

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5227601号
(P5227601)

(45) 発行日 平成25年7月3日 (2013.7.3)

(24) 登録日 平成25年3月22日 (2013.3.22)

(51) Int.Cl.

F I

G O 6 F 13/00 (2006.01)

G O 6 F 13/00 3 O 1 K

G O 6 F 13/38 (2006.01)

G O 6 F 13/38 3 3 O Z

G O 6 F 13/00 3 O 1 J

請求項の数 3 (全 11 頁)

(21) 出願番号 特願2008-28199 (P2008-28199)
 (22) 出願日 平成20年2月7日 (2008.2.7)
 (65) 公開番号 特開2009-187405 (P2009-187405A)
 (43) 公開日 平成21年8月20日 (2009.8.20)
 審査請求日 平成22年2月18日 (2010.2.18)

(73) 特許権者 000005049
 シャープ株式会社
 大阪府大阪市阿倍野区長池町2番2号
 (74) 代理人 100075557
 弁理士 西教 圭一郎
 (74) 代理人 100072235
 弁理士 杉山 毅至
 (72) 発明者 入佐 直喜
 大阪府大阪市阿倍野区長池町2番2号
 シャープ株式会社内

審査官 木村 貴俊

最終頁に続く

(54) 【発明の名称】 制御システム

(57) 【特許請求の範囲】

【請求項 1】

システムバスを介して、データ転送を制御するマスタデバイスと、
 前記マスタデバイスからアクセス要求を受けて処理を行うターゲットデバイスとが接続
 される制御システムにおいて、

前記ターゲットデバイスは、

データ転送中にエラーが発生した場合にその原因を示すエラー識別コードを前記マ
 スタデバイスへ出力するコード出力手段を有し、

前記マスタデバイスは、

エラー発生時に、エラー識別コードに基づいてマスタデバイスが実行すべきエラー回
 復処理の内容を予め設定する設定手段と、

前記設定手段で設定された前記エラー回復処理を実行する回復手段とを有し、

前記ターゲットデバイスが出力するエラー識別コードは、エラーの発生原因毎にコード
 を出力する信号線が割り当てられており、

データ転送中にターゲットデバイスでエラーが発生すると、エラーが発生したターゲッ
 トデバイスの優先度を最下位に変更し、以降のデータ転送において、変更された優先度
 に基づいて、各マスタデバイスから各ターゲットデバイスへのコマンド発行の優先度を変更
 することを特徴とする制御システム。

【請求項 2】

前記エラー回復処理は、データ転送の続行を含むことを特徴とする請求項 1 記載の制御

10

20

システム。

【請求項 3】

前記設定手段は、データ転送方式がアイソクロナス転送のときは、エラーの発生原因がタイムアウトかまたは、データ自身に誤りがある場合は、前記エラー回復処理としてリトライを行わないように設定することを特徴とする請求項 1 記載の制御システム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、システムバスを介して、データ転送を制御するマスタデバイスと、マスタデバイスからアクセス要求を受けて処理を行うターゲットデバイスとが接続される制御システムに関する。

10

【背景技術】

【0002】

一般的に、システムバスを介してマスタデバイスとターゲットデバイスとが接続された制御システムにおいては、あるターゲットデバイスから別のターゲットデバイスへデータを転送する場合、マスタデバイスがそのデータ転送の制御を行う。

【0003】

マスタデバイスは、まずデータ転送元のターゲットデバイスに対してリードコマンドを発行する。リードコマンドを受け取った転送元ターゲットデバイスは、指定されたアドレスのリードデータを用意してマスタデバイスへと返す。マスタデバイスは、そのデータを転送先のターゲットデバイスへライトコマンドを発行してライトする。これによりターゲットデバイス間でのデータ転送を行うことができる。

20

【0004】

ここで、転送元のターゲットデバイスに接続されている外部デバイスがリードコマンドを受け取ってから、規定の応答時間内にリードデータが用意できない場合、マスタデバイスは、リードデータが用意できるまで待つか、または転送元のターゲットデバイスがタイムアウトエラーを出力してくるのを待つ必要がある。

【0005】

マスタデバイスがタイムアウトエラーを検出した場合、マスタデバイスはホストに対して割り込みを出力し、ホストは割り込み要因となるエラーの原因を確認して、そのエラーに対する回復処理を行わなければならない。

30

【0006】

このようにデータ転送中にエラーが発生すると、ホストはそのエラー処理に時間を割かれることになり、システム全体としてはスループットが下がる要因となる。

【0007】

特許文献 1 記載のバスインターフェイスは、転送データの送受信処理にエラーが生じた場合には、伝送エラー対策として、それらの処理の再試行（リトライ）を行う。

【0008】

タイムアウトエラーが発生した場合は、バスマスタは設定された回数までリトライをする方法、転送データに CRC エラーが発生した場合は、バスエラー割り込み BERR をアサートし、設定された回数までリトライを行う方法など、バスマスタ自身がエラー回復処理を行うような方法が開示されている。

40

【0009】

【特許文献 1】特開 2003 - 196228 号公報

【発明の開示】

【発明が解決しようとする課題】

【0010】

しかしながら、特許文献 1 記載のバスインターフェイスではバスマスタは、エラーに対してリトライを行う回数を設定できるが、エラーの内容に応じたエラー回復処理はできていない。

50

【 0 0 1 1 】

本発明の目的は、スループットを向上させるとともに、柔軟なエラー回復処理を行うことができる制御システムを提供することである。

【課題を解決するための手段】

【 0 0 1 2 】

本発明は、システムバスを介して、データ転送を制御するマスタデバイスと、前記マスタデバイスからアクセス要求を受けて処理を行うターゲットデバイスとが接続される制御システムにおいて、

前記ターゲットデバイスは、

データ転送中にエラーが発生した場合にその原因を示すエラー識別コードを前記マスタデバイスへ出力するコード出力手段を有し、

前記マスタデバイスは、

エラー発生時に、エラー識別コードに基づいてマスタデバイスが実行すべきエラー回復処理の内容を予め設定する設定手段と、

前記設定手段で設定された前記エラー回復処理を実行する回復手段とを有し、

前記ターゲットデバイスが出力するエラー識別コードは、エラーの発生原因毎にコードを出力する信号線が割り当てられており、

データ転送中にターゲットデバイスでエラーが発生すると、エラーが発生したターゲットデバイスの優先度を最下位に変更し、以降のデータ転送において、変更された優先度に基づいて、各マスタデバイスから各ターゲットデバイスへのコマンド発行の優先度を変更することを特徴とする制御システムである。

【 0 0 1 3 】

また本発明は、前記エラー回復処理は、データ転送の続行を含むことを特徴とする。

【 0 0 1 4 】

また本発明は、前記設定手段は、データ転送方式がアイソクロナス転送のときは、エラーの発生原因がタイムアウトかまたは、データ自身に誤りがある場合は、前記エラー回復処理としてリトライを行わないように設定することを特徴とする。

【発明の効果】

【 0 0 1 7 】

本発明によれば、ターゲットデバイスにおけるエラー発生時に、エラー発生原因ごとに予め設定されたエラー回復処理をマスタデバイスが実行することで、ホストに対してエラー割り込み通知することなく回復処理が行われるため、制御システム全体のスループットを向上させることができる。

また、エラーの発生原因毎にコードを出力する信号線が割り当てられているので、複数のエラーが同時に発生してもマスタデバイスに対し通知することができ、エラーの発生原因に応じて取りこぼすことなく、エラー回復処理を行うことができる。また、信頼性が高くないターゲットデバイスの優先順位を下げることで、信頼性の高いターゲットデバイスの帯域が向上し、システム全体のスループットを向上させることができる。

【 0 0 1 8 】

また本発明によれば、前記エラー回復処理は、データ転送の続行を含むのでアプリケーションに応じて柔軟なシステムを構築できる。

【 0 0 1 9 】

また本発明によれば、データの信頼性よりリアルタイム性が要求されるアイソクロナス転送方式を採用する場合は、タイムアウトエラーやデータ自身に誤りがある場合に、リトライを行わないようにすることでデータ転送を継続させることができる。

【 0 0 2 2 】

また、エラー発生時にホストへエラー通知するかどうかを任意に設定することで、アプリケーションに応じて、ホストへ通知させる場合と、マスタデバイスが自動回復処理させる場合とを自由に選択することができる。

【 0 0 2 3 】

また、エラーの発生によってマスタデバイスの転送順位の優先度をさげることもでき、システムの最適化を行なうことも可能となる。

【発明を実施するための最良の形態】

【0024】

図1は、本発明の実施の一形態である制御システム1の構成を示す概略図である。

制御システム1は、CPU2、マスタデバイス3、ターゲットデバイス4およびシステムバス5を含み、ターゲットデバイス4は、外部デバイス6と接続されている。

【0025】

一般に制御システム1は、複数のターゲットデバイス4を含み、それぞれのターゲットデバイスは、個別に外部デバイス6と接続して外部デバイス6とのデータ送受信を制御する。

10

【0026】

本実施形態では、3個のターゲットデバイス4（第1ターゲットデバイス4a、第2ターゲットデバイス4b、第3ターゲットデバイス4c）を含み、各ターゲットデバイス4a、4b、4cは、それぞれ第1外部デバイス6a、第2外部デバイス6b、第3外部デバイス6cと接続してデータ送受信を制御する。

【0027】

ここで外部デバイス6は、たとえば半導体メモリ、ハードディスクドライブなどの外部記憶装置、USBインターフェイスやLANインターフェイスなどの通信デバイス、または液晶ディスプレイなどの表示装置などで、接続されるモジュールによってそれぞれのバス仕様で接続される。

20

【0028】

マスタデバイス3は、システムバス5を介してのデータ転送の制御を司り、ホストであるCPU2または外部ホストからの命令を受けてターゲットデバイス4間でデータ転送を行わせる。一般的には、DMA（Direct Memory Access）転送という。ターゲットデバイス4は、マスタデバイス3からのアクセス要求をうけて外部デバイス6との各種データの送受信処理を行う。

【0029】

データ転送の一例を挙げて説明する。たとえば第1ターゲットデバイス4aから第2ターゲットデバイス4bへデータ転送を行う場合、まずマスタデバイス3は、第1ターゲットデバイス4aに対してリードコマンドを発行する。第1ターゲットデバイス4aは、第1外部デバイス6aに対して、指定されたアドレスのリードコマンドを発行する。第1外部デバイス6aは、指定されたアドレスのデータをデータバス上にのせる。第1ターゲットデバイス4aは、第1外部デバイス6aからリードデータを取り込んだらマスタデバイス3に対して出力する。

30

【0030】

次にマスタデバイス3は、第2ターゲットデバイス4bに対してライトコマンドを発行し、第1ターゲットデバイス4aから出力されたリードデータをライトデータとして出力する。第2ターゲットデバイス4bは、ライトデータを第2外部デバイス4bに出力するとともにライトコマンドを発行する。この一連の処理を予め設定されたデータサイズ分、転送するまで繰り返す。

40

【0031】

設定されたデータサイズ分のデータ転送が終了したら、マスタデバイス3は、CPU2に対し割り込み信号を出力し、データ転送処理が終了したことを通知する。データ転送において、正常に転送が行われていれば問題はないが、何らかの理由によってターゲットデバイス4の応答が遅れたり、データ自身に誤りが含まれてしまう場合がある。

【0032】

たとえばターゲットデバイス4から一定時間内に応答がないタイムアウトエラーが発生すると、マスタデバイス3は、CPU2に対して割り込みを出力してエラーが発生したことを通知するか、もしくは再度リードコマンドを発行するか、いずれかの処理を行う必要

50

がある。

【 0 0 3 3 】

また、データ転送方式がアイソクロナス転送方式のように一定の帯域のデータ転送帯域を保証しなければならない方式である場合、ターゲットデバイス4のタイムアウトエラーや、データ自身に誤りが含まれていることを示すパリティエラーが発生しても、データ転送を続行する必要があり、そのためにマスタデバイス3は、発生したエラーを無視するような制御を行う必要がある。

【 0 0 3 4 】

また、マスタデバイス3が複数存在し、競合動作している場合は、各マスタデバイス3の優先度は、図示しないアービタ回路によって調整される。通常は、各マスタデバイス3を均等にするか、またはアプリケーションによって優先度を変更するが、データ転送中にエラーが発生したターゲットデバイス4の優先度を下げることにより、エラーによるシステム障害の影響をなくすようにすることができる。

10

【 0 0 3 5 】

図2は、ターゲットデバイス4の構成を示すブロック図である。

コマンド受信部40は、マスタデバイス3からのコマンドを受信、解析して制御部41に伝達する。制御部41は、外部デバイス6とのバスインターフェースのプロトコルに合わせてコマンドを発行する。タイムアウトエラー検出部42、パリティエラー検出部43、アドレスエラー検出部44は、それぞれ、データ転送中にエラーが発生したかどうかを検出する。

20

【 0 0 3 6 】

本実施形態では、タイムアウトエラー、パリティエラー、アドレスエラーの3つのエラーの発生を検出することができる。各エラー検出部によりエラーが検出されると、エラー識別コード生成部45に通知され、ここで予め設定されたコードに変換されマスタデバイス3に対して、エラーの発生原因を示すエラー識別コードが送出される。

【 0 0 3 7 】

図3は、マスタデバイス3の構成を示すブロック図である。

コマンド発行部30は、アクセスしたいターゲットデバイス4のアドレスを出力し、リードまたはライトコマンドを発行する。データバッファ31は、ライトデータまたはリードデータを一時的に格納する。ターゲットデバイス4からのエラー識別コードは、エラー識別コード解析部32で解析され、エラーの発生原因をエラー処理制御部33へ通知する。

30

【 0 0 3 8 】

エラー処理設定部34は、エラーが発生した時にマスタデバイス3がどのようなエラー回復処理を行うのかを予めCPU2が設定する。エラー処理制御部33では、マスタデバイス3でエラーが発生し、エラー識別コードを受信すると、エラー処理設定部34に設定された、エラー識別コードに対応付けられたエラー回復処理に基づいて、制御部35へ指示する。

【 0 0 3 9 】

図4は、制御システム1のタイミングチャートである。

40

前提条件として、エラー回復処理の内容は、エラー処理設定部34によって、パリティエラーに対してはデータ転送を続行し、タイムアウトエラーに対してはリトライを行うように設定されていることとする。

【 0 0 4 0 】

タイムチャートにおいて、mから始まる信号はマスタデバイス3がドライブする信号で、sから始まる信号はターゲットデバイス4がドライブする信号である。

【 0 0 4 1 】

まずマスタデバイス3は、サイクルT1において、madrにアドレス、mcmdにリードコマンドを出力してリードを実行する。

【 0 0 4 2 】

50

ターゲットデバイス 4 は、リードデータが用意できたら、サイクル T 4 ~ T 7 において、sdata にリードデータ、sresp にデータ識別子を出力する。データ識別子には、V および E があり、V は Valid で確定データであることを示し、E は Error でデータに何らかのエラーが発生したことを示す。

【 0 0 4 3 】

図 4 に示したチャートではサイクル T 7 で識別子 E が出力されており、エラーが発生したことを示している。さらに、エラー発生原因は serrid (Error ID) で出力される。エラー識別コードは、たとえば図 5 に示すようなコード対応テーブルで設定されている。エラーの発生原因毎に serrid のビット数を割り当てる構成としている。したがって、エラー発生原因の種類が 4 種類であれば serrid は 4 ビットを用意すればよい。

10

【 0 0 4 4 】

0 bit 目が 1 であればパリティエラーが発生していることを示す。同様に 1 bit 目が 1 であればタイムアウトエラー、2 bit 目が 1 であれば C R C エラー、3 bit 目が 1 であれば例外アドレスエラーが発生していることを示す。

【 0 0 4 5 】

このようにエラー発生原因ごとに 1 つのビットを割り当てることで、複数のエラーが同時に発生しても、マスタデバイス 3 はエラー発生原因を取りこぼすことが無いので、エラーの回復処理を確実に行うことが可能となる。

【 0 0 4 6 】

ここでエラー発生原因について説明する。

20

パリティエラーは、データの誤りをチェックするパリティチェックのエラーのことで、パリティチェックはデータを適当な長さ (8 bit 程度) に区切り、その中に含まれる 0 の数または 1 の数を演算してその結果をパリティビットとしてデータとともに付加する。受信側でデータからパリティ演算をした結果と、パリティビットとを比較して一致していなければどこかのビットが化けて、エラーが発生したことが分かる。

【 0 0 4 7 】

タイムアウトエラーは、リードコマンドに対する応答が規定時間内になかったことで発生するエラーで、ターゲットデバイス 4 に接続される外部デバイス 6 の応答速度の遅延等により発生する。

【 0 0 4 8 】

30

C R C エラーは C R C (Cyclic Redundancy Check (巡回冗長検査)) のエラーで、パリティエラー同様、データの誤りによって発生する。C R C 生成多項式と呼ばれる、シフトや加算などを組み合わせた方式で計算を行なう。

【 0 0 4 9 】

パリティチェックでは、0 の数または 1 の数に基づいているので、2 つ以上のデータが同時に化けた場合、それらが相殺されて正常とみなされてしまうことがあるが、C R C ではその確率を低く抑えることができる。

【 0 0 5 0 】

例外アドレスエラーは、ターゲットデバイスに割り当てられているアドレス領域外にアクセスした場合に発生するエラーで、リードデータが無効であることを示す。

40

【 0 0 5 1 】

たとえばパリティエラーが発生したとすると、パリティエラーに対する回復処理は、予めデータ転送を続行するように設定されているので、サイクル T 8 ~ T 12 において、マスタデバイス 3 は受信したデータをそのまま別のターゲットデバイス 4 にライトコマンドを発行して転送する。

【 0 0 5 2 】

さらに、サイクル T 14 から 2 回目のデータ転送が行われるが、サイクル T 30 になっても応答できなかったためターゲットデバイス 4 はエラー識別コードとして E 2 (タイムアウトエラー) を serrid に出力する。タイムアウトエラーが発生した場合は、その回復処理としてリトライを行うように設定されているので、マスタデバイス 3 は、サイクル T 3

50

4でリードコマンドのリトライを行っている。

【0053】

図6は、制御システム1のデータ転送処理を示すフローチャートである。

まずステップS1では、デバイスの初期設定を行う。CPU2が動作させたいマスタデバイス3およびターゲットデバイス4の初期パラメータを設定する。

【0054】

次に、ステップS2ではマスタデバイス3を起動させる。CPU2は、マスタデバイス3を動作状態することで、データ転送を開始する。

【0055】

ステップS3では、データ転送中にエラーが発生したかどうかを、マスタデバイス3が判断し、エラーが発生していればステップS4で、エラー発生原因を特定して、予め設定されたエラー回復処理を実行する。エラーが発生していなければステップS5で、設定されたサイズ分のデータ転送が終了したかどうかをマスタデバイス3が判断する。データ転送が終了していれば処理を終了し、終了していなければステップS3に戻る。

【0056】

本発明の他の実施形態では、エラーが発生したターゲットデバイスへのアクセスの優先度を下げること、システムの最適化を行なうことが可能となる。

【0057】

本実施形態の制御システムの構成は、図1に示した制御システム1と異なりマスタデバイス3個（以下ではマスタデバイス3a、3b、3cとする）と、ターゲットデバイス3個とを含んで構成され、ターゲットデバイスのうち1つがタイムアウトエラーとなった場合を例に説明する。

【0058】

データ転送は、マスタデバイス3aがターゲットデバイス4aにアクセスし、同様にマスタデバイス3bがターゲットデバイス4bに、マスタデバイス3cがターゲットデバイス4cにアクセスするものとする。また、アクセスの優先順位はターゲットデバイス4a > ターゲットデバイス4b > ターゲットデバイス4cのように、ターゲット4aが最も優先順位が高く設定されている。

【0059】

図7は、アクセスの優先度を調整しない場合のタイミングチャートである。

タイミングチャートでは、マスタデバイス3a、3b、3cおよびシステムバス上の動作状態を示している。例えば“TgtA RD”は、ターゲットデバイス4aへのリードコマンドを示す。

【0060】

時刻t1で3つのマスタデバイス3a、3b、3cが同時にターゲットデバイス4a、4b、4cに対して、それぞれリードコマンドを発行し、予め定める優先順位に基づいてマスタデバイス3aによるターゲットデバイス4aへのリードコマンドが時刻t2でシステムバス上に発行される。

【0061】

ターゲットデバイス4aでは、何らかの不具合が発生し、時刻t5でターゲットデバイス4aがタイムアウトエラーを返したとすると、時刻t6でマスタデバイス3bのリードコマンドがシステムバス上に発行され、時刻t9でターゲットデバイス4bからのリードデータが返ってくると同時に、マスタデバイス3aが再度リードコマンドを発行する。

【0062】

この時点でマスタデバイス3aとマスタデバイス3cとがリードコマンドを発行しているが、優先順位に基づいてマスタデバイス3aのリードコマンドがシステムバス上に発行される。ターゲットデバイス4aの不具合が解消していない場合は、時刻t13でターゲットデバイス4aが再びタイムアウトエラーを返す。その後時刻t14では、マスタデバイス3cのリードコマンドがシステムバス上に発行され、時刻t17でターゲットデバイス4cからのリードコマンドが返される。

10

20

30

40

50

【 0 0 6 3 】

このように優先度の調整を行わない場合、タイムアウトエラーが発生したターゲットデバイス 4 a へのアクセスがシステムバスを占有することとなり、無駄な帯域を消費してシステム全体の処理を遅延させることになる。

【 0 0 6 4 】

これに対して本実施形態では、優先度の調整を行うことで、システム全体としての処理の最適化を行う。

【 0 0 6 5 】

図 8 は、アクセスの優先度を調整する場合のタイミングチャートである。

優先度調整を行う場合、時刻 t 9 までの動作は、優先度調整の有無にかかわらず前述の通りであるが、時刻 t 5 でタイムアウトエラーが発生した時点で、優先順位がターゲットデバイス 4 b > ターゲットデバイス 4 c > ターゲットデバイス 4 a のように変更され、ターゲットデバイス 4 a の優先順位が最下位になる。

【 0 0 6 6 】

したがって、時刻 t 1 0 では変更された優先順位に基づいて、マスタデバイス 3 a ではなく、マスタデバイス 3 b のリードコマンドがシステムバス上に発行される。

【 0 0 6 7 】

同様に時刻 t 1 4 でもマスタデバイス 3 c のリードコマンドがシステムバス上に発行される。

【 0 0 6 8 】

そして、優先順位の最も低いターゲットデバイス 4 a へのマスタデバイス 3 a のリードコマンドは、他のマスタデバイス 3 b、3 c のリードコマンドが発行されていない時刻 t 1 8 で発行されることになる。

【 0 0 6 9 】

このようにタイムアウトエラーなどのエラーが発生したターゲットデバイスへのアクセスの優先度を下げることにより、他のターゲットデバイスのコマンドが優先的に処理され、システムバスの帯域を有効利用してシステム全体としての処理の最適化を行うことができる。

【 0 0 7 0 】

なお、ターゲットデバイスの優先順位は、CPU 2 の記憶領域や、他の記憶装置などに予め記憶しておき、ターゲットデバイスのエラーが発生した時点で優先順位を更新するようにすればよい。

【図面の簡単な説明】

【 0 0 7 1 】

【図 1】本発明の実施の一形態である制御システム 1 の構成を示す概略図である。

【図 2】ターゲットデバイス 4 の構成を示すブロック図である。

【図 3】マスタデバイス 3 の構成を示すブロック図である。

【図 4】制御システム 1 のタイミングチャートである。

【図 5】エラー識別コードと発生原因との対応を示すコード対応テーブルの例を示す図である。

【図 6】制御システム 1 のデータ転送処理を示すフローチャートである。

【図 7】アクセスの優先度を調整しない場合のタイミングチャートである。

【図 8】アクセスの優先度を調整する場合のタイミングチャートである。

【符号の説明】

【 0 0 7 2 】

- 1 制御システム
- 2 CPU
- 3 マスタデバイス
- 4 ターゲットデバイス
- 5 システムバス

10

20

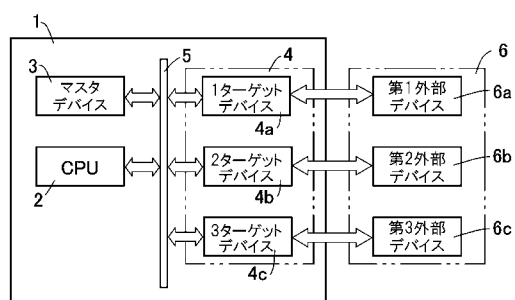
30

40

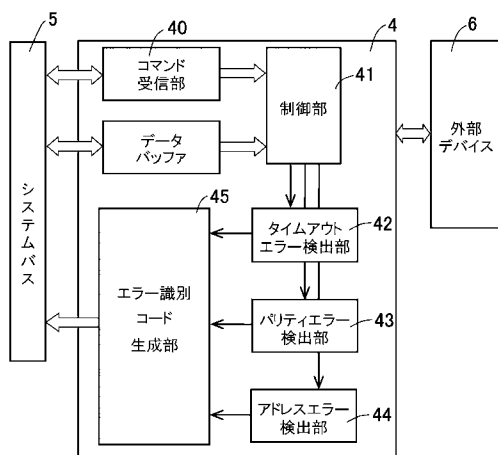
50

6 外部デバイス

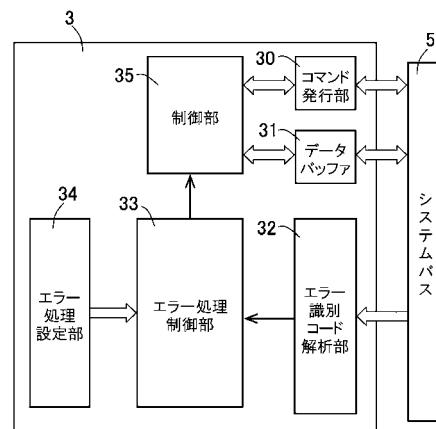
【図1】



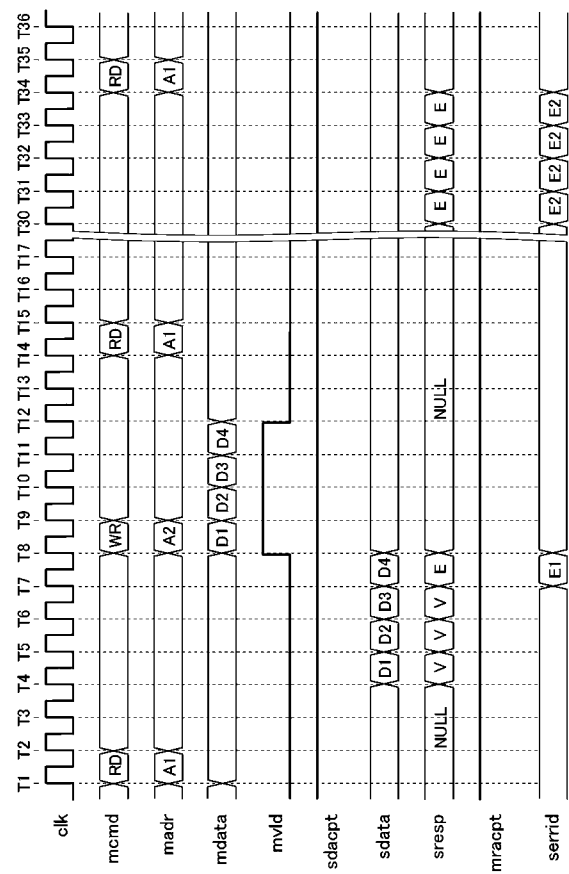
【図2】



【図3】



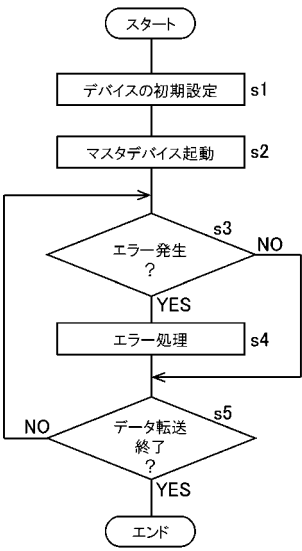
【図 4】



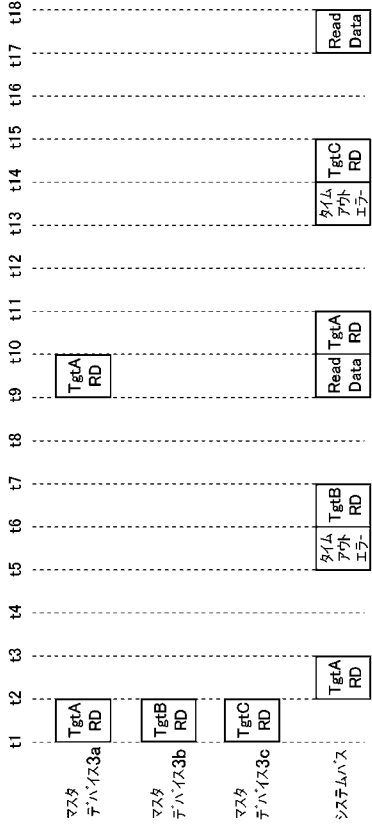
【図 5】

serrid[3:0]	エラー内容
0000	Null
xxx1	パリティエラー
xx1x	タイムアウトエラー
x1xx	CRCエラー
1xxx	例外アドレスエラー

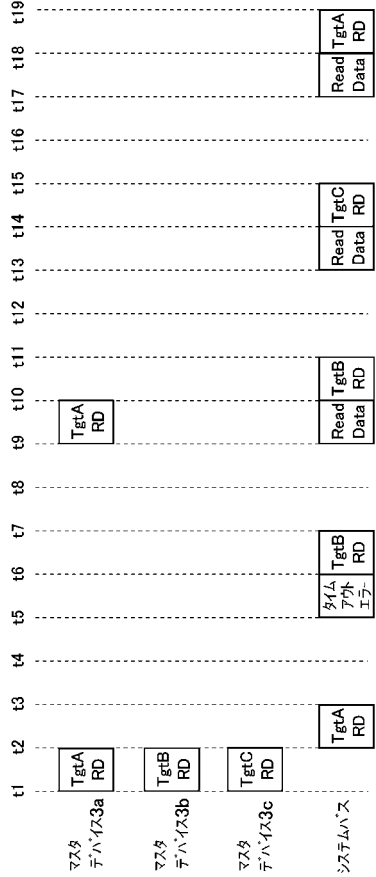
【図 6】



【図 7】



【図 8】



フロントページの続き

- (56)参考文献 特開平03-051937(JP,A)
特開平11-331399(JP,A)
特開昭60-204050(JP,A)
特開2003-150356(JP,A)
特開2003-218878(JP,A)
特開平11-017855(JP,A)
特開2003-131865(JP,A)
特開平02-143630(JP,A)
特開平05-314674(JP,A)

- (58)調査した分野(Int.Cl., DB名)

G06F 13/00 - 13/42