

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5377275号
(P5377275)

(45) 発行日 平成25年12月25日(2013.12.25)

(24) 登録日 平成25年10月4日(2013.10.4)

(51) Int.Cl.		F I			
HO4L	7/02	(2006.01)	HO4L	7/02	Z
HO4L	7/04	(2006.01)	HO4L	7/04	B

請求項の数 16 (全 29 頁)

(21) 出願番号	特願2009-295617 (P2009-295617)	(73) 特許権者	000001007
(22) 出願日	平成21年12月25日(2009.12.25)		キヤノン株式会社
(65) 公開番号	特開2011-135531 (P2011-135531A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成23年7月7日(2011.7.7)	(74) 代理人	100126240
審査請求日	平成24年12月20日(2012.12.20)		弁理士 阿部 琢磨
		(74) 代理人	100124442
			弁理士 黒岩 創吾
		(72) 発明者	桃井 昭好
			東京都大田区下丸子3丁目30番2号キヤノン株式会社内
		(72) 発明者	森下 浩一
			東京都大田区下丸子3丁目30番2号キヤノン株式会社内
		審査官	白井 亮

最終頁に続く

(54) 【発明の名称】 情報処理装置又は情報処理方法

(57) 【特許請求の範囲】

【請求項1】

外部デバイスに対してクロックを供給する供給手段と、
前記外部デバイスに対してキャリブレーションパターンを送信するように指示する指示手段と、

前記クロックと同期させて前記外部デバイスにより出力される外部デバイスデータを受信する受信手段と、

前記クロックをどのタイミングで停止させるかを示すゲーティング情報に基づいて前記供給手段による前記クロックの供給を停止させる制御手段と、

前記受信手段の受信することになるキャリブレーションパターンとサイクル遅延量との関係を示す遅延情報を保持する記憶手段と、

前記制御手段を機能させた状態で前記受信手段の受信したキャリブレーションパターンと前記遅延情報とを比較することでサイクル単位の遅延量を検出する検出手段と、

前記サイクル遅延量に基づいて前記受信手段のデータ取り込みタイミングを遅延させる遅延手段と

を有することを特徴とする情報処理装置。

【請求項2】

前記供給手段の供給するクロックを発振する発振手段と、

前記受信手段の受信するデータを前記発振手段の発振するクロックの1サイクルより小さい量だけ遅延させて位相ズレを補正する第1遅延手段と、

10

20

前記受信手段を制御する制御信号を前記発振手段の発振するクロックの1サイクルの整数倍だけ遅延させて前記受信手段に入力させることで、前記受信手段における周期単位のデータ取り込みタイミングのズレを補正する第2遅延手段とを更に有することを有することを特徴とする請求項1に記載の情報処理装置。

【請求項3】

前記第2遅延手段は入出力が直列に接続されている複数のフリップフロップを有し、前記制御信号を周期単位で遅延させる場合に前記複数のフリップフロップのうちの1つの出力を選択的に出力することを特徴とする請求項2に記載の情報処理装置。

【請求項4】

前記受信手段に受信させるキャリブレーションパターンを送信するように前記外部デバイスに対して指示する指示手段と、

前記クロックをどのタイミングで停止させるかを示すゲーティング情報に基づいて前記供給手段による前記クロックの供給を停止させる第一クロック制御モードと、前記クロックの供給を停止させない第二クロック制御モードとを選択的に機能させる選択手段とを有し、

前記選択手段は、前記キャリブレーションパターンの位相ズレを調節する場合に前記第二クロック制御モードを機能させ、前記キャリブレーションパターンのサイクル遅延を調節する場合に前記第一クロック制御モードを機能させることを特徴とする請求項1乃至3のいずれか1項に記載の情報処理装置。

【請求項5】

受信するキャリブレーションパターンの期待値を保持する記憶手段と、

前記選択手段によって前記制御手段を機能させた状態で前記受信手段の受信したキャリブレーションパターンと前記記憶手段の保持する期待値とを比較して一致するかを判定する判定手段と

を更に有することを特徴とする請求項4に記載の情報処理装置。

【請求項6】

前記判定手段は前記受信手段の受信したキャリブレーションパターンと前記期待値とが一致することを判定し、前記受信手段によるデータの取り込みタイミングの調整が完了したことを判断することを特徴とする請求項5に記載の情報処理装置。

【請求項7】

前記キャリブレーションパターンは複数の値によって構成され、当該複数の値のうち連続する所定数の値の夫々が直前の値と異なっていることを特徴とする請求項4乃至6のいずれか1項に記載の情報処理装置。

【請求項8】

前記所定数は前記情報処理装置と前記外部デバイスとの間で発生し得る遅延量に対応するサイクル数以上であることを特徴とする請求項7に記載の情報処理装置。

【請求項9】

前記受信手段の受信したデータを保持し、自身がデータの保持をできない場合に通知する保持手段を更に有し、

前記選択手段は、前記キャリブレーションパターンに基づくデータの取り込みタイミングの調節が完了した場合に、前記保持手段からの通知を受けて前記供給手段による前記クロックの供給を停止させる第三クロック制御モードを機能させることを特徴とする請求項4乃至8のいずれか1項に記載の情報処理装置。

【請求項10】

前記保持手段はFIFO構造であり、前記通知がFIFOの残量情報を示す信号に基づいていることを特徴とする請求項9に記載の情報処理装置。

【請求項11】

前記外部デバイスとの通信の種別を判断する識別手段を更に有し、前記外部デバイスが所定の種別でないと判断できる場合に、前記選択手段は前記受信手段のデータ取り込みタイミングを調節する際に前記第三クロック制御モードを機能させることを特徴とする請求

10

20

30

40

50

項 9 又は 10 に記載の情報処理装置。

【請求項 12】

前記ゲーティング情報は、周期的なタイミングで前記供給手段がクロックを停止することを示す情報を有することを特徴とする請求項 1 乃至 11 のいずれか 1 項に記載の情報処理装置。

【請求項 13】

前記ゲーティング情報は、前記制御手段によって前記クロックの停止し始めるタイミング、前記クロックの停止を解除させるタイミング、クロックを停止させる周期の少なくとも何れか 1 つを示すことを特徴とする請求項 1 乃至 12 のいずれか 1 項に記載の情報処理装置。

10

【請求項 14】

前記外部デバイスとの通信の種別を判断する識別手段を更に有し、前記外部デバイスが所定の種別であると判断できる場合に、前記選択手段は前記受信手段のデータ取り込みタイミングを調節する際に前記第一クロック制御モードを機能させることを特徴とする請求項 4 乃至 13 のいずれか 1 項に記載の情報処理装置。

【請求項 15】

前記制御手段が、少なくとも前記受信手段の受信しているキャリブレーションパターンの値が変動するタイミングの 1 つで前記クロックを停止し始めることを特徴とする請求項 1 乃至 14 のいずれか 1 項に記載の情報処理装置。

【請求項 16】

前記制御手段が、少なくとも前記受信手段の受信しているキャリブレーションパターンの値が変動するタイミングの 1 つで前記クロックの停止を解除することを特徴とする請求項 1 乃至 15 のいずれか 1 項に記載の情報処理装置。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、外部デバイスコントローラなどの情報処理装置が外部デバイスにクロックを供給し、外部デバイスがこのクロックに同期して出力する外部デバイスデータを取り込む情報処理装置又は情報処理方法に関する。

【背景技術】

30

【0002】

特許文献 1 に開示されているように、外部デバイスコントローラを有する情報処理装置について、外部デバイスが情報処理装置に通信可能に接続されている場合に、情報処理装置の外部デバイスコントローラから外部デバイスへ動作クロックを供給する技術がある。ここでは、外部デバイスコントローラの供給したクロックに同期して外部デバイスがデータを出力するように設定しておき、外部デバイスコントローラが外部デバイスの出力したデータを取り込むように構成することが一般的である。

【0003】

この方法を用いると、外部デバイスコントローラは、外部デバイスへのクロックの供給を一時的に停止する（クロックゲーティングに相当する）ことで、外部デバイスから外部デバイスコントローラへのデータの供給を一時的に停止する事ができる。例えば、外部デバイスコントローラ内の受信バッファの許容容量までデータが蓄積された場合、外部デバイスコントローラはクロックの供給を停止し、データの供給を停止できるので、バッファの容量が小さくても好適にバッファのオーバーフローを抑制できる。

40

【先行技術文献】

【特許文献】

【0004】

【特許文献 1】特開昭 59 - 173839

【発明の概要】

【発明が解決しようとする課題】

50

【0005】

外部デバイスから情報処理装置の受信するデータが、情報処理装置のクロックに対して1サイクル以上遅延する場合、情報処理装置側でデータを取り込むタイミングを調節する処理が増加してしまう。

【課題を解決するための手段】

【0006】

以上の課題を解決するために本発明に係る情報処理装置は、

外部デバイスに対してクロックを供給する供給手段と、

前記クロックと同期させて前記外部デバイスにより出力されるデータを受信する受信手段と、前記受信手段の受信することになるデータとサイクル遅延量との関係を示す遅延情報

10

を保持する記憶手段と、
前記受信手段の受信したデータと前記遅延情報とを比較することでサイクル遅延量を検出する検出手段と、

を有することを特徴とする。

【発明の効果】

【0007】

効率的にキャリブレーションを実施することでキャリブレーションに要する時間の増加を抑制できる。

【図面の簡単な説明】

【0008】

20

【図1】本発明の一実施例における外部デバイスコントローラの構成図である。

【図2】参考例のスキュー制御部の構成図とタイミングチャートを示す。

【図3】サイクル制御部の構成図とタイミングチャートである。

【図4】キャリブレーション処理のフローチャートと出力クロック制御部の構成図である。

。

【図5】キャリブレーションパターン受信中にゲーティングパターンに基づいて出力クロックをゲーティングしているタイミングチャートである。

【図6】サイクル指標と受信されるキャリブレーションパターンの相関を示すタイミングチャートである。

【図7】サイクル指標と受信されるキャリブレーションパターンの相関を示すタイミング

30

チャートである。

【図8】ゲーティングサイクル数と受信されるキャリブレーションパターンの相関を示すタイミングチャートである。

【図9】ゲーティングサイクル数と受信されるキャリブレーションパターンの相関を示すタイミングチャートである。

【図10】ゲーティング間のサイクル数と受信されるキャリブレーションパターンの相関を示すタイミングチャートである。

【図11】ゲーティング間のサイクル数と受信されるキャリブレーションパターンの相関を示すタイミングチャートである。

40

【図12】ゲーティング間のサイクル数と受信されるキャリブレーションパターンの相関を示すタイミングチャートである。

【図13】ゲーティング間のサイクル数と受信されるキャリブレーションパターンの相関を示すタイミングチャートである。

【図14】ゲーティング開始位置・ゲーティング終了位置と受信されるキャリブレーションパターンの相関を示すタイミングチャートである。

【図15】ゲーティング開始位置・ゲーティング終了位置と受信されるキャリブレーションパターンの相関を示すタイミングチャートである。

【図16】ゲーティング開始位置・ゲーティング終了位置と受信されるキャリブレーション

50

ンパターンの相関を示すタイミングチャートである。

【図18】正しいサイクル設定を用いた場合の外部デバイスコントローラと外部デバイスの扱う信号のタイミングチャートである。

【図19】外部デバイスコントローラを有するシステム構成の概略図である。

【図20】実際に受信されるキャリブレーションパターンと実際のサイクル数との差の関係の例を示す。

【図21】参考例の外部デバイスコントローラの構成とコマンドやデータのフォーマットを示すタイミングチャートを示す。

【図22】参考例のキャリブレーション処理のフローチャートである。

【図23】参考例のキャリブレーション処理のタイミングチャートである。

【図24】参考例の外部デバイスコントローラが扱う信号のタイミングチャートである。

【発明を実施するための形態】

【0009】

図面を参照して本発明の実施例を以下に説明する。まず、外部デバイスコントローラが外部デバイスから受信するデータについて、外部デバイスコントローラの出力クロックに対して1サイクル以上遅延する際の課題解決のための実施例を以下に説明する。

【0010】

図19は外部デバイス107と通信を行う情報処理装置とを備えたシステムのブロック図である。外部デバイス107と通信を行う機能を有するLSIであるASIC100は、CPU101、DRAMコントローラ104、DMAコントローラ102、外部デバイスコントローラ103とCLOCKジェネレータ105を有する。(ASICはApplication Specific Integrated Circuitの略称で、DMAはDirect Memory Accessの略称。)また、発振手段としてのCLOCKジェネレータ105はCPU101、DMAコントローラ102、外部デバイスコントローラ103及びDRAMコントローラ104の用いるクロック(cpu_clock113、dmac_clock114、host_clock115、dram_clock116)を生成して供給する。CPU101はCPU I/F110を介して、外部デバイスコントローラ103とDMAコントローラ102とDRAMコントローラ104へレジスタアクセスを行う。DMAコントローラ102はCPU I/F110を介して、DRAMコントローラ104との間でデータ転送を行う。DRAMコントローラ104はDRAM I/F117を介してDRAM106との間でデータ転送を行う。外部デバイスコントローラ103はDMA I/F111を介して、DMAコントローラ102との間でデータ転送を行う。また、外部デバイスコントローラ103は外部デバイスI/F112を介して、外部デバイス107との間でデータ転送を行う。

【0011】

次に、外部デバイスコントローラについて説明する。図1は本発明に係る一実施例における外部デバイスコントローラ103の構成を示す。

【0012】

外部デバイスコントローラ103はCLOCKジェネレータ105からホストクロック115(図中、host_clock)を受け取る。ホストクロック115は外部デバイスコントローラ103内の各ブロックに接続されており、外部デバイスコントローラ103の各ブロックは、このホストクロック115に同期して動作する。

【0013】

CPU I/F制御部201は、CPU101からのカードへ送信するデータやコマンド、レジスタアクセスを受付ける一方で、CPU101へカードから受信したコマンドやデータを送信する。DMA I/F制御部204は、外部デバイス107へ送信するデータをDMAコントローラ102から受け、一方で外部デバイス107から受信するデータをDMAコントローラ102へ送信する。

【0014】

外部デバイスコントローラ103は、送信コマンドパラレルシリアル変換部125、受

10

20

30

40

50

信コマンドシリアルパラレル変換部126を介して外部デバイス107とコマンドをやりとりする。まず、CPU I/F制御部201はCPU I/F110を介してCPU101から受け取ったパラレル形式の送信コマンド223(s_cmd)を送信コマンドパラレルシリアル変換部125(以降、送信コマンドPS変換部)に送信する。送信コマンドPS変換部125は、受け取ったパラレル形式の送信コマンド223をシリアル形式の送信コマンド224(s_cmd_data)に変換して、外部デバイス107に送信する。

【0015】

外部デバイス107は受け取ったシリアル形式の送信コマンド224のデコードを行い、送信コマンドを検知する。さらに、外部デバイス107は送信コマンドの検知結果を示す検知情報をシリアル形式の受信コマンド226(r_cmd_data)として、外部デバイスコントローラ103に送信する。

10

【0016】

受信コマンドシリアルパラレル変換部126(以降、受信コマンドSP変換部)は外部デバイス107の出力するシリアル形式の受信コマンド226を受信し、パラレル形式の受信コマンド225(r_cmd_reg)に変換してCPU I/F制御部201、CPU I/F110を介してCPU101へ送信する。

【0017】

また、外部デバイスコントローラ103は、送信バッファ207、送信データパラレルシリアル変換部208、スキュー制御部211、受信データシリアルパラレル変換部210、受信バッファ209を介して外部デバイス107とデータをやりとりする。また、出力クロック244をデアサート(de-assert)して外部デバイス107からのデータを停止させるために、出力クロック制御部213と出力クロックゲーティング部214を有している。さらに外部デバイスコントローラ103は有効なサイクル調整を行うために、ゲーティングモード設定レジスタ605とゲーティングパターン設定レジスタ606を有している。

20

【0018】

また、外部デバイスコントローラ103はスキュー調整(補正)をするために、スキュー制御部211とスキュー設定レジスタ212とを有する。ここで、スキュー調整(補正)とは、受信データシリアルパラレル変換部210(以降、受信データSP変換部)に入力されるデータ又は外部デバイスコントローラ103(スキュー制御部211)のデータ取り込みタイミング(ラッチタイミング)を(ホストクロック115に対して)1サイクル以内の遅延によって調整(補正)することを示す。

30

【0019】

また、外部デバイスコントローラ103はサイクル調整(補正)をするために、サイクル設定レジスタ601、サイクル演算部602、サイクル制御部603および期待値設定レジスタ604を有する。ここで、サイクル調整(補正)とは、受信データSP変換部210の制御信号である受信イネーブル信号250(rcv_en)を(ホストクロック115に対して)サイクル単位で遅延させて受信データSP変換部210のデータ取り込み停止タイミング、データ取り込み再開タイミングを調整(補正)することを示す。

40

【0020】

スキュー設定レジスタ212はCPU I/F制御部201からスキュー設定値227(skew_reg)を受け取り保持する。スキュー制御部211は、スキュー設定レジスタ212からスキュー選択値238(skew_sel)を受け取り、設定値に応じて外部デバイス107から受け取ったシリアル形式の受信データ236(d2h_data、以降、単に受信データ236と略する場合もある)を遅延させる。

【0021】

サイクル設定レジスタ601はCPU I/F制御部201からサイクル設定値256(cycle_reg)を受け取り保持する。期待値設定レジスタ604はCPU I/F制御部201から期待値設定値253(exp_reg)を受け取り保持する。期待値

50

設定値 253 の例を図 20 に示す。実際のサイクル数との差が 0 の場合に受信されるパターンは真値（外部デバイス 107 の出力したパターン）と一致するが、外部デバイスコントローラ 103 の受信するパターンはキャリブレーションを行った際のサイクル設定値と正しいサイクル設定値との差（後述するサイクル遅延量に相当する）に応じて変化する。サイクル演算部 602 はサイクル設定レジスタ 601 からサイクル設定値 255（cycle_val）と、期待値設定レジスタ 604 から期待値 254（exp_pattern）を受け取り、望ましいサイクル設定値を演算する。具体的には、受信したキャリブレーションパターンと期待値とを比較して一致するかを判定する。一致した場合の期待値に対応したサイクル設定値と、正しいサイクル設定値の差を求める。その後、求めた差を、キャリブレーションを行った際のサイクル設定値に加算することで、正しいサイクル設定値を演算する。サイクル制御部 603 は、サイクル演算部 602 から正しいサイクル選択値 249（cycle_sel）を受け取る。さらに、サイクル制御部 603 は出力クロック制御部 213 から出力クロックイネーブル信号 243（クロック制御信号）を受け取る。

10

【0022】

〔データ受信処理〕

次に、外部デバイスコントローラ 103 が外部デバイス 107 からデータ（外部デバイスデータ）を受信する処理について説明する。

【0023】

外部デバイスコントローラ 103 はデータ受信を開始する際に、まず前述したコマンド送受信処理によって、外部デバイス 107 からのデータ受信を指示するコマンドを外部デバイスコントローラ 103 は外部デバイス 107 に送信する。さらに、外部デバイス 107 はこの送信コマンドに対する返答を受信コマンドとして外部デバイスコントローラ 103 に送信し、さらにデータを送信する。

20

【0024】

データの受信は以下のように行う。まず、外部デバイス 107 から送信されたシリアル形式の受信データ 236（d2h_data）をスキュー制御部 211 が受信する。

【0025】

スキュー制御部 211 は、シリアル形式の受信データ 236 と、外部デバイスコントローラのクロック 115（host_clk）との間のスキュー調整を行う（詳細は後述する）。スキュー調整後受信データ 235 は受信データ SP 変換部 210 に入力される。

30

【0026】

受信データ SP 変換部 210 は、出力クロック制御部 213 の出力する出力クロックイネーブル信号 243 を、後述するサイクル制御部 603 によって遅延させた受信イネーブル信号 250（詳細は後述する）を受け取り可能な構成になっている。そして、受信イネーブル信号 250 がアサート（assert）されていれば、受信データ SP 変換部 210 は入力されたスキュー調整後受信データ 235 を受信し、パラレル形式の受信データ 234（r_data_buf）へ変換する。

【0027】

受信データ SP 変換部 210 は、不図示の K 段のシフトレジスタ（直列入力並列出力型フリップフロップ）を有しておりシリアル形式で受信するデータを K bit のパラレル形式のデータとして送出する構成になっている。従って、受信データ SP 変換部 210 は受信イネーブル信号がアサートされ続けていれば、K サイクルに 1 回パラレル形式のデータを送出する。ここで、受信データ SP 変換部 210 が外部デバイス 107 からのデータ（外部デバイスデータ）を取り込むタイミングとは、（外部デバイス 107 側からみて）1 段目（0 bit）のフリップフロップのラッチタイミングに相当するといえる。

40

【0028】

保持手段としての受信バッファ 209 は受信バッファフル信号 241 によって、自身が現在保持しているデータ以上のデータを保持できないことを通知可能に構成されている。そこで、受信イネーブル信号 250 がアサートされ且つ受信バッファ 209 の受信バッ

50

アフル信号 241 (r _ b u f f _ f u l l) がデアサートされていれば、受信データ S P 変換部 210 は受信バッファ 209 に平行形式の受信データ 234 を送信する。一方で、受信イネーブル信号 250 がアサートされていなければ、受信データ S P 変換部 210 はスキュー調整後受信データ 235 の受信を停止する。

【 0029 】

受信データ S P 変換部 210 はデータの受信を開始すると、受信ステータス信号 239 (r c v _ s t a t u s) をアサートする。受信データ S P 変換部 210 は最終データを外部デバイス 107 から受信するまで受信ステータス信号 239 をアサートし続け、スキュー調整後受信データ 235 のエンドビットを検出したらデアサートする。受信データ S P 変換部 210 が受信バッファ 209 に平行形式の受信データ 234 を送信すると、
10

【 0030 】

受信バッファ 209 は受信データ S P 変換部 210 から平行形式に変換された受信データ 234 (r _ d a t a _ b u f f) を受信し保持する。ここで、受信バッファ 209 がフルになった場合、受信バッファ 209 は出力クロック制御部 213 及び受信データ S P 変換部 210 へ受信バッファフル信号 241 をアサートする。一方で、受信バッファ 209 がエンプティ - になった場合には、受信バッファ 209 は DMA I / F 制御部 204 に受信バッファエンプティ - 信号 233 をアサートする。
20

【 0031 】

DMA I / F 制御部 204 は受信バッファ 209 の受信バッファエンプティ - 信号 233 のデアサートを受けて、受信バッファ 209 に外部デバイス 107 からの受信データが残っていることを検出する。そして、受信バッファ 209 が保持している平行形式の受信データ 232 (r _ d a t a _ d m a) を受信バッファ 209 から受信し、DMA I / F 111 へ送信する。ただし、受信バッファ 209 がエンプティ - になった場合には、データの受信を停止させる。データの受信を停止させないと、受信バッファ 209 のバッファアンダーランが発生することがある。従って、受信バッファ 209 の受信バッファエンプティ - 信号 233 がアサートされている場合には、DMA I / F 制御部 204 は平行形式の受信データ 232 の受信を停止し、DMA I / F 111 へのデータの
30

【 0032 】

受信バッファ 209 の受信バッファエンプティ - 信号 233 がデアサートされたら、DMA I / F 制御部 204 は平行形式の受信データ 232 の受信を再開し、DMA I / F 111 への受信データの送信を再開する。

【 0033 】

一方で、受信バッファ 209 がフルの状態 で新たな受信バッファへの書込みが発生する場合には、受信バッファ 209 のバッファオーバーランが発生するので、データの受信を停止する。受信データ S P 変換部 210 がデータ受信中であり、且つ、受信バッファ 209 がフルの場合に、出力クロック制御部 213 は出力クロックイネーブル信号 243 をデアサートする。
40

【 0034 】

受信データ S P 変換部 210 が、データ受信中であることは、受信ステータス信号 239 のアサートによって検出される。受信データ S P 変換部 210 は最初の受信データを受信すると受信ステータス信号 239 をアサートし、最終データを受信するまでアサートし続け、最終データを受信するとデアサートする。そのため、受信データ待ちの状態 (スタートビット検出待ちの状態) では、出力クロックイネーブル信号 243 はデアサートしない。

【 0035 】

外部デバイス 107 は出力クロック 244 (d e v _ c l k) のデアサートを受けて、受信データ 237 (d 2 h _ d a t a ') の送信を停止する。データ受信は、受信バッ
50

ア209の受信バッファフル信号241 (r _ b u f f _ f u l l) がデアサートされるまで停止し続ける。受信バッファ209の受信バッファフル信号241 (r _ b u f f _ f u l l) がデアサートされると、受信データSP変換部210はデータの受信を再開する。そして、出力クロックゲーティング部214は出力クロック244のゲーティングを解除し、外部デバイス107は受信データ237の送信を再開する。

【0036】

〔スキュー補正構成〕

次に、スキュー補正をするための構成の詳細を説明する。

【0037】

図2(a)はスキュー制御部211のブロック図である。スキュー制御部211はCLKジェネレータ105 (図19) からホストクロック115 (h o s t _ c l k) を受け取る。ホストクロック115は入出力が直列に接続されているN1個の遅延素子216 (第1遅延手段) によって遅延させられる。各遅延素子216の出力は遅延選択部217に入力され、スキュー選択値238 (s k e w _ s e l) の値に基づいて出力に用いる遅延素子216が選択される。

10

【0038】

選択された遅延付きクロック信号246 (c l k _ w i t h _ s k e w) はフリップフロップ218へクロックとして入力される。一方、外部デバイスから送信されるシリアル形式の受信データ236 (d 2 h _ d a t a) はフリップフロップ218が受信し、フリップフロップ218によって遅延付きクロック信号246 (c l k _ w i t h _ s k e w) に同期させられる。同期させたシリアル形式の受信データは、スキュー調整後のシリアル形式の受信データ235 (d 2 h _ d a t a _ 1 d 、以降スキュー調整後受信データ235と称す)として、スキュー制御部211が受信データSP変換部210に出力する。

20

【0039】

なお、スキュー制御部211のN1個の遅延素子は、好ましくは(ホストクロック115の)1クロックをN1等分した遅延、もしくはそれより少し小さい遅延を生じさせる。

【0040】

図2(b)はスキュー設定値を0, 1, 2とした場合に、スキュー制御部211が扱う信号のタイミングチャートを示す。図2(b)の各信号を図1および図2(a)の符号と対応させると、上から順にh o s t _ c l k 115、d h 2 _ d a t a 236、c l k _ w i t h _ s k e w 246 (s k e w _ s e l 238 = 0, 1, 2)、d 2 h _ d a t a _ 1 d 235 (s k e w _ s e l 238 = 0, 1, 2)となる。

30

【0041】

図2(b)のタイミングチャートでは、スキュー選択値238 (s k e w _ s e l) が0と1の時には、c l k _ w i t h _ s k e w 246の立ち上がりでd 2 h _ d a t a 236が不定である。従って、フリップフロップ218に取り込まれるデータd 2 h _ d a t a _ 1 d 235も不定になってしまい、データの取り込みが正常に出来てない。一方、スキュー選択値238が2の時には、c l k _ w i t h _ s k e w 246の立ち上がりでd 2 h _ d a t a 236が安定したデータを出力している為、フリップフロップ218に取り込まれるデータd 2 h _ d a t a _ 1 d 235も正常である。スキュー選択値238の調整は、後述するキャリブレーションシーケンスによって行われる。

40

【0042】

〔サイクル補正構成〕

次に、サイクル遅延を補正するための構成の詳細を説明する。

【0043】

図3(a)はサイクル制御部603の概略構成を示す。サイクル制御部603は入出力が直列に接続されているN2個のフリップフロップ701 (第2遅延手段) を有し、フリップフロップ701は出力クロックイネーブル信号243を夫々1クロックの遅延を生じさせる。サイクル制御部603は、入力された出力クロックイネーブル信号243をサイクル設定値256の示すサイクル数だけフリップフロップ701によって遅延させる。遅

50

延させた出力クロックイネーブル信号243は、受信イネーブル信号250として、受信データSP変換部210に入力されるようにしている。

【0044】

図3(b)はサイクル制御部603が扱う各種信号の波形を示す。図3(b)の各信号を図1の符号と対応させて説明すると、上から順にhost_clk115、dev_clk_en243、rcv_en250となる。サイクル制御部603は、入力される出力クロックイネーブル信号243をサイクル選択値249(cycle_sel=0,1,2,3)の値に応じて遅延させ、受信イネーブル信号250として出力する。サイクル選択値249の示す値を2とすると、サイクル制御部603はセクタ702によって2つのフリップフロップ701を通過した出力を選択することで、出力クロックイネーブル信号243を2サイクル遅延させて、受信イネーブル信号250として出力する。図3の構成ではN2個のフリップフロップ701があるので、1サイクルの整数倍(最大でN2サイクル)だけ出力クロックイネーブル信号243を遅延させることができる。

10

【0045】

本実施例で受信データSP変換部210は、受信イネーブル信号250のデアサートによってデータ取り込み停止タイミングを判断し、受信イネーブル信号250を再びアサートした際にデータ取り込み再開タイミングを判断する。

【0046】

ゲーティングモード設定レジスタ605はCPU I/F制御部201から、ゲーティングモードレジスタ設定値(247 gate_reg)を受け取り、保持する。また、ゲーティングモード設定レジスタ605は、受け取ったゲーティングモードレジスタ設定値247をゲーティングモード設定値(251 gate_mode)として出力クロック制御部213に出力する。

20

【0047】

ここで、本実施例の外部デバイスコントローラ103のゲーティングモードには、キャリアレーションモードと通常データ転送モードの二種類のモードがある。ゲーティングパターン設定レジスタ606はCPU I/F制御部201からゲーティングパターンレジスタ設定値(248 pattern_reg)を受け取り、保持する。また、受け取ったゲーティングパターンレジスタ設定値248をゲーティングパターン設定値(252 gate_pattern)として出力クロック制御部213に出力する。

30

【0048】

ゲーティングパターンレジスタ設定値248は、クロックゲーティングパターンを生成するための情報(ゲーティング情報)であり、どのようにクロックゲーティングするかを示す。(詳細には図5に後述するL, N, S, Eを示す情報である。)

図4(b)は本実施例で選択手段として機能する出力クロック制御部213の構成を示す。第一クロック制御部221(第一制御部)は、受信バッファフル信号241と受信ステータス信号239とを受信して、特許文献1や参考例(図21参照)と同様のクロック制御を行う(第三クロック制御モード)。一方で、第二クロック制御部222(第二制御部)は、受信ステータス信号239とゲーティングパターン252に基づいて、後述するクロック制御を行う(第一クロック制御モード)。さらに、第三クロック制御部223(第三制御部)は外部デバイスコントローラのホストクロック115を停止させずにそのまま出力するように制御する出力クロックイネーブル信号243を生成するように制御する(第二クロック制御モード)。

40

【0049】

まず、受信データSP変換部210が受信データ236の何ビット目まで受信したか等の受信状態を示す情報を受信ステータス信号239に付加して出力クロック制御部213へ送信する。第二のクロック制御部222は受信ステータス信号239に付加されている受信情報に基づいて、受信データSP変換部210が何ビット目まで受信したか判断する。そして、ゲーティングパターン252がゲーティングするように示す条件に当てはまっていなければ出力クロックイネーブル信号243を生成する。

50

【 0 0 5 0 】

クロック制御選択部 2 2 0 はセクタであり、ゲーティングモード設定レジスタからのゲーティングモード 2 5 1 が示す設定値に応じて、第一のクロック制御部 2 2 1、第二のクロック制御部 2 2 2 および第三のクロック制御部 2 2 3 から 1 つを選択的に機能させる。

【 0 0 5 1 】

〔コマンド・データフォーマット〕

ここで、外部デバイスコントローラ 1 0 3 と外部デバイス 1 0 7 との間でコマンドやデータの授受に用いる信号のフォーマットについて説明する。

【 0 0 5 2 】

外部デバイスコントローラ 1 0 3 と外部デバイス 1 0 7 は、図 2 1 (b) に示すフォーマットでシリアル形式のコマンド、又は、シリアル形式のデータを互いにやりとりする。なお、以下の説明ではパラレル形式のコマンド、パラレル形式のデータも同様のフォーマットであるとする。

【 0 0 5 3 】

まず、シリアル形式の送信コマンド 2 2 4 とシリアル形式の受信コマンド 2 2 6 の送信時に扱う信号について、図 2 1 (b) に基づいて説明する。送信コマンド 2 2 4 は 1 ビットのスタートビットと N ビットの送信コマンドと M ビットの CRC (巡回冗長検査信号) と 1 ビットのエンドビットによって構成されている。送信コマンド PS 変換部 1 2 5 は、パラレル形式の送信コマンド 2 2 3 の受信を検知すると、まず、1 ビットのスタートビットを送信する。続いて、N ビットのパラレル形式の送信コマンド 2 2 3 をシリアル形式の送信コマンド 2 2 4 に変換して送信する。送信コマンド PS 変換部 1 2 5 は、シリアル形式の送信コマンドの送信と共に CRC の演算を行う。そして、シリアル形式の送信コマンド 2 2 4 を送信した後に、演算された M ビットの CRC を送信する。最後に 1 ビットのエンドビットの送信を行い、コマンド送信を完了する。

【 0 0 5 4 】

シリアル形式の受信コマンド 2 2 6 のフォーマットも図 2 1 (b) に示すとおりであるが、受信コマンドと送信コマンドとが一致する必要はないので、受信コマンドと送信コマンドとでコマンド長や CRC の長さは異なっても構わない。

【 0 0 5 5 】

受信コマンド SP 変換部 1 2 6 は 1 ビットのスタートビットを検出し、コマンドの受信を開始する。続いて、N ビットのシリアル形式の受信コマンドを受信し、パラレル形式の受信コマンドに変換する。受信コマンド SP 変換部 1 2 6 はシリアル形式の受信コマンドの受信と共に CRC の演算を行う。そして、シリアル形式の受信コマンドを受信した後に、演算した CRC と送付される M ビットの CRC との比較 (巡回冗長検査) を行い、CRC エラーを検出する。最後に、1 ビットのエンドビットの受信を行い、コマンド受信を完了する。

【 0 0 5 6 】

送信データ PS 変換部 2 0 8 は、送信バッファ 2 0 7 からパラレル形式の送信データ 2 3 0 (s _ d a t a _ b u f) を受け取ると、送信コマンド PS 変換部 1 2 5 と同様にパラレル形式の送信データをシリアル形式の送信データ 2 3 1 に変換して外部デバイス 1 0 7 へ送信する。ただし、送信データの長さや CRC の長さは、送信コマンドと異なっても構わない。

【 0 0 5 7 】

受信データのフォーマットは図 2 1 (b) に示すとおりである。ただし、受信データの長さや CRC の長さは、送信コマンドと異なっても構わない。

【 0 0 5 8 】

受信データ SP 変換部 2 1 0 は、1 ビットのスタートビットを検出するとデータの受信を開始する。そして、受信コマンド SP 変換部 1 2 6 と同様に処理してシリアル形式のスキュー調整後受信データ 2 3 5 をパラレル形式の受信データ 2 3 4 に変換し受信バッファ

10

20

30

40

50

209へ送信する。送信データPS変換部208と受信データSP変換部210のCRCの演算処理、比較処理は送信コマンドPS変換部と受信コマンドSP変換部の処理と同様であるので省略する。

【0059】

〔キャリブレーション〕

次に、外部デバイスコントローラ103が正しくデータを取り込めるように、各種パラメータ（スキュー設定値、サイクル設定値）を調節するキャリブレーション処理について説明する。

【0060】

まず、図21に示す構成を有する参考例の外部デバイスコントローラ103におけるキャリブレーションフローを、図22を用いて説明する。

10

【0061】

まず、ステップS101で外部デバイスコントローラ103はCPU101からキャリブレーション開始を指示されると、外部デバイス107にキャリブレーションパターンを出力させるための送信コマンドを外部デバイス107に送信する。次に、外部デバイス107はキャリブレーション用の送信コマンドに対する受信コマンドを外部デバイスコントローラ103に送信する。さらに、外部デバイス107は予め決められたキャリブレーションパターンをシリアル形式の受信データ237の代わりに外部デバイスコントローラ103に送信する。外部デバイスコントローラ103は、前述したデータ受信フローに従って、キャリブレーションパターンの受信を行う。受信されたキャリブレーションパターンは、DMAコントローラ102、DRAMコントローラ104を経由して、DRAM106に書き込まれる。

20

【0062】

全てのキャリブレーションパターンがDRAMに106に書き込まれた後に、CPU101はステップS102において、予め期待値としてDRAM等に記憶しているキャリブレーションパターンと、実際に受信されたキャリブレーションパターンとを比較する。両者は一致しているとCPU101が判定した場合には、スキュー設定が正しいと考えられるために、キャリブレーションシーケンスを完了させる。一方、両者が一致しない場合には、スキュー設定が誤っていると考えられる。そこで、CPU101はステップS103において、異なるスキュー設定値227を外部デバイスコントローラ103に設定し、再度キャリブレーションシーケンス（S101、S102）を行う。以上の処理をキャリブレーションが成功するまで繰り返し行う。

30

【0063】

なおここでは、キャリブレーションパターンは外部デバイス107側に予め記憶されており、外部デバイス107はキャリブレーションを開始するコマンドを受信すると記憶しているキャリブレーションパターンを送信するものとする。

【0064】

ここで、スキュー調整が完了した際にステップS101で扱う信号の波形について説明する。図23は参考例における、受信データ236に1サイクル以上の遅延がある場合の、キャリブレーションシーケンス中の波形を示す。（ここで、受信データ236に1サイクル以上の遅延がある場合とは、外部デバイスコントローラ103と外部デバイス107との間で1/2サイクル以上の配線遅延が生じている場合に相当する。）図23の各信号を図21および図2の符号と対応させて説明すると、上から順にdev_clk244、dev_clk_en243、d2h_data236、clk_with_skew246、d2h_data_1d235、host_clk115、r_data_buf234、dev_clk'245、d2h_data'237となる。なお、取り込みクロックと取り込んだデータのタイミングに差があるが、これは実際の回路でクロックのエッジを基準にして取り込む場合に、データがフリップフロップに取り込まれるまでの遅延を示しており、本発明の課題と直接関係が無い箇所が生ずる遅延であり、その遅延量も小さい。従って、受信データが1サイクル以上遅延する際に、この遅延は無視（除外）し

40

50

てもよい。また、図 2 3 についてキャリブレーションの後半の波形は一部省略している。

【 0 0 6 5 】

図 2 3 の例では、外部デバイス 1 0 7 に入力されるクロック 2 4 5 (`dev_clk'`) が外部デバイスコントローラ 1 0 3 の出力する出力クロック 2 4 4 (`dev_clk`) に対し 1 / 2 サイクルだけ遅延している。また、外部デバイスコントローラの受信する受信データ 2 3 6 (`d2h_data`) が、外部デバイスの送信する受信データ 2 3 7 (`d2h_data'`) に対して、1 / 2 サイクルだけ遅延している。

【 0 0 6 6 】

これらの遅延は、外部デバイス 1 0 7 と外部デバイスコントローラ 1 0 3 間の基板上の往復遅延と外部デバイス 1 0 7 内での出力遅延や外部デバイスコントローラ 1 0 3 内の遅延などの影響によるものと考えられる。その結果、図 2 3 の例では、外部デバイスコントローラ 1 0 3 の出力クロック 2 4 4 に対して、外部デバイス 1 0 7 が応答出力するデータを外部デバイスコントローラ 1 0 3 が取り込む受信データ 2 3 6 に 1 サイクルの遅延がついていることになる。

【 0 0 6 7 】

キャリブレーションパターンは通常、受信バッファ 2 0 9 のサイズよりも十分小さい。そのため、参考例における外部デバイスコントローラ 1 0 3 はキャリブレーション中に出力クロックが停止しない。(そもそも、キャリブレーションパターンが受信バッファ 2 0 9 のサイズよりも大きかったとしても、出力クロックが停止するかどうかは、DMA コントローラ 1 0 2 の転送レートなどに依存する。従って、キャリブレーション中に出力クロック 2 4 4 が停止する状況は発生し難い。)

図 2 3 を参照すると、`r_data_buff(0)` 2 3 4 と入力されたシリアル形式の受信データ `d2h_data` 2 3 6 との内容が一致しているので、キャリブレーション処理は完了してしまう。しかし、図 2 4 に示すように実際にデータを受信する際にはデータの取りこぼしなどが発生する。

【 0 0 6 8 】

図 2 4 のタイミングチャートは、受信データ 2 3 6 に図 2 3 と同様に 1 サイクルの遅延がある場合に、実際に外部デバイスコントローラ 1 0 3 が外部デバイス 1 0 7 からデータを受信している際の信号の波形を示している。各信号における図 2 1 や図 2 との対応関係は図 2 3 と同様なので説明を省略する。

【 0 0 6 9 】

図 2 4 では、シリアル形式の受信データ 2 3 6 の “ D 0 ” を受信した時点で、受信バッファ 2 0 9 が受信バッファフル信号 2 4 1 (図 2 4 では不図示) をアサートされる状況が発生している。そして、受信バッファフル信号 2 4 1 のアサートに伴って、出力クロック制御部 2 1 3 は出力クロックイネーブル信号 2 4 3 (`dev_clk_en`) をデアサートする。出力クロックゲーティング部 2 1 4 は出力クロックイネーブル信号 2 4 3 のデアサートを受けて、出力クロック 2 4 4 をゲーティングする。出力クロックゲーティング部 2 1 4 によって出力クロック 2 4 4 をゲーティングしているにも係らず、外部デバイス 1 0 7 からシリアル形式の受信データ “ D 1 ” と “ D 2 ” が送信されてしまっている。

【 0 0 7 0 】

元々、受信バッファ 2 0 9 に入るデータ (受信バッファフル信号 2 4 1 を発行する時の基準となる) が外部デバイス 1 0 7 から外部デバイスコントローラ 1 0 3 に入る時点で 1 / 2 サイクル遅延している。そして、受信バッファフル信号 2 4 1 がアサートされた後に、出力クロック 2 4 4 がゲーティングされるまでの遅延に加えて、出力クロック 2 4 4 がゲーティングされたことを外部デバイス 1 0 7 が認識できるまでの 1 / 2 サイクルの遅延がある。結果的に、外部デバイスコントローラ 1 0 3 はデータ送信の停止を要求していることを外部デバイス 1 0 7 が認識するには、これらの遅延の総和だけ遅れるため、“ D 1 ”、“ D 2 ” が送信されてしまう。

【 0 0 7 1 】

受信データ S P 変換部 2 1 0 は、出力クロックイネーブル信号 2 4 3 のデアサートを受

10

20

30

40

50

けて、すぐにシリアル形式の受信データ236の受信を停止する。これによって、受信データSP変換部210はシリアル形式の受信データ“D1”を受信できなくなってしまう。(図に示すように、受信データSP変換部210が受信し、受信バッファ209に送信するパラレル形式の受信データ234には“D1”が含まれていない。)

また、受信バッファ209が受信バッファフル信号241をデアサートされてから、出力クロックイネーブル信号243がアサートされ、出力クロック244のゲーティングが解除される。出力クロック244は立ち下がり基準にゲーティングしたので、解除されてから1/2経ってから立ち上がる。しかし、出力クロック244のゲーティングが解除されているにも係らず、外部デバイス107と外部デバイスコントローラ間の遅延によって外部デバイス107からシリアル形式の受信データ“D2”が送信され続けてしまう。

【0072】

一方で、受信データSP変換部210は、出力クロックイネーブル信号243のアサートを受けて、すぐにシリアル形式の受信データの受信を再開する。そのため、受信データSP変換部210はシリアル形式の受信データ“D2”を2回受信してしまう。パラレル形式の受信データ234を見るとD2が2回受信されてしまっていることが分かる。

【0073】

このように、外部デバイスコントローラ103と外部デバイス107との間で往復して1サイクル以上の遅延がある場合には、参考例の構成ではキャリブレーションが完了しても(図23)、実際のデータ受信は失敗してしまうことがある(図24)。

【0074】

以上のことから、参考例の外部デバイスコントローラ103では、キャリブレーションシーケンスで受信データ236に発生する1サイクル以上の遅延の有無を検出できない場合があるということが分かる。

【0075】

なお、図24は説明の為に受信バッファフル信号241がアサートされてデアサートされるまでの期間が極端に短く受信バッファが極端に小さい前提になっているが、これは説明を簡単にするためである。

【0076】

次に、本実施例のキャリブレーションフローを図4(a)を用いて説明する。本実施例のキャリブレーションフローでは、まず、第三クロック制御部223を機能させながらスキュー調整を行い、次に第二クロック制御部222を機能させながらサイクル調整を行う。なお、キャリブレーションパターンが、受信バッファのサイズよりも十分に小さく、キャリブレーションパターン受信中に必ずクロックゲーティングが発生しないことが分かっているならば、第三クロック制御手段の代わりに第一クロック制御手段を用いてもよい。

【0077】

ここで、スキュー調整とは図2に示す構成によってホストクロック115と受信データ236の位相のズレ(位相ズレ)を修正するように調整するものである。一方で、サイクル調整とは図3に示す構成によってホストクロック115と受信データ236の周期単位のズレを修正するように調整するものである。以降の説明では周期ズレ(データ取り込みタイミングのズレやデータ取り込み再開タイミングのズレに相当する)を単に遅延サイクル数(サイクル遅延量)とする。

【0078】

まず、スキュー調整フローについて説明する。ステップS1201において、CPU101は外部デバイスコントローラ103に対して、クロックゲーティングモードを、クロックを停止させないキャリブレーションモード(第三クロック制御部を使うモード)に設定する。CPU101は、ステップS1202において、外部デバイスコントローラ103に対し、キャリブレーションパターンの取得を指示する。外部デバイスコントローラ103による、キャリブレーションパターンの取得処理は、参考例と同様である。キャリブレーションパターンの取得が完了すると、CPU101はステップS1203において、受信する予定のキャリブレーションパターンとしてROMやRAMから読み出したものと

10

20

30

40

50

、実際に受信したキャリブレーションパターンとの比較を行う。比較結果が一致している場合には、スキュー設定が正しいと考えられるために、スキュー調整フローを完了する。

【 0 0 7 9 】

一方、比較結果が一致しない場合には、スキュー設定が誤っていると考えられる。その場合、CPU 101 はステップ S 1 2 0 4 においてスキュー設定レジスタ 2 1 2 の値を変更し、スキュー設定を変更してステップ S 1 2 0 2 へ遷移する。スキュー設定を変更し、ステップ S 1 2 0 3 でパターンが一致するまでステップ S 1 2 0 2、S 1 2 0 3 の処理を繰り返し行う。

【 0 0 8 0 】

続いて、サイクル調整フローについて説明する。ステップ S 1 2 0 5 において、CPU 101 は外部デバイスコントローラ 103 のクロックゲーティングモードを、受信ステータス信号 2 3 9 の示す値とゲーティングパターン 2 5 2 とに基づいてクロックを停止させるキャリブレーションモード（第二クロック制御手段を機能させるモード）に設定する。さらに、CPU 101 は外部デバイスコントローラ 103 に対して、ゲーティングパターンレジスタ設定値 2 4 8 を設定する。CPU 101 は、ステップ S 1 2 0 6 において、外部デバイスコントローラ 103 に対し、キャリブレーションパターンの取得を指示する。外部デバイスコントローラ 103 による、キャリブレーションパターンの取得処理の概略は、参考例と同様である。

【 0 0 8 1 】

ただし、実施例 1 の出力クロック制御部 2 1 3 は、ゲーティングパターン 2 5 2 に応じて、出力クロックイネーブル信号 2 4 3 を発行する。そして、出力クロックゲーティング部 2 1 4 はキャリブレーションパターンの受信中に、出力クロックイネーブル信号 2 4 3 に応じて出力クロック 2 4 4 をゲーティングする。ステップ S 1 2 0 6 においてサイクル演算部 6 0 2 がキャリブレーションパターンの取得を完了するとステップ S 1 2 0 7 に遷移する。そして、サイクル演算部 6 0 2 はステップ S 1 2 0 7 において、受信する予定のキャリブレーションパターンとして CPU 101 が DRAM 106 などから読み出して期待値設定レジスタ 6 0 4 に格納させている期待値と、実際に受信したキャリブレーションパターンとの比較を行う。実際に受信するキャリブレーションパターンは、ゲーティングパターンやキャリブレーションパターン（期待値）、現在のサイクル設定値と実際の遅延サイクルとの差、によって異なる。（詳細については後述する。）

図 20 に、キャリブレーションパターンとして“01010101”を用いた一例を示す。詳細は後述するが、現在のサイクル設定値と実際の遅延サイクル数が合致している場合は、外部デバイスコントローラ 103 はキャリブレーションパターンを“01010101”のように受信できる。しかし、現在のキャリブレーションパターンに対して、実際の遅延サイクル数が 1 サイクル多い場合にキャリブレーションパターンとして、“01011101”のように受信する。その場合、サイクル演算部 6 0 2 が前述の処理によって受信したキャリブレーションパターンと、受信する予定のキャリブレーションパターン（期待値）とを比較して、遅延の有無を検出する。遅延がある場合には、受信されるパターン（図 20 参照）に基づいて実際のサイクル数との差を検知する。サイクル演算部 6 0 2（又は CPU 101）は S 1 2 0 8 において、検知された実際の遅延サイクルを、サイクル設定値として設定し、サイクル調整フローを完了する。なお、キャリブレーションシーケンスを終了した場合、出力クロック制御部 2 1 3 が第三クロック制御モードに切り替えてデータの送受信に備える。

【 0 0 8 2 】

上述したように、キャリブレーションパターンのサイズを受信バッファ 2 0 9 のサイズよりも小さくすると、キャリブレーションシーケンス中に受信バッファ 2 0 9 がフルになることはない。しかし、本実施例 1 の外部デバイスコントローラ 103 は受信バッファフル信号 2 4 1 のアサートによらず、出力クロック 2 4 4 をゲーティングパターンに基づいてゲーティングする。サイクル設定又はスキュー設定が誤っていれば、そのとおりに誤ったキャリブレーションパターンが（キャリブレーションが成功していないことがわかるよ

10

20

30

40

50

うに)取得される。図4(a)に示すフローを用いてキャリブレーションを完了した後、第一クロック制御部221を用いるモードに切り替えるため、CPU101はゲーティングモード設定レジスタ605にクロック制御選択部へ“0”を格納させる設定をし、データの送受信に備える。

【0083】

以上の処理によって、本実施例の外部デバイスコントローラ103は、受信データ236に1クロックサイクル以上の遅延が生じても、その遅延の有無を検出できるので、取りこぼしなくデータの送受信をするための各種パラメータ(スキュー設定、サイクル設定)を設定することができる。

【0084】

また、図4(a)に示すように、クロックゲーティングを行わずにスキュー調整でサイクル内の遅延の調整を行った後に、ゲーティングパターンに基づいてクロックゲーティングを行いながらサイクル間の遅延の調整を行うことで効率的にキャリブレーションできる。

このようにスキュー調整とサイクル調整を分離しないと、期待値と受信データ236が一致するまで総当りでキャリブレーションパターンの受信及び比較(最大でN1×N2回)を行わなければならない、キャリブレーション処理にかなりの時間を要する。

【0085】

また、キャリブレーションパターンが受信バッファ209のサイズよりも十分に小さければ、図4(a)のステップS1201~S1204で第三クロック制御部223を用いずに第一クロック制御部221によってスキュー調整をしてもよい。(キャリブレーションパターンの受信中に第一クロック制御部221を用いても必ずクロックゲーティングが発生しないためである。)この場合でも、ステップS1205~S1208の処理によって、図21(a)に示した参考例と比べるとキャリブレーションが成功し易くなる(回数を減らせる)。

【0086】

〔キャリブレーションパターンとゲーティングパターンについて〕

ここで、キャリブレーションパターンとゲーティングパターンの相関について説明する。キャリブレーション処理が完了した際には、サイクル設定値256の示すサイクル数が、実際に遅延しているサイクル数と一致するものとする。図24の例では、サイクル遅延が1サイクルである為に、設定すべき正しいサイクル設定値256の示す値は“1”となる。(図24に示す波形は、サイクル設定値256がサイクル未調節の場合に相当する。)

出力クロック244のゲーティングパターンとキャリブレーションパターン設定を適切にするとそれだけキャリブレーション処理の精度が上がる。例えば、図24の例では、受信データ“D1”を受信出来ず、その代わりに受信データ“D2”が2回受信されてしまっている。このようにデータの置き換えが発生してしまうため、好ましくないキャリブレーションパターン(後述の観点を検討していないパターン)を用いて受信データ236との比較をすると、キャリブレーションパターンの受信及び比較を行う回数が増えたり処理の精度が低下したりする可能性が高くなる。例えば、スキュー調整とサイクル調整について、期待値と受信データ236が一致するまで総当りで比較(最大でN1+N2回となる)を行う必要がある。

【0087】

以下、4つの観点で、出力クロック244のゲーティングパターンとキャリブレーションパターンの相関について説明する。これらの観点を検討しているキャリブレーションパターンを用いることでキャリブレーション処理の比較を短縮することができる(最大でN1+1回となる)。

【0088】

ゲーティングパターンは、キャリブレーションパターンと下記の4つの観点とに基づいて設定する。ゲーティングパターンをキャリブレーションに用いることで、受信データS

10

20

30

40

50

P変換部210のデータ取り込み制御を停止させる制御信号（出力クロックイネーブル信号243）がサイクル遅延量だけ遅延できていない場合に、実際に外部デバイスコントローラ103の受信するデータ（受信データ234）に重複や取りこぼしを発生させることができる。図20は好ましいゲーティングパターン（例として“01010101”）を設定した状況で、サイクル遅延量と受信データSP変換部210が受信することになるパターン（受信データ234）との相関を示している。

【0089】

（1）実際のディレイ量とサイクル設定値との差分（以降、ディレイ指標M）と受信されるキャリブレーションパターンとの相関

ディレイ指標 $M = (\text{実際のディレイ量}) - (\text{サイクル設定値})$ とし、ディレイ指標Mと受信されるキャリブレーションパターンとの相関について説明する。ここで、実際のディレイ量とは、サイクル設定値が“0”の場合（スキュー調節後でサイクル未調節に相当）に、受信データ236が出力クロック244に対して遅延しているサイクル数である。

【0090】

図6、図7は、受信イネーブル信号250の変化タイミングを、ディレイがない時のタイミングに固定し、実際のディレイ量を変化させた場合の平行形式の受信データ234の波形を示す。図6、図7の信号を上から順に図1の符号と対応させると、`dev_clk244`、`dev_clk_en243`、`dh2_data236`、`host_clk115`、`rcv_en250`、`r_data_buf234`である。また、図6、図7はディレイ指標Mが0, 1, 2, 3となる場合の4セットを示している。

【0091】

図6、図7では、D3を受信した時点で、出力クロックイネーブル信号243のゲーティングを行い、出力クロック244をデアサートにしている。図6、図7に示すように、受信するキャリブレーションパターン（`dh2_data236`に相当する）は、ディレイ指標Mによって異なる。ディレイ指標が、 $M > 1$ の場合には、ゲーティングを開始した時点で受信したデータ（図6、図7の例ではD3）を基準として、1サイクル後のデータからMサイクル後のデータまでがM+1サイクル後のデータに置き換わる。一方 $M = 0$ の時は、実際のディレイ量とサイクル設定値があっているため、正しいパターンが受信されている。

【0092】

（2）ゲーティング量（N）と受信されるパターンとの相関

出力クロック244をゲーティングし続けるサイクル数をゲーティング量Nとして、（スキュー調節後でサイクル調節が誤っている場合に）Nと受信されるキャリブレーションパターンの相関について説明する（Nの定義は図5参照）。

【0093】

図8、図9、図10はNが夫々3, 2, 1である時の、キャリブレーション中の波形を示す。図8、図9、図10の`r_data_buf234`の夫々の波形を参照するとNの値にかかわらず、Mサイクル分のデータが置き換わっていることが分かる。ただし、どのデータに置き換わるかはNの値によって異なっている。例えば、 $N = 3$ の場合は、D1~D3がD4に置き換わり、 $N = 2$ の場合はD1がD3に、D2とD3がD4に置き換わる。また、 $N = 1$ の場合は、D1がD2に、D2がD3に、D3がD4に夫々置き換わる。

【0094】

（3）ゲーティング間のサイクル（L）と受信されるパターンとの相関

ゲーティングを複数回行う場合、前のゲーティングタイミングから、次のゲーティングタイミングまでのサイクルをLとし、（スキュー調節後でサイクル調節が誤っている場合に）Lと受信されるキャリブレーションパターンとの相関について説明する。（Lの定義は図5参照）。図11、図12、図13、図14にLが1~4の時の、キャリブレーション中の波形を示す。L \geq Mならば、複数回のゲーティングは、独立したゲーティングとして扱える。すなわち、Mサイクル分のデータの置き換えが、ゲーティングの回数分だけ

10

20

30

40

50

発生する。そうでない場合には、複数回のゲーティングは、互いに相関を持ち、 $M * (ゲーティングの回数) - (M - L)$ 個のデータが連続して置き換わる。

【0095】

(4)ゲーティング開始位置(S)及びゲーティング終了位置(E)と受信されるパターンとの相関

ゲーティング開始位置をS、ゲーティング終了位置をEとし、(スキュー調節後でサイクル調節が誤っている場合に)S、Eと受信されるキャリブレーションパターンとの相関について説明する(S、Eの定義は図5参照)。

【0096】

図15、図16、図17にSとスタートビットの位置関係、及びEとエンドビットの位置関係を変えた場合のキャリブレーション中の波形を示す。図15は、ゲーティング開始位置Sがスタートビット検出後であり且つゲーティング終了位置Eがエンドビット検出前である場合の波形を示す。図16は、ゲーティング開始位置Sがスタートビット検出前であり且つゲーティング終了位置Eがエンドビット検出後である場合の波形を示す。図17はゲーティング開始位置Sがスタートビット検出後でありゲーティング終了位置Eがエンドビット検出後である場合の波形を示す。

【0097】

図16のように、ホストクロック115の1サイクルと同期するように周期的にクロックゲーティングを行う場合は、 $L > M$ を満たさず、且つSがスタートビットよりも前の位置で且つEがエンドビットよりも後の位置であると、キャリブレーションに適さない。サイクル調整が誤っているにも関わらず正しくデータが受信できてしまうためである。

【0098】

一方、Sがスタートビット後、又は、Eがエンドビット後であれば、サイクル調整が誤っている状況で正しくデータが受信できていないので、このゲーティングパターンはキャリブレーションに適していることが分かる。従って、S又はEのどちらか1つのだけをキャリブレーションパターン受信中に配置するか、ゲーティング間のサイクルLを考慮する必要がある。

【0099】

また、図15、図16、図17には直接示していないが、受信データSP変換部210が受信するキャリブレーションパターンの値が変動するタイミング(後述する所定数の異なる値が入力されている間)で、ゲーティング開始位置Sもしくはゲーティング終了位置Eがくるようにゲーティングパターンを設定しなければならない。すなわち、キャリブレーションパターンの値が変動するタイミングの1つで出力クロック244を停止し始める又は停止を解除するようにゲーティングパターンを設定する。例えば、値が変動するタイミングとは“01”(又は“10”)の場合であって、ゲーティング開始位置Sは“01”の“1”(又は“10”の“0”)、ゲーティング終了位置Eは“01”の“0”(又は“10”の“1”)にくるようにゲーティングパターンを設定する。

【0100】

以上説明したように、キャリブレーションパターンとゲーティングパターンには(1)~(4)に示す相関がある。図6、図7の例では、キャリブレーションパターンの“D3”を受信した次のサイクルでゲーティングを行っている。この例の場合では、サイクル設定値と実際のディレイとの関係から、“D4~D6”が“D5~D7”に置き換わる可能性がある。そのため、仮にキャリブレーションパターンの“D4~D7”が全て同じ値であった場合、置き換わる前の値と置き換わる後の値が同じ値である為に、ディレイ(サイクル遅延)の検出には好ましくない。図6、図7の例では、(1)~(4)の相関を考慮して、キャリブレーションパターンの“D4~D7”を以下のパターンにすることが好ましい。

“D4~D7 = 0101”

これは、(1)~(4)の相関に基づいて設定している、もちろん“1”と“0”は逆であっても、その場合に対応する期待値を設定しておけばよい。

【 0 1 0 1 】

さらに、このキャリブレーションパターン（期待値）を用いた場合の、実際のサイクル数との差と受信されるパターン（受信データ236）との関係を図20に示す。受信されるパターンと図20に示す遅延量との関係を示す情報に基づいて実際のサイクル数との差が判定可能である。なお、Mに応じて1：1で異なるキャリブレーションパターンが受信されるようなパターンを設定する必要がある。従って、（1）～（4）の相関に沿ったゲーティングパターンを用いる場合、キャリブレーションパターン中の連続する値（ビット、“1”や“0”など）について、先に外部デバイスコントローラ103へ入力される値と異なる値が所定数だけ連続する場合、所定数サイクルまでの遅延を検出できる。すなわち、検知され得る遅延サイクル数以上、直前に（受信データSP変換部210へ）入力される値と異なる値が連続することが好ましい。上述の“D4～D7=0101”の場合“101”の部分で、直前に入力される値と異なる値が連続していることを保証できる。

10

【 0 1 0 2 】

なお、この関係を示す遅延情報（図20のようなルックアップテーブルや単なる数列など）は予め試験をしてCPU101が参照可能なDRAM106やその他の記憶装置に記憶しておき、サイクル遅延量を判定する際にCPU101が参照すればよい。

【 0 1 0 3 】

次に、図1の構成に正しいパラメータ（スキュー設定値、サイクル設定値）を用いて動作させた場合の、データ受信中に外部デバイスコントローラ103と外部デバイス107の扱う信号の様子を図18に示す。図18の例も図24の例と同様に、サイクル遅延が1サイクルであるが、キャリブレーションを完了し、1サイクルに対応するサイクル選択値249（サイクル設定値256）が設定されている時の波形を示している。（なお、受信データ236に発生する遅延が1サイクルである場合の、正しいサイクル選択値249（サイクル設定値256）の示す設定は1になる。サイクル選択値249の示す設定が1である為、受信イネーブル信号250は、出力クロックイネーブル信号243に対して、サイクル制御部603によって1サイクル遅延する。）

20

図18の例では、シリアル形式の受信データ236について“D0”の受信を開始した時点で、出力クロックイネーブル信号243をデアサートして、出力クロック244をゲーティングしている。出力クロック244をゲーティングしているが、外部デバイス107からシリアル形式の受信データ237について“D1”と“D2”が送信される。出力クロックイネーブル信号243のデアサート開始から1サイクル後に、受信イネーブル信号250もデアサートされる。受信データSP変換部210は、受信イネーブル信号250のデアサートを受けて、すぐにシリアル形式の受信データの受信を停止する。

30

【 0 1 0 4 】

そのため、スキュー調整後受信データ235の“D1”を受信したまま、データの受信を停止する。パラレル形式の受信データ234の1ビット目データを見ると“D1”が受信できており、受信データSP変換部210のデータ取り込み停止タイミングがサイクル遅延に対応するように調節できていることが分かる。

【 0 1 0 5 】

また、出力クロックイネーブル信号243はデアサートされた1サイクル後に再びアサートされ、これを受けて出力クロック244のゲーティングが解除されている。ここで、出力クロック244のゲーティングが解除されていても、前述の受信データ236に1サイクル以上の遅延があるため、外部デバイス107からシリアル形式の受信データ236の“D2”が送信され続ける。

40

【 0 1 0 6 】

サイクル設定値256の示す遅延量（1サイクル）によって、出力クロックイネーブル信号243のアサートの1サイクル後に、受信イネーブル信号250もアサートされる。受信データSP変換部210は、受信イネーブル信号250のアサートを受けて、すぐにシリアル形式の受信データの受信を再開する。そのため、シリアル形式の受信データ236の“D2”を正しく受信できる。パラレル形式の受信データ234の1ビット目を見る

50

と“D2”が正しく受信出来ており受信データSP変換部210のデータ取り込み再開タイミングがサイクル遅延に対応するように調節できていることが分かる。

【0107】

以上、本実施例では、設定が正しくないままキャリブレーションが完了してしまうことを抑制できる。従って、受信データ236に1サイクル以上の遅延があっても正しく検出ができる。

【0108】

また、本実施例のサイクル制御部603によると、正しいサイクル設定値256が設定されていれば、外部デバイスコントローラ103と外部デバイス107との間で生ずるサイクル遅延の量に対応するように、受信データSP変換部210のデータ取り込み停止タイミングとデータ取り込み再開タイミングとを遅延させられる。これにより、図24に示したようなデータの取りこぼしが発生することを抑制する。

10

【0109】

また、上述の実施例ではスキュー制御部211とサイクル制御部603を別々に構成しているが、スキュー調整とサイクル調整を単一の構成として組み合わせてもよいし、受信データSP変換部210に組み合わせてもよい。さらに、スキュー調整ではズレを補正したクロックを入力することで調整しているが、データを供給する系にスキュー調整やサイクル調整用の遅延構成(遅延素子、フリップフロップ)を直接配置し、セレクト等で遅延量を選択してもよい。

【0110】

20

また、上述の実施例ではキャリブレーション処理を実行する機会については言及していないが、公知の外部デバイスコントローラと同様の機会にキャリブレーションをすれば本発明の効果は得られる。例えば、ASIC100の起動時や、外部デバイスコントローラ103の初期化時、所定時間毎もしくは所定サイクル数毎、また外部デバイス107との接続をASIC100が検出した時に実施をすればよい。

【0111】

なお、前述の実施例において、サイクル演算部602、サイクル設定レジスタ601、期待値設定レジスタ604、はハードウェアとして説明しているが、CPU101で代用してソフトウェアで実現してもよい。その場合、CPU101がサイクル演算部602としての機能を実現するためのプログラムをDRAM106等から読み出して実行し、各種レジスタに相当する記憶領域はCPU101のキャッシュ(もしくはDRAM106)上に確保して、前述する各種レジスタの格納する値を記憶領域に記憶する。

30

【0112】

前述の外部デバイス107と外部デバイスコントローラ103間の配線遅延について、外部デバイス107が外部デバイスIF112に着脱できる構成である場合に遅延量のバラツキが大きくなると考えられる。実際には配線の長さや材質、温度上昇以外にも、接触不良など種々の要因による遅延を含むことが考えられる。

【0113】

上述の実施例ではホストクロック115がどの程度の周波数であるかは例示していないが、動作周波数が高くなればなるほど、図21(a)に示す構成で受信データ236に発生する1サイクル以上の遅延検出が困難になり、キャリブレーション不良も生じ易い。外部デバイスコントローラ103と外部デバイス107との間でより高速にデータをやりとりするにはホストクロック115の周波数を高めることが考えられるため、本発明は高速で信頼性の高いデータ通信の実現にも寄与するといえる。

40

【0114】

また、上述の実施例でサイクル演算部602がキャリブレーションパターン(受信データ234)と期待値とを比較してサイクル調整を行っているが、スキュー調整と同様にCPU101が期待値を読み出し、比較処理をするようにしてもよい。

【0115】

また、上述の実施例では受信データ236のデータ部分(図21bのD0、D1...)に

50

ついて比較しているが、算出手段としての受信データSP変換部210の算出したCRC部分(図21bのCRC0、CRC1...)を比較に用いるようにしてもよい。その場合は、予め記憶しているキャリブレーションパターンについて、スキュー設定とサイクル設定が正常な場合に受け取るCRCを予め演算して記憶しておく必要がある。

【0116】

なお、図1では外部デバイスコントローラ103と外部デバイス107の間では1bit幅のバスを用いて通信している様に説明しているが、4bit幅のバスや8bit幅のバス等でもよく、本発明はバス幅には限定されず適用できる。しかし、例えば4bit幅(8bit)のバスを用いる場合などは、スキュー制御部211にフリップフロップ218と遅延選択部217を4つ配置し1bit幅ずつスキュー調整をできるようにしておい

10

【0117】

また、上述の実施例ではホストクロック115の動作周波数が単一の例しか説明していないが、外部デバイス107を識別して動作周波数を切り替えるようにしてもよい。例えば、CLOCKジェネレータ105から外部デバイスコントローラ103の間に、ホストクロックを分周させる分周回路や逡倍回路等を設けて外部デバイスコントローラ103に入力するホストクロックの周波数を替えてもよい。この場合CLOCKジェネレータ10

20

【0118】

この時、前述のキャリブレーションが不調な場合に外部デバイス107に入力されるホストクロック115をより低い周波数に減少させて外部デバイスとの通信を安定させるようにしてもよい。キャリブレーションが不調な場合とは、例えば前述のキャリブレーションが単位時間当たり所定回数以上発生してしまう場合やキャリブレーションに要する時間が所定時間以上(例えば、総当りに要する回数以上になる場合)を要してしまう場合などが挙げられる。また、外部デバイスIF112の物理的なコネクタ形状を規格に定められた特定種類の外部デバイス107と嵌合するように設計する場合、分周回路や逡倍回路で切り替える周波数として特定種の規格で定められている周波数を用いても良い。(例

30

【0121】

また、上述の実施例における送信バッファ207や受信バッファ209はFIFO構造にしてもよい。その場合、バッファフル信号やバッファエンプティ信号はFIFOの空き容量を示す情報(残量情報)に基づいて作成したり、残量情報をそのまま代用したりしてもよい。この場合、受信バッファ209のFIFOが格納できるデータのサイズがキャリブレーションパターンのデータのサイズ以上である場合は、キャリブレーションパターンを一旦FIFOに書き込んだ後に、CPU101が比較にそのまま用いるようにしても

40

【0122】

50

また、上述の実施例における出力クロックイネーブル信号243などの制御信号は、イネーブル信号をデアサートするタイミングにディスエーブル信号をアサートするように構成しても構わない。

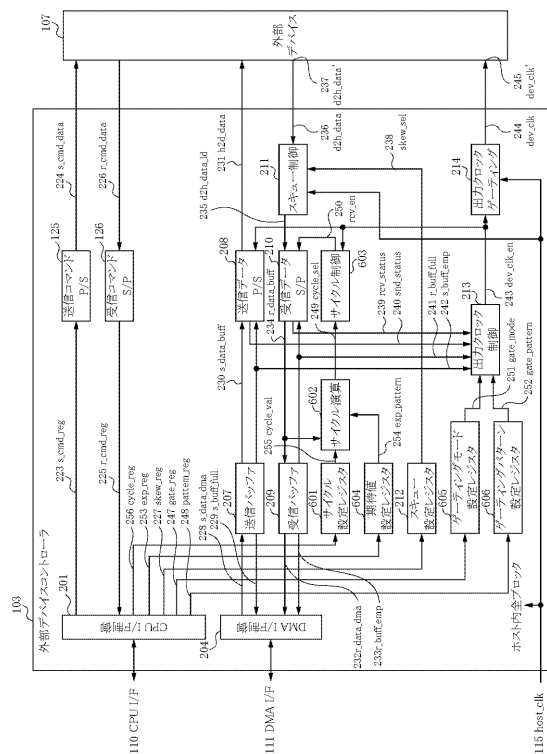
【0123】

なお、上述の実施例では外部デバイスコントローラ103を有する情報処理装置として本発明を説明したが、上述の外部デバイス107の構成を情報処理装置内部として有する場合にも適用できるので、デバイスコントローラに適用できるともいえる。また、本発明を適用する情報処理装置としては、画像処理装置や計算処理装置など種々の装置が挙げられる。

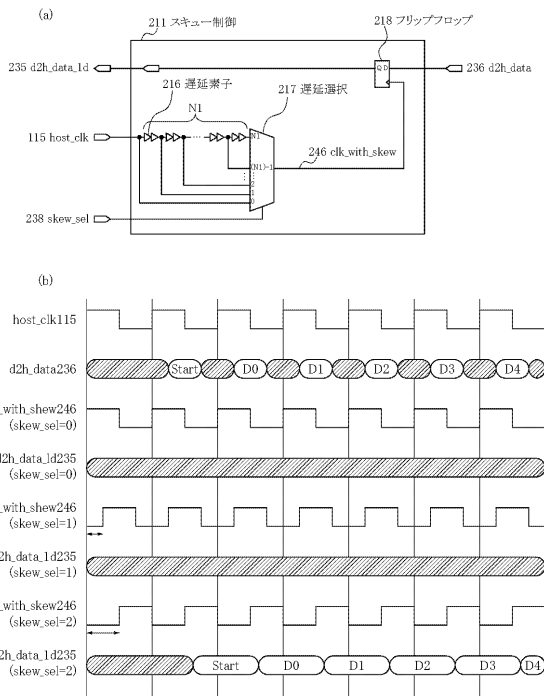
【0124】

また、本発明は、以下の処理を実行することによっても実現される。即ち、上述した実施例の機能を実現するソフトウェア(プログラム)を、ネットワーク又は各種記憶媒体を介してシステム或いは装置に供給し、そのシステム或いは装置のコンピュータ(またはCPUやMPU等)がプログラムを読み出して実行する処理である。

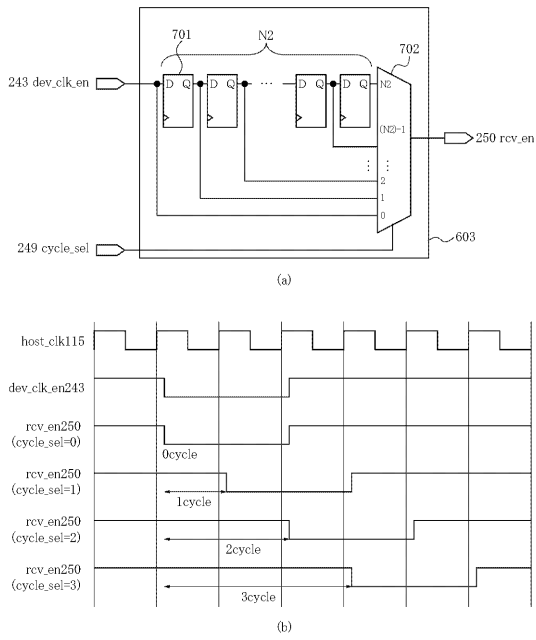
【図1】



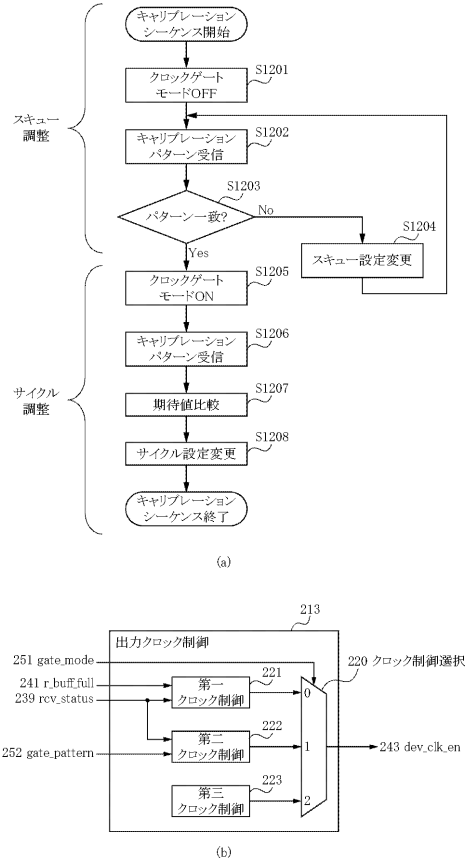
【図2】



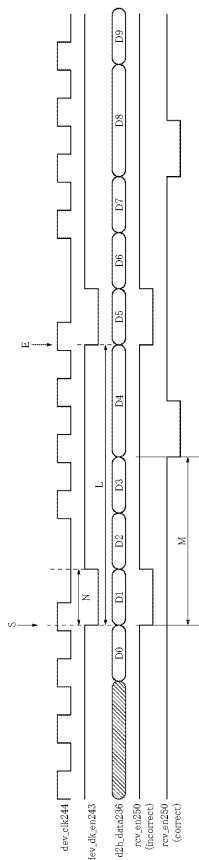
【図3】



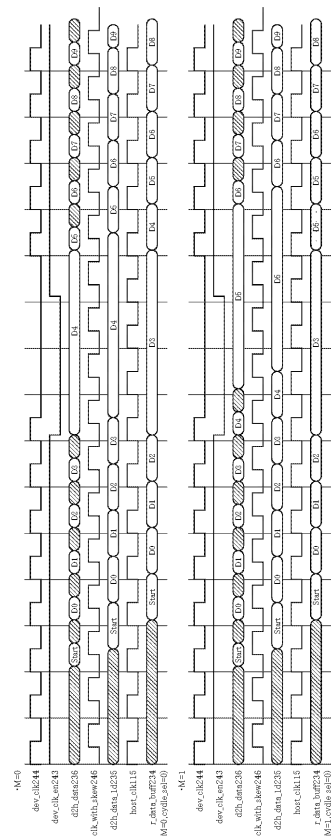
【図4】



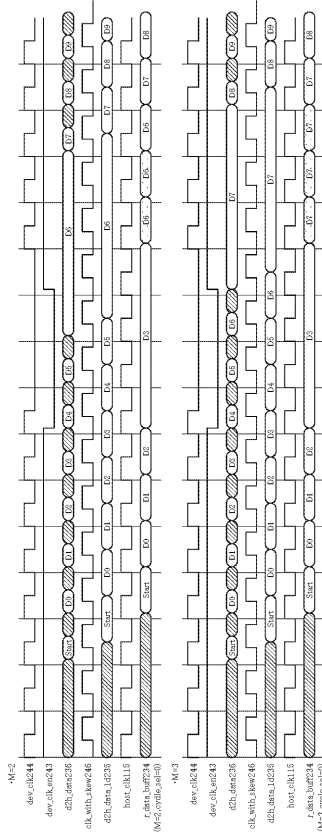
【図5】



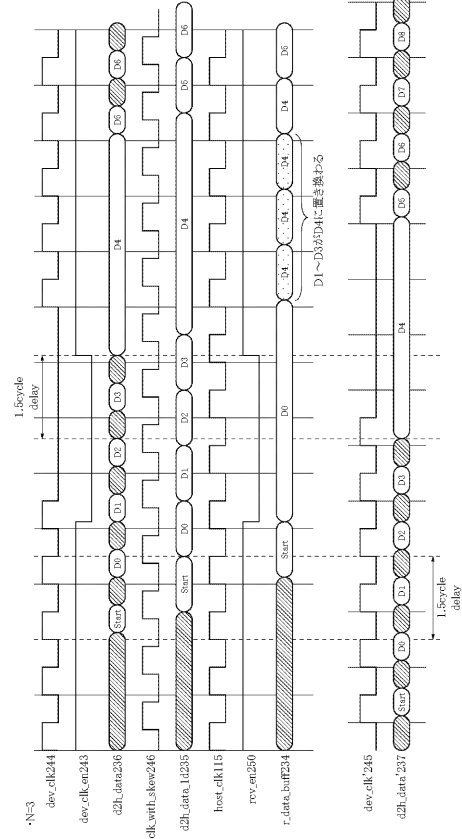
【図6】



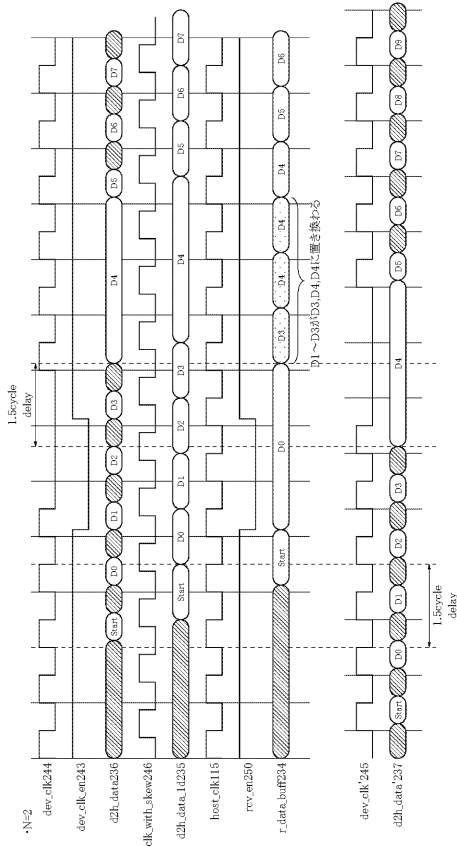
【図7】



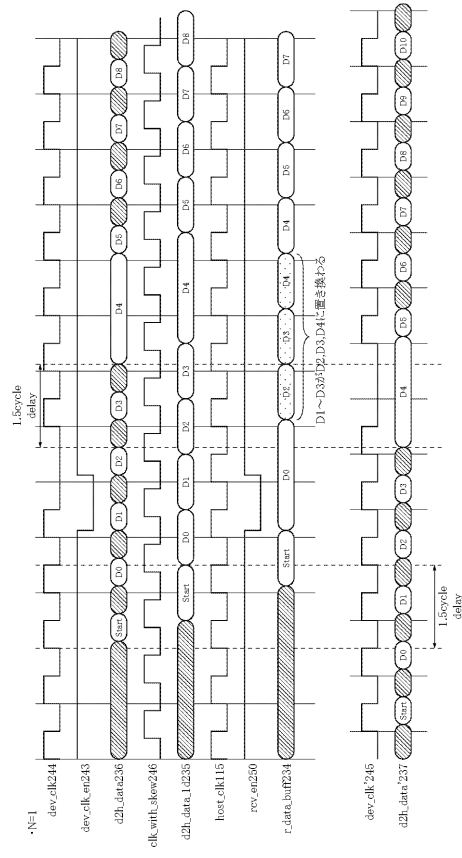
【図8】



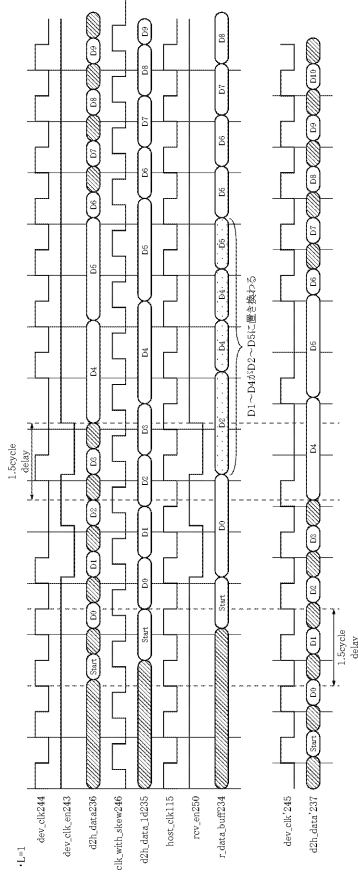
【図9】



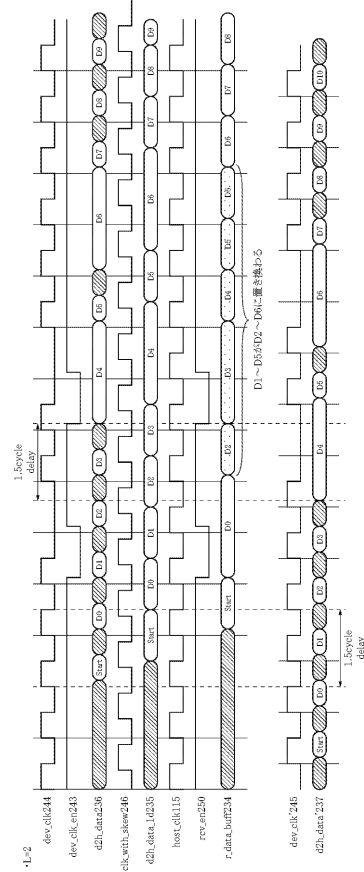
【図10】



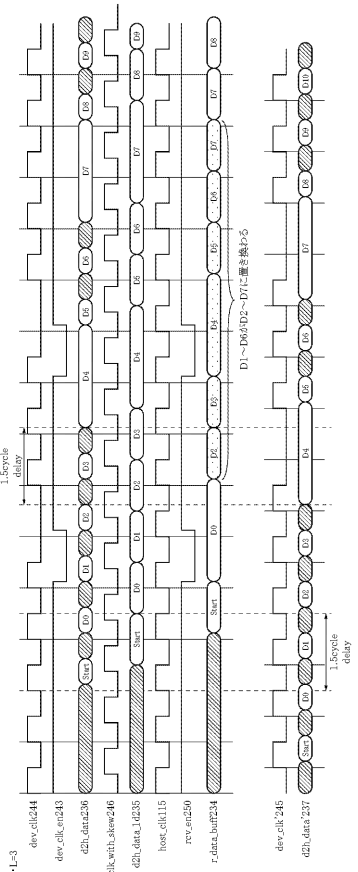
【図 1 1】



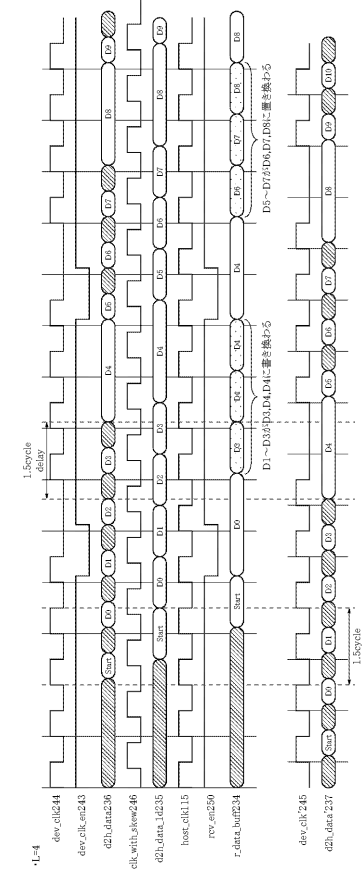
【図 1 2】



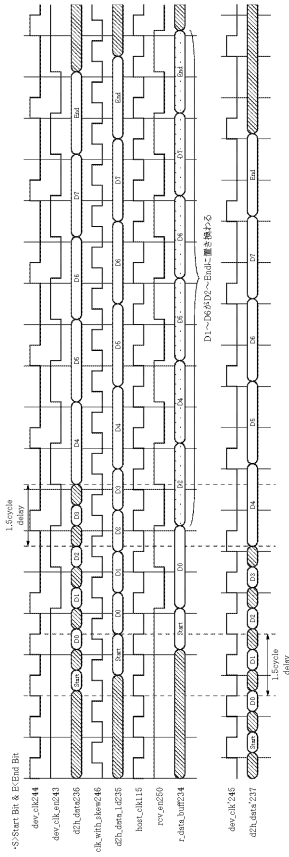
【図 1 3】



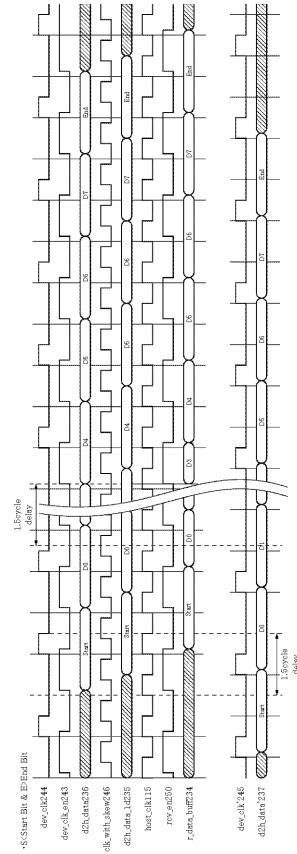
【図 1 4】



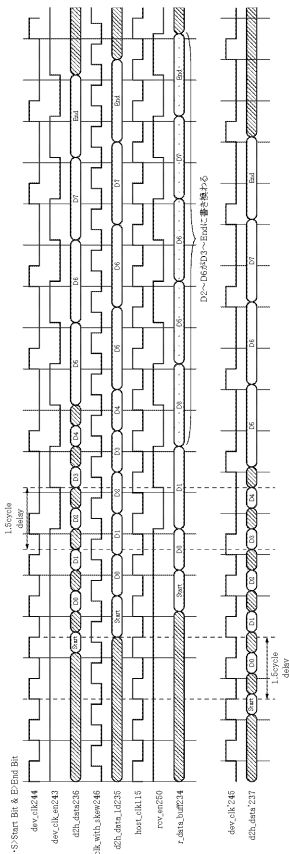
【図 15】



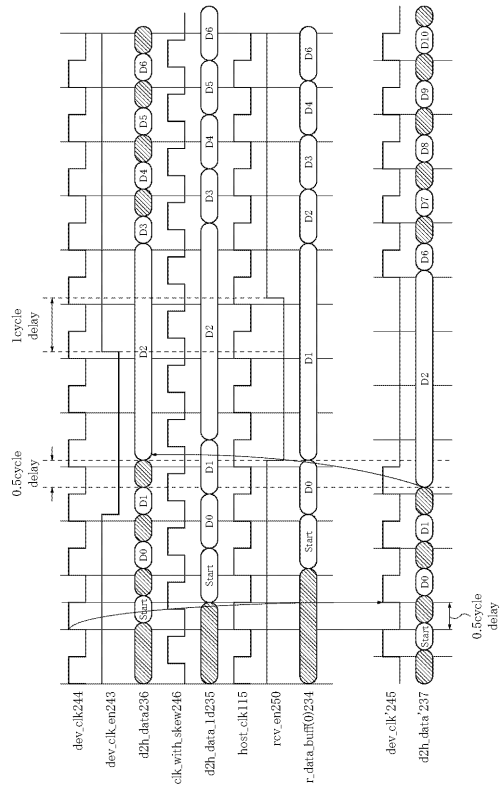
【図 16】



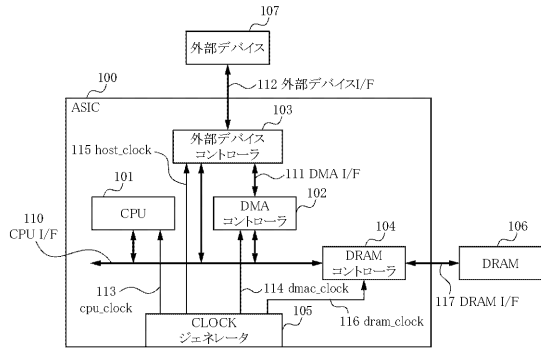
【図 17】



【図 18】



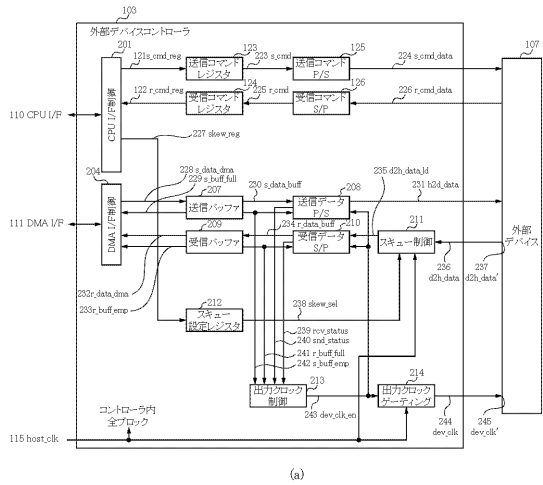
【図19】



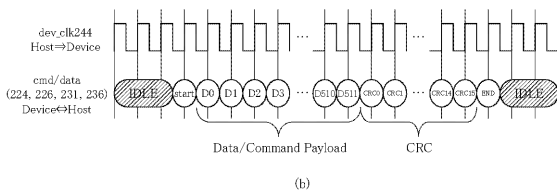
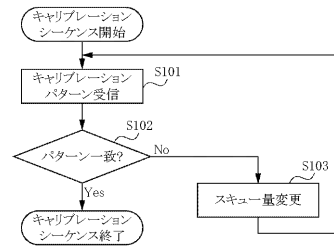
【図20】

実際のサイクル数との差	受信されるパターン
0	01010101
1	01011101
2	01010001
3	01011111

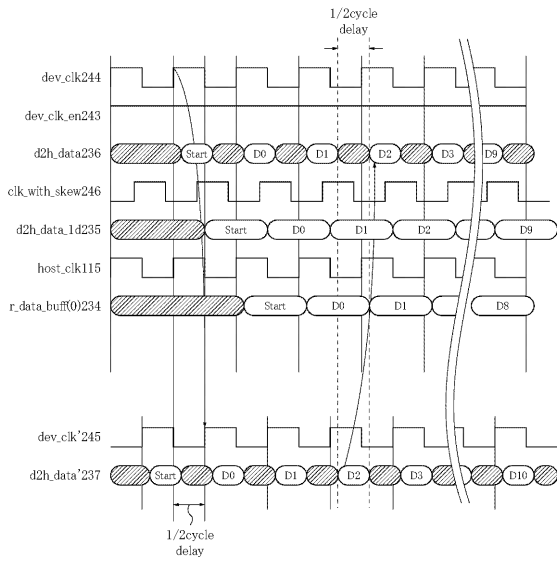
【図21】



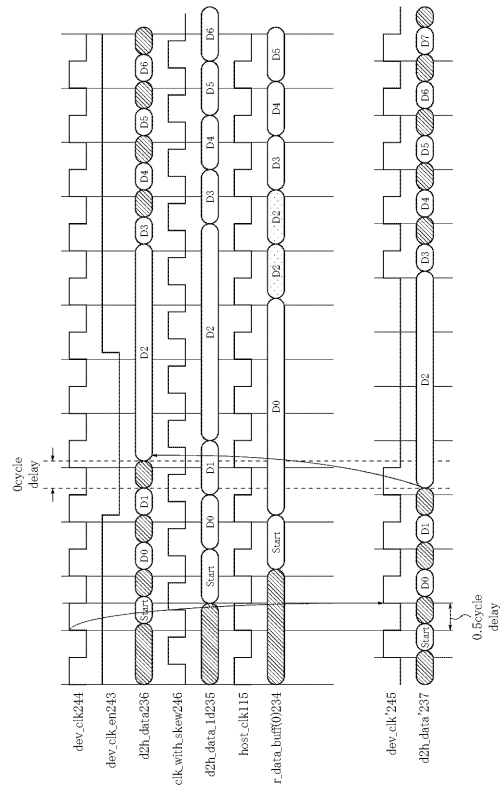
【図22】



【 2 3 】



【 2 4 】



フロントページの続き

- (56)参考文献 特開平09-069829(JP,A)
国際公開第2007/113945(WO,A1)
特開昭59-173839(JP,A)

(58)調査した分野(Int.Cl., DB名)

H04L 7/02
H04L 7/04