

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4837659号
(P4837659)

(45) 発行日 平成23年12月14日(2011.12.14)

(24) 登録日 平成23年10月7日(2011.10.7)

(51) Int.Cl. F I
G O 6 F 13/38 (2006.01) G O 6 F 13/38 3 5 0

請求項の数 5 (全 10 頁)

(21) 出願番号	特願2007-516102 (P2007-516102)	(73) 特許権者	306043703
(86) (22) 出願日	平成17年6月9日(2005.6.9)		エヌエックスピー ビー ヴィ
(65) 公表番号	特表2008-502976 (P2008-502976A)		N X P B. V.
(43) 公表日	平成20年1月31日(2008.1.31)		オランダ国 5 6 5 6 エイジー アイ
(86) 国際出願番号	PCT/IB2005/051888		ドーフエン ハイ テク キャンパス 6
(87) 国際公開番号	W02005/125093		O
(87) 国際公開日	平成17年12月29日(2005.12.29)	(74) 代理人	100075812
審査請求日	平成20年6月9日(2008.6.9)		弁理士 吉武 賢次
(31) 優先権主張番号	04102725.1	(74) 代理人	100088889
(32) 優先日	平成16年6月15日(2004.6.15)		弁理士 橋谷 英俊
(33) 優先権主張国	欧州特許庁 (EP)	(74) 代理人	100107582
			弁理士 関根 毅
		(74) 代理人	100112793
			弁理士 高橋 佳大

最終頁に続く

(54) 【発明の名称】 分割トランザクションを処理するためのバス・コントローラ

(57) 【特許請求の範囲】

【請求項 1】

トランザクション内のデータを転送するためのホスト・コントローラであって、各トランザクションはホスト・コントローラのメモリ内に保持される転送ディスクリプタによって記述され、前記トランザクションは分割トランザクションを含み、分割トランザクションのための前記転送ディスクリプタは、前記分割トランザクションが開始分割トランザクションが完了分割トランザクションを示すように設定されてよいビットを含み、分割トランザクションを備えたトランザクションが第1分割トランザクションによって開始された後は、それに続く分割トランザクションが、該トランザクションが完了するまで、ホスト・マイクロプロセッサによる介入なしで前記ホスト・コントローラにより自動的に生成される、ホスト・コントローラ。

【請求項 2】

バルク分割トランザクションの場合、第1データは、転送ディスクリプタによって記述された第1分割トランザクションに従って転送され、前記転送ディスクリプタは、前記第1データが確認通知された場合、第2分割トランザクションを定義するように変更される、請求項1に記載のホスト・コントローラ。

【請求項 3】

前記ホスト・マイクロプロセッサとシステム・メモリを相互接続するメモリ・バスへの接続のための第1インターフェースと、
内部メモリと、

外部バスへの接続のための第2インターフェースとをさらに備える、請求項1または2に記載のホスト・コントローラ。

【請求項4】

トランザクション内のデータを転送するためのホスト・コントローラの動作の方法であって、各トランザクションは転送ディスクリプタによって記述され、前記トランザクションは分割トランザクションを含み、

分割トランザクションを記述するために転送ディスクリプタを使用し、前記転送ディスクリプタは前記ホスト・コントローラのメモリ内に保持されており、前記分割トランザクションが開始分割トランザクションか完了分割トランザクションかを示すために前記転送ディスクリプタ内にビットを設定すること、および

分割トランザクションを備えたトランザクションが第1分割トランザクションによって開始された後は、それに続く分割トランザクションを、該トランザクションが完了されるまで、ホスト・マイクロプロセッサによる介入なしで前記ホスト・コントローラにより自動的に生成することを備える、方法。

【請求項5】

バルク分割トランザクションの場合、第1データは転送ディスクリプタによって記述された第1分割トランザクションに従って転送され、前記転送ディスクリプタは、前記第1データが確認通知された場合に第2分割トランザクションを定義するように変更される、請求項4に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、バス・コントローラに関し、特に、外部バスを使用して他の電子機器へのおよび他の電子機器からのデータの転送を制御するために電子機器内に含まれることができる装置に関する。

【背景技術】

【0002】

電子機器が、ユニバーサル・シリアル・バス(USB)を使用したデータの送信を可能にするインターフェースを備えることは普通になってきている。

【0003】

電子機器のアイテムがUSBシステムを使用して相互接続されている場合、機器の1つのアイテムがUSBホストとして指定され、その他のアイテムはUSB装置として指定される。USBを介した通信の開始およびスケジュールを行うのはUSBホストである。たとえば、USBホストはパーソナル・コンピュータ(PC)でよく、プリンタ、デジタル・カメラ、および携帯情報端末(PDA)など様々なUSB装置に接続されてよい。

【0004】

しかし、たとえば、PCを通じた接続を必要とせず、カメラを直接プリンタに接続するためにUSB接続を使用することも可能である。USBホストとして機能することができるためには、この例ではカメラでよい機器のアイテムは、必要な機能を備えていなければならない。本発明は、さらに特に、この機能を機器のアイテムに与えるために、機器のアイテムに含まれることができる集積回路の形態をした装置に関する。しかし、機器のアイテムはその他の機能を有し、そのUSB相互接続性は、その機能の一小部分にすぎないことが理解されるであろう。さらに、機器の諸アイテムに、それらが特別に強力なプロセッサを有することを必要とせずにUSBホストとして働くことができる機能を与えるためには、機器のアイテムに装置を含めることができることが望ましい。

【発明の開示】

【発明が解決しようとする課題】

【0005】

したがって、装置は、それが組み込まれるべき機器のアイテムの中央処理ユニット(CPU)にできるだけ少なく依存して動作できることが望ましい。たとえば、装置は、好ま

10

20

30

40

50

しくは、機器のアイテムのバス・システム内でスレーブとして動作し、CPUがバス・マスタとしてのままであることができるようにする。さらに、装置はCPUにできるだけ小さな処理負担をかけ、特にCPUに対する割り込み要求の数を最小限にすることが望ましい。さらに、装置は、できるだけ広く様々な機器のアイテムに組み込まれることができるように、どんな特別なオペレーティング・システムを使用するCPUにも依存すべきでない。

【課題を解決するための手段】

【0006】

本発明によれば、開始分割トランザクションおよび完了分割トランザクションがCPUによるどんな介入をも必要とせずに自動的に処理されるホスト・コントローラが提供される。特に、ペイロードの転送は単一の転送ディスクリプタを使用して処理され、データはペイロード全体が転送されてしまうまで転送されることができる。

10

【発明を実施するための最良の形態】

【0007】

図1は、USBホストとして動作する電子機器のアイテム10の関連部分のブロック概略図である。本発明は、マイクロプロセッサおよびシステム・メモリの機能的制限が、パーソナル・コンピュータ(PC)においてよりも関係がある、カメラ、セットトップ・ボックス、携帯電話、またはPDAなどの装置に特に適用可能である。しかし、本発明は、USBホストとして動作することができるどんな装置にも適用可能である。

【0008】

20

装置10が、本発明の理解には関係がないので図1には示されていない多くの特徴を有するであろうことは明らかであろう。

【0009】

装置10は、プロセッサ・コアを含むホスト・マイクロプロセッサ(CPU)20を有する。CPU20は、周辺バス32によってシステム・メモリ30に接続される。

【0010】

ホスト・コントローラ40はまた、周辺バスまたはメモリ・バス32によってホスト・マイクロプロセッサ20およびシステム・メモリ30に接続される。ホスト・コントローラ40は、USBバス42のためのインターフェースを有し、これを通して複数のUSB装置に接続されることができる。この図示された実施形態では、ホスト・コントローラ40は、USB2.0ホスト・コントローラであり、ホスト・コントローラのここに記載されていない特徴は、USB2.0仕様に明記されている通りでよい。

30

【0011】

通常通り、ホスト・コントローラ40は、適切なフォーマットでプロセッサ20によって準備されたデータを取り出し、バス・インターフェースを介してデータを送信するように適合される。USB通信には、2つのカテゴリのデータ転送、すなわち非同期転送および周期的転送がある。制御およびバルク・データは、非同期転送を使用して送信され、アイソクロナスおよび割り込みデータは、周期的転送を使用して送信される。非同期転送のためにはキュー・トランザクション・ディスクリプタ(qTD)データ構造が使用され、周期的転送のためにはアイソクロナス・トランザクション・ディスクリプタ(iTD)データ構造が使用される。

40

【0012】

プロセッサ20は適当な構造のデータを準備し、それをシステム・メモリ30に記憶し、ホスト・コントローラ40は、その場合、システム・メモリ30からデータを取り出さなければならない。

【0013】

図2は、組み込みUSBホスト・コントローラ40の構造をさらに詳細に示す。

【0014】

前述のように、ホスト・コントローラ40は、メモリ管理ユニット、スレーブDMAコントローラ、割り込み制御ユニット、およびハードウェア構成レジスタを含むインターフ

50

エース 44 に接続されるメモリ・バス 32 のための接続を有する。インターフェース 44 はまた、制御および割り込み信号のための接続 46、ならびに RAM 構造をサポートするレジスタ 48 およびホスト・コントローラ 40 のオペレーショナル・レジスタをも有する。

【0015】

インターフェース 44 は、この好ましい実施形態ではデュアル・ポート RAM であるホスト・コントローラのオン・チップ RAM 50 に接続され、データが同時にメモリに書き込まれたりメモリから読み出されたりすることができるようにするが、均等には、適当なアービタを備えたシングル・ポート RAM でもよい。メモリ 50 は、やはり USB バス 42 のためのインターフェースを含む、ホスト・コントローラ論理ユニット 52 に接続される。制御信号は、レジスタ 48 から内部バス 54 によって論理ユニット 52 に送信されることができる。

10

【0016】

図 3 は、本発明による装置の動作方法を説明するために、ホスト・コントローラ 40 上で動作するソフトウェアを部分的に示す概略図である。

【0017】

ホスト・コントローラ 40 は、一般には通常のものである、USB ドライバ・ソフトウェア 80 および USB 拡張ホスト・コントローラ・インターフェース・ソフトウェア 82 を実行する。

【0018】

ホスト・コントローラ 40 はまた、データの送信先である全てのエンドポイントのための転送ベースの転送ディスクリプタのリストを準備する USB EHCI インターフェース・ソフトウェア 84 も実行する。

20

【0019】

EHCI インターフェース・ソフトウェア 84 は、既存の周期的および非同期ヘッダのために EHCI ホスト・スタック 82 によって生成されたパラメータを使用するように書き込まれ、全ての様々な形の USB 転送、特に、高速アイソクロナス、バルク、割り込みおよび制御、ならびにスタート/ストップ分割トランザクションなどの高速 USB 転送のために使用されることができる。

【0020】

ホスト・マイクロプロセッサ 20 は、ホスト・コントローラ 40 がバス 32 をマスタすることを必要とせずに、周辺バス 32 を通して転送ベースの転送ディスクリプタをホスト・コントローラ 40 の RAM 50 に書き込む。言い換えれば、ホスト・コントローラ 40 は、スレーブとしてのみ働く。この場合、転送ベースの転送ディスクリプタは、ホスト・コントローラ 40 の RAM 50 にメモリマップされることができる。

30

【0021】

有利には、ホスト・コントローラ 40 のビルトイン・メモリ 50 は、ホスト・マイクロプロセッサ 20 にマップされ、トランザクションがホスト・マイクロプロセッサ 20 からスケジュールされることができる容易さを向上する。

【0022】

デュアル・ポート RAM 50 の使用は、1つの転送ベースの転送ディスクリプタがホスト・コントローラ 40 によって実行されている間に、ホスト・マイクロプロセッサ 20 がデータを別のブロック・スペースに書き込んでいることができることを意味する。

40

【0023】

前述のように、ホスト・コントローラ 40 は、USB 2.0 ホスト・コントローラであり、USB 2.0 仕様に定義されているハイスピード・バスを介してデータを転送することができる。しかし、ホスト・コントローラ 40 はまた、USB 仕様に定義されているように、フル・スピード装置およびロー・スピード装置にデータを転送することもできる。詳細には、データは開始分割トランザクションおよび完了分割トランザクションを使用してハブを介して送信される。その場合、ハブは、ハイスピード・バスに接続されることが

50

できない機器のアイテムに接続されてもよい。

【0024】

本発明によれば、開始分割トランザクションおよび完了分割トランザクションは、単一の転送ディスクリプタによって処理される。さらに詳細には、単一の転送ディスクリプタは、ペイロード全体の転送のために、開始分割(SS)トランザクションと完了分割(CS)トランザクションの両方を処理することができる。ソフトウェアは、SS/CS転送ディスクリプタが、ハブ・アドレス、ポート番号、装置およびエンドポイント番号を備えたフル・スピード(FS)エンドポイントまたはロー・スピード(LS)エンドポイントのためのバルク/制御/ISO/割り込み転送かどうかを示しさえすればよい。

【0025】

バルク・トランザクションの場合、ハードウェアは、開始分割トランザクションを送達し、ハブからの確認通知(ACK)を待ち、次いで、このACKを使用して最初の開始分割トランザクションを完了分割トランザクションにリセットする。アクティブ転送ディスクリプタのための次の開始分割トランザクションおよび完了分割トランザクションは、転送ディスクリプタによって指定されたデータが全て送信または受信されてしまうまで、ハードウェアによってアクティブにされる。したがって、たとえば、ペイロードのサイズが256であり、最大パケットのサイズが64である場合、ハードウェアは、転送ディスクリプタに指定されているペイロード全体の転送をCPUの介入なしに完了するために、4組のSS/CSトランザクションを生成する。

【0026】

通常通り、開始分割転送および完了分割転送は、1つのマイクロフレームに関してアイソクロナス転送ディスクリプタおよび割り込み転送ディスクリプタが完了された後で、いつでも開始することができる。SS/CSトランザクションのさらなる詳細は、USB2.0仕様、たとえばセクション11.17に示されている。

【0027】

図4は、開始分割トランザクションおよび完了分割トランザクションによるバルク・データのハイ・スピード転送のための転送ディスクリプタの構造を示し、図5(図5aおよび5bに分割されている)は、図4に示された転送ディスクリプタの詳細な定義を示す。

【0028】

したがって、開始分割バルク転送および完了分割バルク転送に関する各転送ディスクリプタは、CPUの介入なしに、4KまでのデータをFS/LS装置に転送することができる。NAKが3つ続いた場合、ハードウェアはトランザクションを中止し、CPUに対する割り込み要求を生成する。ハードウェアによるエラーおよびNAK処理のための一般条件は、USB2.0仕様11.17による。

【0029】

図6は、開始分割トランザクションおよび完了分割トランザクションによるアイソクロナス・データのハイ・スピード転送のための転送ディスクリプタの構造を示し、図7(図7a、7b、7cおよび7dに分割されている)は、図6に示された転送ディスクリプタの詳細な定義を示す。

【0030】

したがって、アイソクロナス・エンドポイントに関する単一の開始分割および完了分割転送ディスクリプタは、装置に対する1Kまでのデータを処理することができる。

【0031】

図8は、開始分割トランザクションおよび完了分割トランザクションによる割り込みデータのハイ・スピード転送のための転送ディスクリプタの構造を示す。図8に示された転送ディスクリプタの詳細な定義は、パラメータMaxPacketSizeが使用されることを除いて、図7に示されたものと同じである。MaxPacketSizeはFS/LS装置のMPSである。さらに、送受信される総バイト数は、MPSより大きくてもよい。この場合、パラメータUframeはmsでのポーリング・レートである。送信する総(Total)バイト数を示すパラメータの値は、図7に示されている1023に制限

10

20

30

40

50

されない。このパラメータはまた、バルク転送の開始分割のための1023にも制限されないことにも留意すべきである。

【0032】

転送ディスクリプタはまた、アクティブ・ビットの設定も可能にし、トランザクションが処理される特定のマイクロフレームを示す。したがって、US SS = 0000 0001、US CS = 0000 0100は、開始分割トランザクションにマイクロフレーム0で実行させ、完了分割トランザクションにマイクロフレーム2で実行させる。

【0033】

したがって、CPUの介入を必要とせずに分割トランザクションの実行を可能にする装置が記載される。

【図面の簡単な説明】

【0034】

【図1】本発明によるホスト・コントローラを組み込んだ電子機器のアイテムのブロック概略図である。

【図2】本発明によるホスト・コントローラのブロック概略図である。

【図3】図2のホスト・コントローラ内のソフトウェアの構造を示す図である。

【図4】本発明による第1転送ディスクリプタを示す図である。

【図5a】本発明による第1転送ディスクリプタを示す図である。

【図5b】本発明による第1転送ディスクリプタを示す図である。

【図6】本発明による第2転送ディスクリプタを示す図である。

【図7a】本発明による第2転送ディスクリプタを示す図である。

【図7b】本発明による第2転送ディスクリプタを示す図である。

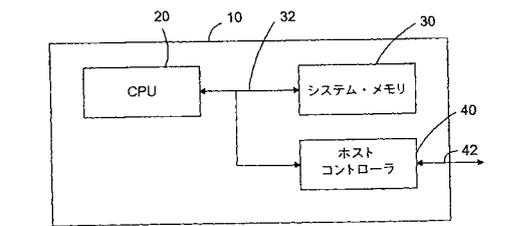
【図7c】本発明による第2転送ディスクリプタを示す図である。

【図7d】本発明による第2転送ディスクリプタを示す図である。

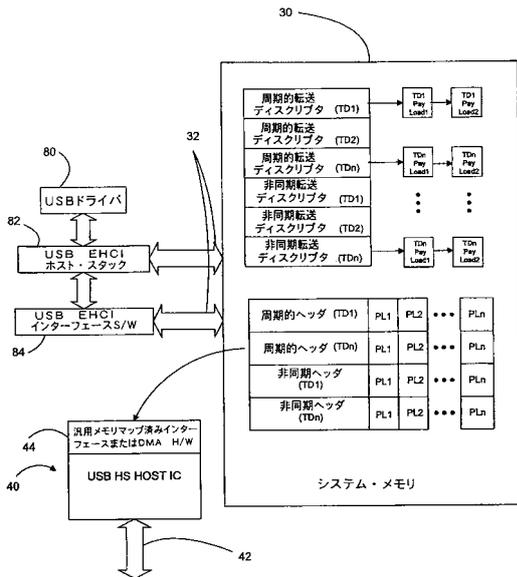
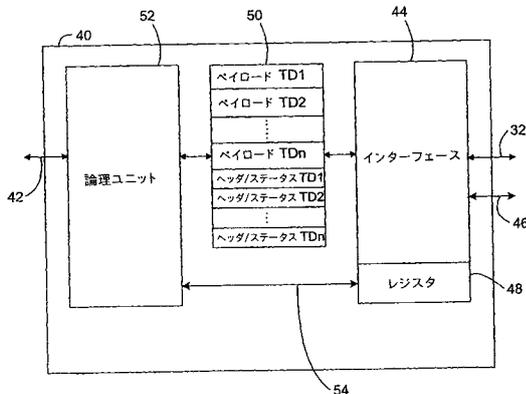
【図8】本発明による第3転送ディスクリプタを示す図である。

【図1】

【図3】



【図2】



10

20

【図7a】

ビット	名前	R/W	説明
W0			
0	V	SW/HW	1: S/Wは、ペイロードがms限度を越えても送受信されるべき場合は、1に更新。現行PTDはアクティブ。 0: このビットは、PTD全体が(usofまたはSof)を越えて実行された場合、あるいは致命的エラーが見つかった場合、非アクティブにされる。
1	X		X
2	X		X
3-14	NrBytesToTransfer (11 downto 0)	SW writes	このフィールドは、いくつかのバイトがこのデータ構造によって転送されることが示す。これはデータ・フィールドの深さを示すために使用される。このフィールドは81TDで1023、ISO FS装置のため最大許容ペイロードである。間接的にこのフィールドはダウンストリーム装置のMPSに示す。
18-28	TT_MPS_Len (10:0)	SW writes	OUT ISOのためのみ。 このフィールドは、必要総バイト数に応じて分割ごとに送信されることができる最大のバイト数を示す。全msのために送信されるべき総バイト数が188より大きい場合、このフィールドは188であるべきである。あるいは、それが188より小さい場合は、このフィールドは送信される総バイト数に等しくなければならない。
29-30	X	SW writes	N.A. X
31-34	EndPI(3:0)	SW writes	これは開始側のエンドポイントのUSBアドレス(16までのエンドポイントを1usOF/asecに入れる)。
35-41	Device Address(6:0)	SW writes	これはこのバッファが参照するエンドポイントを含む開始側のUSBアドレス。
42-43	Token(1:0)	SW writes	このトランザクションのためのトークンPID 00b: OUT 01b: IN 10b: SETUP 11b: PING(N.A.)
44-45	EPTType(1:0)	SW writes	トランザクション・タイプ 00b: 制御 01b: アイソクロナス 10b: バルク 11b: プリンク(N.A.)
46	S	SW writes	0: HSTランザクション 1: 分割ランザクション
47	X	SW	X 転送
48-49	XX SE(1:0)	SW	OUT ISO向けに送出された全てのusofのためのこのSE値は、H/Wによって自動的に処理される。
50-56	Port Number(6:0)	SW	00, N.A. INT
57-63	Hub Address(6:0)	SW	00, N.A. for INT
W0の終了			

【図7b】

W1			
0-7	Uframes (7:0)	SW writes	ISO: BIT7: 3は実行するフレームを決定 INT: BIT7: 3はMSポーリング・レート。 ポーリング・レートは、2** (b-1)usofで、bは4から16。 bが4の場合、全てのMSが実行。 B RATE Bn7:3 • 2 00001 • 4 00010/00011 • 8 00100/00101 • 16 01000/1001 UP TO 32 MS
8-23	Data Start Address(15:0)	SW writes	これはUSBバスによって/から送信/受信されるデータのための開始アドレス。 S/Wによって書き込まれたこのアドレスは、ビルトイン/メモリのアドレスNOT THE CPU ADDRESS。
24	Ex	SW	このシリアル・ビットはusofのためのISOのためのものであって、usofのためのISOが実行された場合、このビットがセット。これは、BusofのためのUSA (7:0) によって置換。 現在、これはISOでは使用されていない。
25-28	RL	SW	000にセットする。ISOにはN.A.
29	CSE		CSおよびSSが実行。このビットは同じアイソクロナス・エンドポイントに対してCSおよびSSトランザクションの両方が実行された場合にセット。CSおよびSS実行。これは、分割アイソクロナス・トランザクションがケース2aとbを区別するために使用。このビットは、BPSだけがセットされた場合にセットされることができる。これは、完了トランザクションの結果が、バック・ポインタのアクティブ・ビットがクリアされなければならないことである場合、分割トランザクションが実行されなければならないことを要する。
30	ECS		バック・ポインタ・ステータス。このビットは、sITDのために使用されるだけである。セットされた場合、このワード内のステータス情報はバック・ポインタによって参照されたデータ構造に対応。トランザクションはフレーム番号を使用して送出されるので、トランザクションはUframeおよびUframe+1によって送出されなければならない。ステータスは同じPTDの0および1のステータスが記憶されなければならない。
31	BPS (7-bit)		
32-43	NrBytesTransferred(11:0)	HW updates	このフィールドは、いくつかのバイトがこのトランザクションのために送信または受信されるを示す。
44-46	NrBytesTransferred(12:14)		MSBをPTD当たり可能な32Kに増大された転送されるNバイトに拡張。
44-50	Stybus(6:0)		ISOにはN.A.
51-54	NakCnt(3:0)		ISOにはN.A.
55-58	Cerr (1:0)		エラー・カウンタ。このフィールドはQH内のCerrフィールドに対応。このフィールドはアイソクロナス・トランザクションにはゼロフォルト。 ISOには00, N.A.

【図7c】

57	Dt		PTDを開始するようにデータ・グループ・ビットWP設定
58	P/E		0 ISOでN.A.
59	SC		0 ISOでN.A.
60	Xtr		トランザクション・エラー。これは全てのusOFごとに1ビットを有し、このビットはここではN.A.
61	B		パブリック・ビット、全てのusOFのステータスを参照する。ここではN.A.
62	H		停止ビット、全てのusOFのステータスを参照する。ここではN.A.
63	A		usOFのアクティブ・ビット、USAを参照し、ここではN.A.
W1の終了			
W2			OUT/IN INT
0-7	Usof Active (USOC) (7:0)	SW writes 0 → 1 HW writes 1 → 0 after processed	OUT/INのためにビットW1 (7:3) のフレーム番号がUSBバスのフレーム番号に一致する場合。これらのビットは、usOF向けに送信される前に1ビットごとにチェック。 例 1,1,1,1,1,1,1: INTを送信、b=1: 全ms 0,1,0,1,0,1,0: INTを送信、b=2: 0,2,4および6usof。 1,0,0,0,1,0,0,0: INTを送信、b=3: 全て4usof。 H/Wがそれを処理した後、それらはHWによってリセット。 INT TOKENは、usOFまたはまたはまたは3だけが、1だけにセットされることができる。4以上には何も送られることができない。
8-10	Status0 (2:0)	最初0にリセット	usof0でのISO IN/OUTステータス: ビット2: UnderRun (OUTトークンのみ) ビット1: バブル (INトークンのみ) ビット0: トランザクション/エラー (INおよびOUT)
11-13	Status1 (2:0)	HW	usof1でのISO IN/OUTステータス
14-16	Status2 (2:0)	HW	usof2でのISO IN/OUTステータス
17-19	Status3 (2:0)	HW	usof3でのISO IN/OUTステータス
20-22	Status4 (2:0)	HW	usof4でのISO IN/OUTステータス
23-25	Status5 (2:0)	HW	usof5でのISO IN/OUTステータス
26-28	Status6 (2:0)	HW	usof6でのISO IN/OUTステータス
29-31	Status7 (2:0)	HW	usof7でのISO IN/OUTステータス
ITD中に各usOFごとに受信されるバイト数。ISO IN			
39-32	USCS_7_0	SW writes 0 → 1 HW reset after processed	どの転送のために、全てのビットが1にセットされることができる。
39-46	Iso_IN_0	HW	アクティブが1にセットされ、フレーム番号が正しい場合、usof0中に受信されるバイト数。
47-54	Iso_IN_1	HW	アクティブが1にセットされ、フレーム番号が正しい場合、usof1中に受信されるバイト数。

【図7d】

55-63	Iso_IN_2	HW	usof2のために受信されたバイトの、ビット(7:0)。
W3	Iso_IN_3	HW	アクティブが1にセットされ、フレーム番号が正しい場合、usof3中に受信されたバイト数。
0-7	Iso_IN_4	HW	アクティブが1にセットされ、フレーム番号が正しい場合、usof4中に受信されたバイト数。
8-15	Iso_IN_5	HW	アクティブが1にセットされ、フレーム番号が正しい場合、usof5中に受信されたバイト数。
16-23	Iso_IN_6	HW	アクティブが1にセットされ、フレーム番号が正しい場合、usof6中に受信されたバイト数。
24-31	Iso_IN_7	HW	アクティブが1にセットされ、フレーム番号が正しい場合、usof7中に受信されたバイト数。
32-39	Iso_IN_8	HW	アクティブが1にセットされ、フレーム番号が正しい場合、usof8中に受信されたバイト数。
40-47	Iso_IN_9	HW	アクティブが1にセットされ、フレーム番号が正しい場合、usof9中に受信されたバイト数。

【 ☒ 〇 】

割り込みOUT/IN
開始/完了分割

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

X		最大パケット長										NBytesToTransfer										X	X	V	W0 (31-0)																	
BF	C	RL	no ping for FS	Ex	データ開始アドレス										Ms Polling rate by Bit 7:3 Bit 2:0 = XXX										X	X	V	W1 (31-0)														
S	E	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	Status [2:0]	W2 (31-0)				
		use07	use06	use05	use04	use03	use02	use01	use00	use00	use01	use02	use03	use04	use05	use06	use07	use07	use06	use05	use04	use03	use02	use01	use00	use00	use01	use02	use03	use04	use05	use06	use07	use07	use06	use05	use04	use03	use02	use01	use00	W3 (31-0)
NBytesReceived_CS_IN_6		NBytesReceived_CS_IN_5										NBytesReceived_CS_IN_4										NBytesReceived_CS_IN_3																				
63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32											
Hub Address		PortNumber										XX	LL	SS	EP Type	Token	デバイスアドレス										34-31	EndPt	W0 (63-32)													
A	H	B	Xr	SC	P/	E	dt	Cter	S-Bytes for CHFSplit, no need for SJTD										S-Bytes for CHFSplit, no need for SJTD										NBytesTransferred 4K, for FSLS Speed only, Interrupt													
NBytesReceived_CS_IN_2		NBytesReceived_CS_IN_1										NBytesReceived_CS_IN_0										NBytesReceived_CS_IN_0																				
NBytesReceived_CS_IN_7		NBytesReceived_CS_IN_7										NBytesReceived_CS_IN_7										NBytesReceived_CS_IN_7																				

フロントページの続き

- (72)発明者 ヨウ、ケイ・チャン
オランダ国5656、アーアー、アインドーフエン、ケアオブ、プロフ・ホルストラーン、6
- (72)発明者 ウェン、エフ・ムー
オランダ国5656、アーアー、アインドーフエン、ケアオブ、プロフ・ホルストラーン、6

審査官 横山 佳弘

- (56)参考文献 特開2004-110619(JP,A)
国際公開第2001/008019(WO,A1)
特開2003-150534(JP,A)
国際公開第03/107199(WO,A1)
特開2004-021613(JP,A)

- (58)調査した分野(Int.Cl., DB名)
G06F 13/38
G06F 13/28