions involve the transmission of up to three packets. Each transaction begins when the host controller, on a scheduled basis, sends a USB packet describing the type and direction of the transaction, the USB device address, and endpoint number. This packet is referred to as the "token packet". The USB device that is addressed selects itself by decoding the appropriate address fields. In a given transaction, data is transferred either from the host to a device or from a device to the host. The direction of data transfer is specified in the token packet. The source of the transaction then sends a data packet or indicates it has no data to transfer. The destination, in general, responds with a handshake packet indicating whether the transfer was successful.

The USB architecture comprehends three basic types of data transfer: (1) isochronous transfers: periodic, continuous communication between host and device, typically used for time-relevant information, which occupies a pre-negotiated amount of USB bandwidth with a pre-negotiated latency; (2) interrupt transfers: low-frequency, bounded latency communication, typically used for asynchronous data such as characters or coordinates with human perceptible echo or feedback response characteristics, and; (3) bulk transfers: non-periodic, large-packet bursty communication, typically used for data that can use any available bandwidth and can also be delayed until bandwidth is available.

25 Usually both types of bus station, host and device, comprise a processor or computing system to either process or generate data or to control the flow of data or both. A straightforward implementation of a USB system as described above could lead to a significant overhead for both the processor in the bus station acting as a host and the processor in the bus station acting as a device. This holds especially for the interrupt transfers. Due to the polled nature of the bus system, the host needs to interrogate the involved device, for instance a keyboard, on a regular basis.

WO 03/029996

PCT/IB02/03812

3

It is inter alia an object of the invention to reduce the amount of overhead needed to transport data from a processor in a first station of a bus system to a second station of the bus system or vice versa.

To this end the invention provides a bus system as defined in the opening paragraph which is characterized in that said first station comprises a processor, a bus interface, and a buffer coupled to said interruptible processor and said bus interface, where

- said processor is operable to generate request properties for said requests and for handling said data items;
- said buffer is operable to store said request properties;
- said bus interface is operable to generate said requests from said stored request properties and to repeatedly send said requests.

In operation the processor within the first station needs to provide information to the second station or needs to obtain information from the second station. To this end, the first station needs to communicate with the second station. In a polled bus system, such as a USB system, the communication is initiated by the first station, being a host in a USB system. If the second station, being a device in a USB system, is only capable of receiving or providing data on an irregular basis, it happens frequently that the first station needs to poll the second station quite a number of times before a successful transfer of information takes place. In other words under these circumstances the first station needs to re-send the same request relatively often to the second station before a successful transfer of information takes place.

In a conventional approach said processor would need to prepare said request every time it needs to be send and said bus interface would then actually send said request via said bus to said second station. In the bus system according to the invention the processor generates or determines the properties of the request. By enhancing the capabilities of the bus interface, said bus interface can actually generate the request out of said properties provided by the processor and send the request over the bus to said second station as many times as is required.

The bus system according to the invention also comprises a buffer within the first station that is used to store the request properties provided by the processor. The buffer is preferably accessible for both the processor and the bus interface. In this way the communication overhead of the processor is reduced.

An embodiment of the bus system according to the invention is characterized in that said request properties comprise a polling rate of said requests and, if applicable, a

WO 03/029996

PCT/IB02/03812

4

payload of said requests. Including the polling rate in the request properties allows more flexibility. The processor may have information on the characteristics of the second station that can be used to adjust the polling rate of the bus interface. For instance, a USB system establishes a time base of either 1 millisecond, called a frame, or of 125 microsecond, called a microframe. If the polling rate information were not included in the request properties, the bus interface would use a fixed polling rate, presumably sending a request at the maximum polling rate of once per (micro)frame. The polling rate information being included in the request properties, the bus interface adjusts the polling rate as needed, using low polling rates whenever possible and high polling rates when required. Bus load and communication overhead for the second station are thus reduced.

In case the first station sends a request to receive data to the second station the request itself does not necessarily contain a payload, i.e. a data item. Including payload information in the request properties that are provided by the processor has the advantage that data can be sent repeatedly from the first station to the second station, without requiring interference from the processor within the first station, thus further reducing the communication overhead of the processor.

A further embodiment of the bus system according to the invention is characterized in that said second station responds to said requests, that said processor is an interruptible processor and said bus interface is operable for interrupting said processor upon reception of selected responses of said second station. It is not necessary to interrupt said processor if no further action is required. Thus it is not necessary to report back the result of the communication with the second station if either the communication is completed as expected or if no new information, no data items, are provided by the second station. For example, in USB system the second station may respond by sending a message with a data item acknowledge signal, a negative acknowledge signal, by sending a signal to indicate it is stalled. Furthermore, if the second station does not respond by sending a signal, or if no signal is received by the first station due to an error in the communication between the first station and the second station, this may be interpreted as a response too. Thus the selected responses may comprise the absence of a response signal from the second station. Communication overhead of the processor is further reduced if the processor is only interrupted upon reception of the responses that require further action by the processor. To this end the bus interface selects those responses out of the received responses. Programmed to handle such situations the processor can decide upon necessary further actions.

WO 03/029996

PCT/IB02/03812

5

A bus interface according to the invention comprises a connection for a bus and an input for receiving request properties from an interruptible processor and is characterized in that it comprises a buffer operable to store said request properties and a controller operable to generate requests out of the stored request properties and to repeatedly send said requests to the connection.

These and other aspects of the invention will be apparent from and elucidated with reference to the embodiments described hereinafter.

10 Fig. 1 is a diagram showing the interrupt transaction format as specified in the USB specification, revision 2.0;

Fig. 2 is a block diagram showing a bus system according to the invention;

Fig. 3A is a message diagram showing transactions for data transmission;

Fig. 3B is a message diagram showing transactions for data transmission;

15 Fig. 4 is a block diagram showing a bus interface according to the invention; and

Fig. 5 is schematic overview of a buffer organized in a first-in, first-out manner.

In these figures identical parts are denoted by identical references.

20

Fig. 1 is a diagram showing the interrupt transaction format as specified in the USB specification, revision 2.0. Within USB all communication is initiated by the host. All USB devices are accessed by a unique USB address. Each device additionally supports one or more endpoints with which the host may communicate.

25 The USB supports functional data and control exchange between the USB host and USB devices. USB data transfers take place between host software and a particular endpoint on a USB device. The USB architecture comprehends three basic types of transfer:

- isochronous transfers: isochronous or streaming real time data which occupies a prenegotiated amount of USB bandwidth with a pre-negotiated latency;
- 30 - interrupt transfers: asynchronous interactive data such as characters or coordinates with human perceptible echo or feedback response characteristics, and

WO 03/029996

6

PCT/IB02/03812

bulk transfers: asynchronous block transfer data which is generated or consumed in relatively large and bursty quantities and has wide dynamic latitude and transmission constraints.

If there are no on-going transactions, a USB system is in an Idle mode 101.

- 5 Bus transactions including data transfers generally involve three stages: a Setup stage 102, a Data stage 103, and a Status stage 104.

Bus transactions including data transfers generally involve the transmission of three different packets, e.g. a Token packet 105, 106, a Data packet 107, 108, and a Handshake packet 109, 110, 111, 112, 115, 116.

- 10 Interrupt transactions may consist of IN transfers or OUT transfers. In the case of an IN transfer the USB host will start the transaction by sending an IN token 105. Upon receipt of the IN token, a USB device may return data 107, a negative acknowledge signal (NAK) 111, or a signal indicating the USB device is unable to complete a command (STALL) 112. If the endpoint has no new interrupt information to return (i.e. no interrupt is pending), the USB device returns a NAK handshake 111 during the Data stage 103. If the
15 USB device is unable to complete a command, the USB device returns a STALL handshake 112. If an interrupt is pending, the USB device returns the interrupt information as a Data packet 107. The USB host in response to receipt of the Data packet 107 issues either an ACK handshake 109 (acknowledge signal) if the data was received error-free or returns no
20 handshake, indicated by line 113, if the Data packet 107 was received corrupted. Furthermore the USB host will not issue an ACK 109 in case no response to the IN token 105 is received at all from the USB device. This situation is indicated by line 114. After completion of the IN transfer, the USB system returns to the Idle mode 101.

- In the case of an OUT transfer the USB host will start the transaction by
25 sending an OUT token 106. This is followed by sending a Data packet 108. Upon receipt of the Data packet 108, the USB device may issue in the Status stage 104 an ACK handshake 110 if the data was received error free, a NAK handshake 115 indicating the device is not able to process the data, or a STALL handshake 116 indicating the USB device is currently stalled. The USB device returns no handshake, indicated by line 117, if the Data packet 107
30 was received corrupted. Furthermore the USB device will not issue an ACK handshake 110 in case it does not receive data from the USB host, a situation indicated by line 118. After completion of the OUT transfer the USB system returns to the Idle mode 101.

All packets begin with a SYNC field, which is used by input circuitry to align incoming data with a local clock. A packet identifier (PID) immediately follows the SYNC

WO 03/029996

PCT/IB02/03812

7

field of every USB packet. The PID consists of a four-bit ID field followed by a four-bit check field. The PID indicates the type of packet (e.g. Token 105, 106, Data 107, 108, and Handshake 109, 110, 111, 112, 115, 116) and the format of the packet along with the type of variant detection applied to the packet. The four-bit check field of the PID ensures reliable
5 decoding.

A Token packet 105, 106 includes an ADDR field which specifies the target USB device, via its USB address, that is the source or destination of a Data packet 107, 108, depending on the value of the Token PID. USB devices must perform a complete decoding of the ADDR field.

10 A Data packet 107, 108 includes a data field which may range from zero to N bytes and must amount to integral numbers of bytes. Data bits within each byte are shifted out most significant bit first. The data field of a Data packet is divided into fields which give extra information to the USB device regarding the request being made. One field, the command code, tells what the targeted USB device should do, such as set up a read from a
15 USB device space or set up a write to a USB device space. The read space command is used to request information from a given location in a given space of a USB device. The command is used to set up the read, which is followed by a request from the host for the USB device to send data. The USB device then sends the data it has retrieved from the previously specified space. The write space uses exactly the same definition as the read space. However, the Data
20 packet contains data transferred by the host to the USB device.

A Handshake packet 109, 110, 111, 112, 115, 116 consists of only a PID. As indicated above, Handshake packets are used to report the status of a data transfer and can return values indicating successful reception of data, CRC failure, flow control, and certain fault conditions.

25 Fig. 2 is a block diagram showing a bus system according to the invention. The bus system 201 comprises a first station 202, acting as a USB host, and second stations 203 and 204, acting as USB devices. The stations 202, 203, 204 are coupled by bus connections 205. The first station 202 is shown in more detail. It comprises an interruptible processor 206 and a bus interface 207. The bus interface comprises a buffer 208 and a
30 controller 209. It is not necessary that the buffer is part of the bus interface. In an alternative arrangement the buffer could be placed within the first station 202, but outside the bus interface. Both the processor 206 and the controller 209 have access to the buffer 208. The controller is coupled to the bus connections 205. Furthermore the controller 209 has an interrupt connection 210 to the processor 206.

WO 03/029996

PCT/IB02/03812

8

Fig. 3A is a message diagram showing transactions for data transmission. The message diagram illustrates the operation of the system of Fig. 1 for interrupt transactions consisting of IN transfers. Going top-down the time increases. Information exchange between the processor 206 and the bus interface 207 is symbolized by arrows between a first line 301 and a second line 302. Information exchange between the bus interface and the second station 203, 204 is symbolized by arrows between the second line 302 and a third line 303.

The processor 206 requires data from one of the second stations 203, 204, a USB device. Thus the bus interface 207 should repeatedly issue a request for data to the bus connections 205. To reduce the communication overhead for the processor it should be sufficient to have the processor provide the request properties only once. If the addressed USB device returns data, the bus interface 207 should interrupt the processor via the interrupt connection and provide the processor with the data. In case an error occurs the processor should also be notified. In these situations the bus interface 207 should apply an interrupt to the processor too. In other situations no action is required from the processor, so that the bus interface 207 should not interrupt the processor.

In an IN transfer 304 the processor 206 in a message 309 sends the request properties to the bus interface 207, where the request properties are stored in the buffer. Based on the request properties within the message 309, which are retrieved from the buffer, the bus interface 207 initiates the IN transfer by issuing an IN token 310 on the bus. The second station, the addressed USB device, responds by sending a message 311 with a data item. Upon reception of this message the bus interface 207 sends an interrupt signal 312 via the interrupt connection 210, to the processor 206 to inform the processor 206 that new data is available. Furthermore, it issues an Acknowledge signal 313 to the addressed USB device. This completes the IN transfer 304.

In an IN transfer 305, following the IN transfer 304 with identical request properties, the bus interface 207 again retrieves the request properties from the buffer, while no new information is required from the processor 206. The bus interface 207 initiates the IN transfer by issuing an IN token 314 on the bus. The addressed USB device responds by sending a message 315 with a data item. Due to an error on the bus, the second message 315 is received corrupted by the bus interface 207. Therefore, the bus interface 207 does not issue an ACK handshake. Instead it sends an interrupt signal 316, via the interrupt connection 210, to the processor 206 to inform the processor 206 an error occurred. This completes the IN transfer 305.

WO 03/029996

PCT/IB02/03812

9

In an IN transfer 306, following the IN transfer 304 with identical request properties, the bus interface 207 again retrieves the request properties from the buffer 208, while no new information is required from the processor 206. The bus interface 207 initiates the IN transfer by issuing an IN token 317 on the bus. Due to a bus error the IN token 317 is
5 received corrupted at the addressed USB device. For this reason it does not respond to the IN token 317. As no response follows by the USB device, the bus interface 207 sends an interrupt signal 318, via the interrupt connection 210 to the processor 206 to inform the processor 206 an error occurred. This completes the IN transfer 306.

In an IN transfer 307, following the IN transfer 304 with identical request
10 properties, the bus interface 207 again retrieves the request properties from the buffer 208, while no new information is required from the processor 206. The bus interface 207 initiates the IN transfer by issuing an IN token 319 on the bus. The addressed USB device responds with a NAK handshake 320, indicating it has no data available. This completes the IN transfer 307.

In an IN transfer 308, following the IN transfer 304 with identical request
15 properties, the bus interface 207 again retrieves the request properties from the buffer 208, while no new information is required from the processor 206. The bus interface 207 initiates the IN transfer by issuing an IN token 321 on the bus. The addressed USB device responds with a STALL handshake 322, indicating it cannot complete the command. Upon receiving
20 the STALL handshake 322, the bus interface 207 sends an interrupt signal 323 via the interrupt connection 210 to the processor 206. This completes the IN transfer 308.

It will be clear that the IN transfers 304, 305, 306, 307 and 308 are just
illustrative examples to clarify the functioning of the bus system and that the functioning of
the bus system is limited neither to the IN transfers shown nor to the order in which the IN
25 transfers are shown.

Fig. 3B is a message diagram showing transactions for data transmission. The
message diagram illustrates the operation of the system of Fig. 1 for interrupt transactions
consisting of OUT transfers. Going top-down the time increases. Information exchange
between the processor 206 and the bus interface 207 is symbolized by arrows between a first
30 line 301 and a second line 302. Information exchange between the bus interface and the
second station 203, 204 is symbolized by arrows between the second line 302 and a third line
303.

The processor 206 needs to transfer data to one of the second stations 203,
204, a USB device. Thus, the bus interface 207 should repeatedly issue a request to receive

WO 03/029996

PCT/IB02/03812

10

data to the bus connections 205. To reduce the communication overhead for the processor 206 it should be sufficient to have the processor 206 provide the request properties only once. If the addressed USB device accepts data, the bus interface 207 should interrupt the processor via the interrupt connection and inform the processor. In case an error occurs the processor 5 206 should also be notified. In these situations the bus interface 207 should apply an interrupt to the processor 206 too. In other situations no action is required from the processor 206, so that the bus interface 207 should not interrupt the processor 206.

In a OUT transfer 324 the processor in a message 329 sends the request properties to the bus interface 207, where the request properties are stored in the buffer. 10 Based on the request properties within the message 329, which are retrieved from the buffer, the bus interface 207 initiates the OUT transfer by issuing an OUT token 330 on the bus. Subsequently, the bus interface issues a Data packet 331. The second station, the addressed USB device, responds by sending an ACK handshake 332. Upon reception of the ACK handshake 332, the bus interface 207 sends an interrupt signal 333, via the interrupt 15 connection 210, to the processor 206 to inform the processor 206 that the data is transferred successfully. In a different arrangement the bus interface 207 does not send the interrupt signal 332 to the processor 206, since the processor 206 will assume that a successful transmission of the data has occurred if it is not interrupted.

In an OUT transfer 325, following the OUT transfer 324 with identical request 20 properties, the bus interface 207 again retrieves the request properties from the buffer 208, while no new information is required from the processor 206. The bus interface 207 initiates the OUT transfer by issuing an OUT token 334 on the bus. Subsequently, the bus interface issues a Data packet 335. The addressed USB device responds by issuing a NAK handshake 336 to indicate it is not ready to process the transferred data. This completes the OUT 25 transfer 325.

In an OUT transfer 326, following the OUT transfer 324 with identical request properties, the bus interface 207 again retrieves the request properties from the buffer 208, while no new information is required from the processor 206. The bus interface 207 initiates the OUT transfer by issuing an OUT token 337 on the bus. Subsequently, the bus interface 30 issues a Data packet 338. In response the USB device sends a STALL handshake 339 to indicate it cannot process the OUT transfer. In receipt of the STALL handshake the bus interface sends an interrupt signal 340, via the interrupt connection, to the processor 206 to inform the processor on the status of the addressed USB device. This completes the OUT transfer 326.

WO 03/029996

PCT/IB02/03812

11

In an OUT transfer 327, following the IN transfer 324 with identical request properties, the bus interface 207 again retrieves the request properties from the buffer 208, while no new information is required from the processor 206. The bus interface 207 initiates the OUT transfer by issuing an OUT token 341 on the bus. Subsequently, the bus interface
5 issues a Data packet 342. Due to an error on the bus, the Data packet 342 is received corrupted by the addressed USB device. For this reason the addressed USB device does not respond. As no response follows by the USB device, the bus interface 207 sends an interrupt signal 343, via the interrupt connection 210, to the processor 206 so as to notify the processor. This completes the OUT transfer 327.

10 In an OUT transfer 328, following the OUT transfer 324 with identical request properties, the bus interface 207 again retrieves the request properties from the buffer 208, while no new information is required from the processor 206. The bus interface 207 initiates the OUT transfer by issuing an OUT token 344 on the bus. Due to the occurrence of an error condition the bus interface does not start or complete an already started issuing of a
15 subsequent Data packet. For this reason the bus interface 207 sends an interrupt signal 345 via the interrupt connection 210, to the processor 206 so as to notify the processor. This completes the OUT transfer 328.

It will be clear that the OUT transfers 324, 325, 326, 327 and 328 are just illustrative examples to clarify the functioning of the bus system and that the functioning of
20 the bus system is limited neither to the OUT transfers shown nor to the order in which the OUT transfers are shown.

Fig. 4 is a block diagram showing a bus interface according to the invention. The block diagram shown in the figure is a block diagram of a bus interface integrated circuit. The bus interface integrated circuit 401 has a dual role. It can be both part of a bus
25 station acting as a USB device and of a bus station acting as a USB host. This is especially useful in a USB On-The-Go (OTG) apparatus. Such a USB OTG apparatus could be, for example, a digital still camera, a digital video camera, a personal digital assistant, or an oscilloscope, which in a certain mode of operation is connected, via USB, for example, to a mass storage apparatus (e.g. a hard disk drive, optical recorder, or solid state recorder), a
30 printer, or a cellular phone. In this mode of operation the USB OTG apparatus should act as a USB host. In another mode of operation the USB OTG is connected, via USB, for example, to a personal computer. In this mode of operation the USB OTG apparatus should act as a USB device. The mode in which the bus interface integrated circuit of the USB OTG

WO 03/029996

PCT/IB02/03812

12

apparatus operates could be software determined and controlled by a processor within the USB OTG apparatus.

In case the USB OTG apparatus is portable it could be advantageous to include a low power consumption mode available in which the bus interface integrated circuits shuts down certain functional blocks or operates at a lower clock speed in time periods with, for example low bus traffic or no bus traffic at all. Such a low power consumption mode could be enhanced by including a so-called remote wake-up feature, controlled by the processor of the USB OTG apparatus, to bring back to operation certain or all functional blocks or to increase the clock speed of the bus interface integrated circuit. The remote wake-up feature could be activated, for example, if the processor anticipates increased bus traffic.

The bus interface integrated circuit 401 comprises a connection 402 with an processor, a connection 403 for connection to a USB in case the bus interface integrated circuit 401 operates as a USB host, and a connection 404 for connection to a USB in case the bus interface integrated circuit 401 operates as a USB device. Note that in a practical implementation connections 403 and 404 could share the same I/O terminals, while an internal multiplexer takes care of the internal routing of the bus signals. Such an implementation could be advantageous in case it is guaranteed that connections 403 are not used simultaneously.

The bus interface integrated circuit 401 also comprises a host-device multiplexer 405, a timing block 406, a host interface 407, a device interface 408, a host controller 409, a device controller 410, a host buffer 411, a device buffer 412, a host transceiver 413, and a device transceiver 414.

The host-device multiplexer 405 is connected to the connection 402 with the processor. Preferably under the control of the processor it determines the internal routing of the communication of the bus interface integrated circuit 401 with the processor. In case the bus interface integrated circuit 401 operates as a USB host, the host-device multiplexer 405 links the host controller 409, via the host interface 407, to the processor. In case the bus interface integrated circuit 401 operates as a USB device, the host-device multiplexer links the device controller 410, via the device interface 408, to the processor. The host controller 409 has access to the bus connection 403 via the host transceiver 413. The device controller 410 has access to the bus connection 404 via the device transceiver 414.

The host buffer 411 is accessible by both the host controller 409 and, via the host-device multiplexer 405, by the processor. It can be used, for example, for storing request

WO 03/029996

PCT/IB02/03812

13

properties issued by the processor and for storing results of the communication between the host controller 411 and a USB device.

The device buffer 412 is accessible by both the device controller 410 and via the host-device multiplexer 405, by the processor. It can be used, for example, for buffering data generated by the processor, so as to be sent to a USB host.

To support co-operation between the bus interface integrated circuit and a wide range of processors, the host buffer 411 and the device buffer 412 are preferably accessible by the processor in both a direct memory access (DMA) mode and a parallel input-output mode.

It is advantageous to support several ways of communication between the processor and the buffers within the bus interface integrated circuit 401. This results in a wider choice of processors. For example, a more sophisticated computing system will support DMA operation. If the bus interface also supports DMA operation, the communication overhead of the processor is further reduced. On the other hand, in a less sophisticated bus station the processor handles all communication without the support of a DMA controller. In such a situation the bus interface needs to support the data exchange between the bus interface and the processor in a parallel input-output mode.

Although the embodiment discussed above relates to a bus interface integrated circuit, it will be clear to a person skilled in the art that different blocks or groups of blocks in the diagram could be embodied in different integrated circuits and discrete components. Thus alternative embodiments relating to the bus interface described by the block diagram could comprise a plurality of integrated circuits and discrete components performing the functions of the various blocks.

Depending on the requirements of the bus system it is advantageous to realize the bus interface as an integrated circuit. The functional integration in an integrated circuit enables a reduction in component count and area minimization and hence will result in a reduction in physical size of a bus station comprising the bus interface. Furthermore, realizing the bus interface as an integrated circuit will allow for a reduction in power consumption.

Depending on the requirements of the bus system, it is advantageous to realize the bus interface with a plurality of integrated circuits and discrete electronic components, utilizing as much as possible standard building blocks. This reduces the cost price of the bus interface.

WO 03/029996

PCT/IB02/03812

14

Fig. 5 is schematic overview of a buffer 501 organized in a first-in, first-out (FIFO) manner. This is an efficient manner to organize the host buffer 411. The top of the FIFO buffer is indicated by an arrow 502. The bottom of the FIFO buffer is indicated by an arrow 503. The buffer stores request properties of several requests 504, 505, 506, for example, for initiating USB transactions or more in particular USB interrupt transactions. In the overview the request properties first issued are indicated by 504, the request properties issued second are indicated by 505, the request properties issued last are indicated by 506. The area 507 indicates the remaining request properties, while the area 508 indicates the remaining space in the FIFO buffer not being used. Request properties are divided in two parts: a transfer descriptor or header 509 and, if applicable, payload data 510. The transfer descriptor 509 describes the characteristics of the transaction to be handled by the host controller 409. The payload data 510 is the data that should be included in the Data packet of the transaction to be handled by the host controller. The transfer descriptor is also used by the host controller 409 to store the results of transactions. Therefore, both the processor and the host controller 409 have access to the FIFO buffer 501.

It is advantageous to arrange the buffer in a first-in first-out mode. In this way it is relatively simple to access the data stored in the buffer for both the processor as the controller.

It is advantageous to use the buffer to store both request properties and the results of the communication with the second station. By giving the processor read access to the buffer, or at least designated sections of the buffer, an efficient way of communication is established, since the same buffer is used to store the request properties and the results of the requests. Thus, the processor can access the buffer in an identical way as when it writes the request properties, only the type of access has to be changed from a write mode to a read mode.

It is advantageous to have the bus interface determine the exact polling interval based on information provided by the processor. Limiting the number of possible polling rates simplifies the task of the bus interface in handling different requests with different polling rates. For example, if the processor may choose any polling rate between 1, 2, 3, and 128 milliseconds, the bus interface maps the chosen polling rate to one of the following actual polling rates: 1, 2, 4, 8, 16, 32, 64, or 128 milliseconds, thus reducing the potential number of different polling rates to handle from 128 to 8. Preferably, the bus interface maps the polling rate chosen by the processor on the actual polling rate that is closest to the chosen polling rate.

WO 03/029996

PCT/IB02/03812

15

The table below shows a format of the transfer descriptor. The format shown consists of 8 bytes, indicated as Byte 0, Byte 1, ..., Byte 7. Every byte consists of 8 bits, enumerated as 0, 1, ..., 7.

WO 03/029996

PCT/IB02/03812

16

-	7	6	5	4	3	2	1	0	
Byte 0	Actual bytes (7:0)								
Byte 1	Completion code				Active	Toggle	Actual bytes (9:8)		
Byte 2	MaxPktSize (7:0)								
Byte 3	Endpoint number (3:0)				Last	Speed	MaxPktSize (9:8)		
Byte 4	Total bytes (7:0)								
Byte 5	0	0	Res	Res	Dir. token (1:0)		Total bytes (9:8)		
Byte 6	Device address								
Byte 7	Interrupt polling interval (4:0)								

The table below clarifies the meaning of the parameters in the table above.

Name	Length (bits)	Description
ActualBytes	10	Contains the number of bytes that were transferred
Completion code	4	Indicates how the last transaction for the involved transfer descriptor was completed, including occurrence of errors
Active	1	Indicates whether or not transactions should be executed
Toggle	1	Used to generate or compare the data PID value. It is updated after each successful transmission or reception of a Data packet
MaxPktSize	10	Indicates the maximum number of bytes that can be sent to or received from the involved endpoint in a single Data packet
Res	1	not used
Speed	1	Indicates the speed of the involved endpoint
TotalBytes	10	Indicates the total number of bytes to be transferred
Dir. token	2	Indicates if the involved transfer descriptor involves IN transfers or OUT transfers
Device address	7	Indicates the address of the USB device that should be addressed
Interrupt polling interval	4	Indicates the polling interval of the interrupt transfers

WO 03/029996

PCT/IB02/03812

17

CLAIMS:

1. A bus system comprising a first station (202) and a second station (203, 204) coupled by a bus for transferring data and control signals, said bus being conceived to operate according to a protocol in which said first station (202) repeatedly sends requests to said second station (203, 204), characterized in that said first station (202) comprises a processor
5 (206), a bus interface (207), and a buffer (208) coupled to said processor (206) and said bus interface (207), where
 - said processor (206) is operable to generate request properties for said requests and for handling said data items;
 - said buffer (208) is operable to store said request properties;
 - 10 - said bus interface (207) is operable to generate said requests from said stored request properties and to repeatedly send said requests.
2. A bus system as claimed in claim 1, characterized in that said request
15 properties comprise a polling rate of said requests.
3. A bus system as claimed in claim 2, characterized in that said request
properties also comprise a payload of said requests.
4. A bus system as claimed in claim 1, characterized in that said second station
20 (203, 204) responds to said requests, that said processor (206) is an interruptible processor and that said bus interface (207) is operable for interrupting said processor (206) upon reception of selected responses of said second station (203, 204).
5. A bus system as claimed in claim 1, characterized in that said bus system is a
25 USB system.
6. A bus interface comprising a connection for a bus (403) and an input for receiving request properties from a processor, characterized in that said bus interface comprises a buffer (411) operable to store said request properties and a controller (409)

WO 03/029996

PCT/IB02/03812

18

operable to generate requests out of the stored request properties and to repeatedly send said requests to the connection (403).

7. A bus interface as claimed in claim 6, characterized in that said request
5 properties comprise a polling rate of said requests.
8. A bus interface as claimed in claim 7, characterized in that said request
properties also comprise a payload of said requests.
- 10 9. A bus interface as claimed in claim 6, characterized in that said bus interface
also comprises an interrupt output for sending an interrupt signal to an interruptible
processor, and that said controller (409) is operable to receive responses from said connection
and to interrupt said processor upon reception of selected responses by sending an interrupt
signal to the interrupt output.
- 15 10. A bus interface as claimed in claim 6, characterized in that said bus interface is
operable to operate as a USB interface.

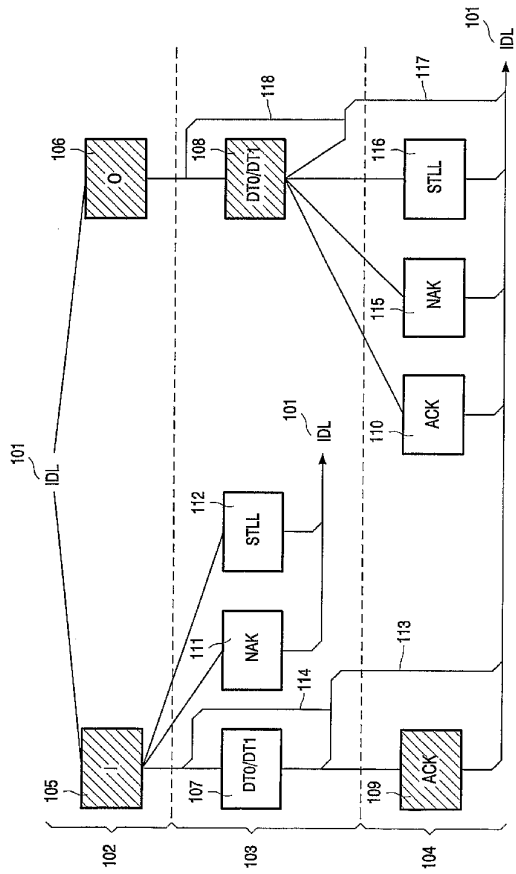


FIG. 1

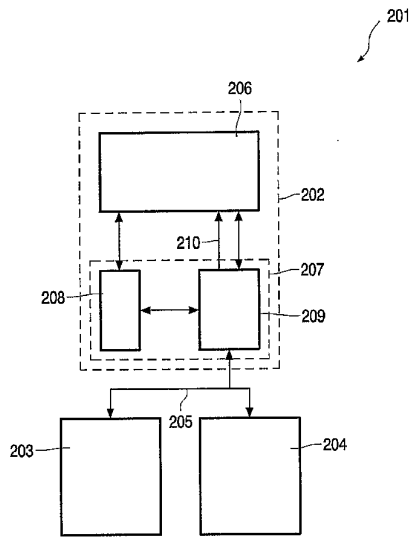


FIG. 2

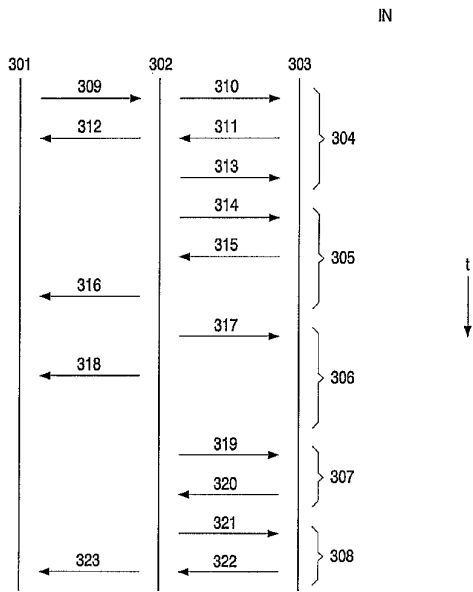


FIG. 3A

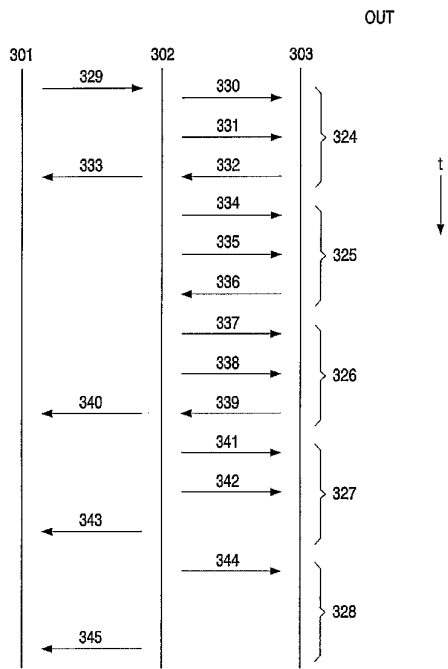


FIG. 3B

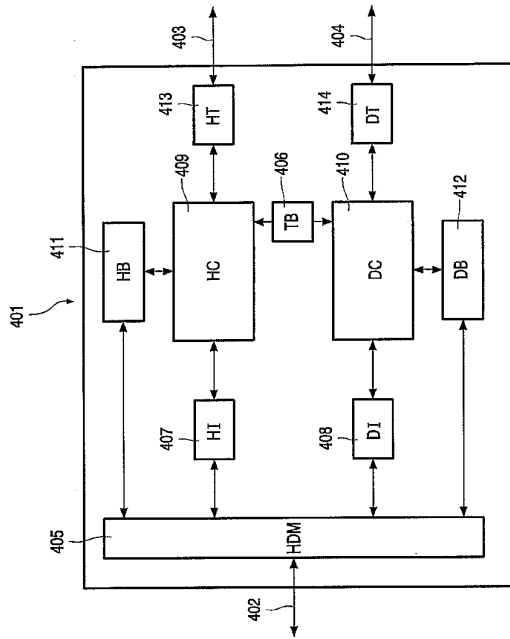


FIG. 4

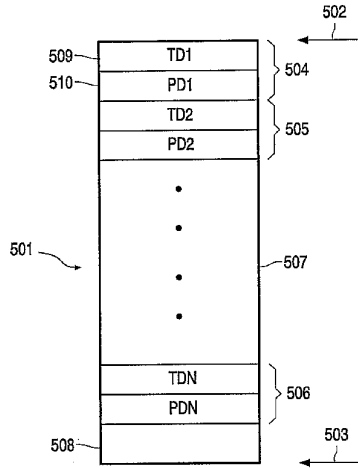


FIG. 5

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		Internat'l Application No
A. CLASSIFICATION OF SUBJECT MATTER IPC 7 606F13/24 606F13/42		PCT/IB 02/03812
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) IPC 7 606F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used) EPO-internal, WPI Data, PAJ, INSPEC, IBM-TDB, COMPENDEX		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 6 145 045 A (BRIEF DAVID ET AL) 7 November 2000 (2000-11-07) column 2, line 9 -column 3, line 48; figure 1	1,4-6,9, 10
Y	ANONYMOUS: "Universal Host Controller Interface (UHCI) Design Guide" 'Online! March 1996 (1996-03), INTEL CORP. XP002225237 Retrieved from the Internet: <URL: http://developer.intel.com/design/USB/UHCI 11D.htm> 'retrieved on 2002-12-17! page 5 -page 6	1,4-6,9, 10
<input type="checkbox"/> Further documents are listed in the continuation of box C. <input checked="" type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents:		
A document defining the general state of the art which is not considered to be of particular relevance		**I* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
E earlier document but published on or after the international filing date		**X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
I document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)		**Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
O document referring to an oral disclosure, use, exhibition or other means		**Z* document member of the same patent family
P document published prior to the international filing date but later than the priority date claimed		
Date of the actual completion of the international search 17 December 2002		Date of mailing of the international search report 13/01/2003
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentstrasse 2 NL - 2220 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016		Authorized officer Henneman, P

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No.
PCT/IB 02/03812

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6145045 A	07-11-2000	DE 19900331 A1 US 6173355 B1	08-07-1999 09-01-2001

フロントページの続き

(72)発明者 チャン, ウァー, ケイ
オランダ国 5 6 5 6 アー アー アインドーフエン プロフホルストラーン 6

(72)発明者 シャンク, シュニユ
オランダ国 5 6 5 6 アー アー アインドーフエン プロフホルストラーン 6

Fターム(参考) 5B077 BA02 FF05 NN02
5K032 CA01 DB20 DB22

【要約の続き】

・インターフェース(207)は、当該格納されたリクエスト特性から当該リクエストを生成し、かつ当該リクエストを繰り返し送信するために動作可能である。

【選択図】 図2