

(19) 日本国特許庁 (JP)

(12) 公 開 特 許 公 報 (A)

(11) 特許出願公開番号
特開2009-75973
(P2009-75973A)

(43) 公開日 平成21年4月9日 (2009. 4. 9)

(51) Int.Cl.			F I			テーマコード (参考)	
G 0 6 F	1 / 0 4	(2006. 01)	G 0 6 F	1 / 0 4	3 0 1 C	2 C 0 6 1	
H 0 4 N	1 / 0 0	(2006. 01)	H 0 4 N	1 / 0 0	C	5 B 0 7 9	
B 4 1 J	2 9 / 3 8	(2006. 01)	B 4 1 J	2 9 / 3 8	Z	5 C 0 6 2	

審査請求 未請求 請求項の数 10 O L (全 20 頁)

(21) 出願番号	特願2007-246094 (P2007-246094)	(71) 出願人	000001007
(22) 出願日	平成19年9月21日 (2007. 9. 21)		キヤノン株式会社
			東京都大田区下丸子3丁目30番2号
		(74) 代理人	100076428
			弁理士 大塚 康德
		(74) 代理人	100112508
			弁理士 高柳 司郎
		(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(74) 代理人	100130409
			弁理士 下山 治
		(74) 代理人	100134175
			弁理士 永川 行光

最終頁に続く

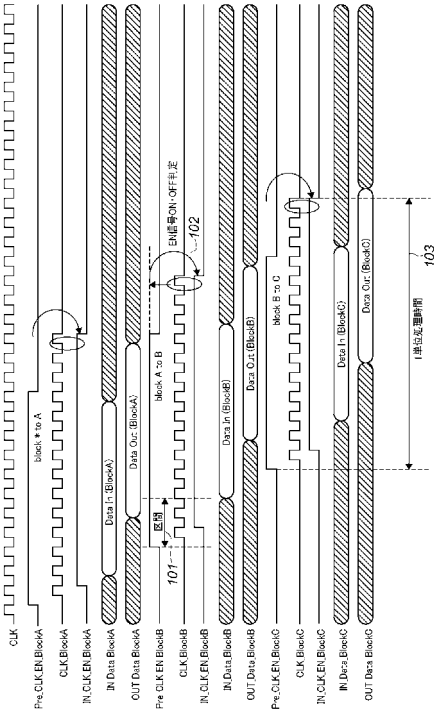
(54) 【発明の名称】 電子機器及び当該電子機器の電力制御方法

(57) 【要約】

【課題】 従来技術は、クロック信号を可変にしてもよいシステムが前提であるため、適用範囲が限られ、また回路構成が複雑になる。

【解決手段】 クロック信号に同期して動作する複数の処理ブロックを具備する電子機器において、複数の処理ブロックのそれぞれは、外部或は前段の処理ブロックからデータとともに入力されるクロックイネーブル信号に応じて自処理ブロックの動作クロック信号を発生し、その動作クロック信号により、入力されたデータを処理し、その処理済みデータを後段の処理ブロックに出力するとともに、後段の処理ブロックに対してクロックイネーブル信号を出力し、データの処理が完了した後、処理済みデータの出力が完了すると、自ブロックの動作クロック信号の発生を停止させる。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

クロック信号に同期して動作する複数の処理ブロックを具備する電子機器であって、
前記複数の処理ブロックのそれぞれは、

外部或は前段の処理ブロックからデータとともに入力される所定時間幅のクロックイネーブル信号に応じて当該処理ブロックの動作クロック信号を発生するクロック発生手段と、

前記動作クロック信号により、入力された前記データを処理する処理手段と、

前記処理手段により処理した処理済みデータを後段の処理ブロックに出力するとともに、
前記後段の処理ブロックに対して所定時間幅のクロックイネーブル信号を出力する出力手段と、

10

前記処理手段により前記入力されたデータの処理が完了した後、前記出力手段による前記処理済みデータの出力が完了すると前記クロック発生手段による前記動作クロック信号の発生を停止させる停止手段と、

を有することを特徴とする電子機器。

【請求項 2】

前記停止手段は、前記処理手段により前記入力されたデータの処理が完了した後、前記出力手段による前記処理済みデータの出力が完了した時点で、前記クロックイネーブル信号がイネーブルでない場合に前記クロック発生手段による前記動作クロック信号の発生を停止させることを特徴とする請求項 1 に記載の電子機器。

20

【請求項 3】

前記処理ブロックが複数の機能を実行できる場合、前記クロックイネーブル信号は各機能ごとに当該処理ブロックに供給されることを特徴とする請求項 1 に記載の電子機器。

【請求項 4】

前記出力手段は、前記処理手段により処理した処理済みデータを出力する準備ができた時点で、前記後段の処理ブロックに対するクロックイネーブル信号を出力することを特徴とする請求項 1 に記載の電子機器。

【請求項 5】

前記複数の処理ブロックのそれぞれは、前記動作クロック信号が供給されないときは電力消費量を減少させた状態となることを特徴とする請求項 1 乃至 4 のいずれか 1 項に記載の電子機器。

30

【請求項 6】

クロック信号に同期して動作する複数の処理ブロックを具備する電子機器の電力制御方法であって、

前記複数の処理ブロックのそれぞれにおいて、外部或は前段の処理ブロックからデータとともに入力される所定時間幅のクロックイネーブル信号に応じて当該処理ブロックの動作クロック信号を発生するクロック発生工程と、

前記動作クロック信号により、前記複数の処理ブロックのそれぞれにおいて入力された前記データを処理する処理工程と、

前記処理工程で処理した処理済みデータを後段の処理ブロックに出力するとともに、前記後段の処理ブロックに対して所定時間幅のクロックイネーブル信号を出力する出力工程と、

40

前記処理工程で、前記入力されたデータの処理が完了した後、前記出力工程での前記処理済みデータの出力が完了すると前記クロック発生工程での前記動作クロック信号の発生を停止させる停止工程と、

を有することを特徴とする電子機器の電力制御方法。

【請求項 7】

前記停止工程は、前記処理工程で前記入力されたデータの処理が完了した後、前記出力工程で前記処理済みデータの出力が完了した時点で、前記クロックイネーブル信号がイネーブルでない場合に前記動作クロック信号の発生を停止させることを特徴とする請求項 6

50

に記載の電子機器の電力制御方法。

【請求項 8】

前記処理ブロックが複数の機能を実行できる場合、前記クロックイネーブル信号は各機能ごとに当該処理ブロックに供給されることを特徴とする請求項 6 に記載の電子機器の電力制御方法。

【請求項 9】

前記出力工程は、前記処理工程で処理した処理済みデータを出力する準備ができた時点で、前記後段の処理ブロックに対するクロックイネーブル信号を出力することを特徴とする請求項 6 に記載の電子機器の電力制御方法。

【請求項 10】

前記複数の処理ブロックのそれぞれは、前記動作クロック信号が供給されないときは電力消費量を減少させた状態となることを特徴とする請求項 6 乃至 9 のいずれか 1 項に記載の電子機器の電力制御方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、電子機器及び当該電子機器の消費電力を抑える電力制御方法に関するものである。

【背景技術】

【0002】

印刷装置等のデジタル機器では、高機能及び高付加価値が求められるため、その消費電力は増加する傾向にある。逆に国際エネルギースタープログラム（TEC基準）などの省エネルギー規格は、年々基準が厳しくなっている。従って、消費者及び製造者の双方にとって消費電力の削減に対する要求は大きい。

【0003】

上述のTEC基準では、稼動電力のみばかりでなく、待機電力も含めて消費電力をある基準値以下に抑えなければならない。印刷装置を例にとると、印刷装置が稼動していない（ジョブの要求がなく待機状態）場合には、必要最低限の回路へのみ電力を供給し、それ以外の回路への電力供給を遮断する方法（スリープ）が一般的である。

【0004】

また、スタンバイ時や印刷待機時においても、大部分の電力供給をオフにしておき、印刷要求を受信した時点で必要な部分にのみ電力を供給するのが理想的である。しかし、このような制御を行うと電源の復帰処理に時間を要するため処理速度が低下する。また接続する機器によっては電源のオン／オフの回数に制限がある（ハードディスク装置など）ため、そのような電源の制御は難しい。また回路等への電源供給を部分的にオン／オフする（例えば、チップ内部の電源を分離して部分的に電源をオフ／オフする）のが困難である等の理由のため、上述の電源供給制御が実現していない。

【0005】

従って、他の方法として、電力を常時供給しておき、クロック信号を制御することで稼動時の消費電力を抑える方法が提案されている。例えば、クロック信号の必要な部分を除き、それ以外の部分は発振を一時的に停止させるゲーテッドクロック等の方法がある。更に、ゲーテッドクロックの制御方法として、ソフトウェアによりゲーテッドクロックレジスタを制御することで実現する方法もある。

【0006】

しかしながらソフトウェアによる制御の場合には、消費電力の削減効率を上げるためにシステムの電源制御の単位を、より細かく分割しなければならない。しかし、より細かく分割するほど処理が煩雑となり、またCPUに対する負荷も増大する。更に、制御可能な時間間隔が大きくなり、きめ細かな動的制御が不可能である。このためソフトウェアによる消費電力の削減には限界がある。

【0007】

そこでクロック信号の制御をハードウェアで行う方法が提案されている。以下に幾つかの例を挙げる。特許文献1は、高速クロックと低速クロックなど複数のクロック信号を準備しておき、処理ユニットの内容に応じて、クロック信号の発振周波数を可変にする方法を開示している。

【0008】

また、クロック信号の制御をハードウェアで行う他の方法を開示する特許文献2では、クロック信号を制御して、クロック信号を供給するメインモジュールと、そのクロック信号が供給される複数のサブモジュールとが存在している。そしてメインモジュールは、処理を要求するサブモジュールに対して、トランザクションと同時にクロック信号の供給を開始する。このトランザクションを受けたサブモジュールは、処理に必要な期間、クロック信号の供給を遅延するクロック遅延リクエスト信号をアサートし続ける。

10

【特許文献1】特開平8-272479号公報（7頁、図1）

【特許文献2】特開2002-108489号公報（6頁、図1）

【発明の開示】

【発明が解決しようとする課題】

【0009】

しかしながら特許文献1は、クロック信号を可変にしてもよいシステムが前提であるため、適用範囲が限られるという欠点がある。また回路構成も優先条件の判定回路や分周回路等を要するため比較的複雑である。また特許文献2は、基本構造がクロック制御におけるマスタとスレーブ構造であり、マスタスレーブ間にハンドシェクが発生する。従って回路構成が複雑になる。またマスタスレーブ構成では、システム構成（作り方）によっては適用しにくい等の欠点がある。

20

【0010】

本発明の目的は、上記従来技術の問題点を解決することにある。

本願発明の特徴は、簡単な構成で、消費電力を低減できる技術を提供できる。

【課題を解決するための手段】

【0011】

上記目的を達成するために本発明の一態様に係る電子機器は以下のような構成を備える。即ち、

クロック信号に同期して動作する複数の処理ブロックを具備する電子機器であって、前記複数の処理ブロックのそれぞれは、

30

外部或は前段の処理ブロックからデータとともに入力される所定時間幅のクロックイネーブル信号に応じて当該処理ブロックの動作クロック信号を発生するクロック発生手段と、

前記動作クロック信号により、入力された前記データを処理する処理手段と、

前記処理手段により処理した処理済みデータを後段の処理ブロックに出力するとともに、前記後段の処理ブロックに対して所定時間幅のクロックイネーブル信号を出力する出力手段と、

前記処理手段により前記入力されたデータの処理が完了した後、前記出力手段による前記処理済みデータの出力が完了すると前記クロック発生手段による前記動作クロック信号の発生を停止させる停止手段と、
を有することを特徴とする。

40

【0012】

上記目的を達成するために本発明の一態様に係る電子機器の電力制御方法は以下のような工程を備える。即ち、

クロック信号に同期して動作する複数の処理ブロックを具備する電子機器の電力制御方法であって、

前記複数の処理ブロックのそれぞれにおいて、外部或は前段の処理ブロックからデータとともに入力される所定時間幅のクロックイネーブル信号に応じて当該処理ブロックの動作クロック信号を発生するクロック発生工程と、

50

前記動作クロック信号により、前記複数の処理ブロックのそれぞれにおいて入力された前記データを処理する処理工程と、

前記処理工程で処理した処理済みデータを後段の処理ブロックに出力するとともに、前記後段の処理ブロックに対して所定時間幅のクロックイネーブル信号を出力する出力工程と、

前記処理工程で、前記入力されたデータの処理が完了した後、前記出力工程での前記処理済みデータの出力が完了すると前記クロック発生工程での前記動作クロック信号の発生を停止させる停止工程と、
を有することを特徴とする。

【発明の効果】

10

【0013】

本発明によれば、データを処理しているブロックの動作クロック信号だけが供給された状態とし、他のデータが存在しないブロックでは動作クロック信号を停止状態にすることにより、電子機器全体の消費電力を低減できる。

【発明を実施するための最良の形態】

【0014】

以下、添付図面を参照して本発明の好適な実施の形態を詳しく説明する。尚、以下の実施の形態は特許請求の範囲に係る本発明を限定するものでなく、また本実施の形態で説明されている特徴の組み合わせの全てが本発明の解決手段に必須のものとは限らない。

【0015】

20

図2は、本発明の実施の形態に係る多機能処理装置(MFP)の制御部224の構成例を示すブロック図である。この多機能処理装置は、FAX機能、コピー機能、プリンタ機能、ストレージ機能などを備える。尚、この実施の形態では、本発明の電子機器の一例としてMFPを例にして説明するが、本発明はこのような装置に限定されるものでなく、例えば家庭用電気製品、PC等の情報処理装置、通信装置、印刷装置等のように商用電力或は電池等からの電力供給を受けて動作する機器や装置全般に適用できる。

【0016】

CPU201は、ROM203或はRAM204にロードされたプログラムに従って、このMFPの動作制御や、各種レジスタの設定や演算処理等を行う。メモリ制御部202は、ROM203とRAM204への入出力制御やDMA(ダイレクトメモリアクセス)制御を行う。ROM203は不揮発性メモリで、プログラムやフォント等の変更しないデータを格納する。RAM204はSDRAMやDDRに代表される揮発性メモリで、プログラムの作業領域や印刷データの格納領域等の書き換えが必要な用途に用いられる。

30

【0017】

ネットワークI/F部205は、ネットワーク207との間のインターフェイスを制御しており、一般的には、TCP/IPのプロトコルに対応している。このMFPは、ネットワーク207を介してホスト機器206などのネットワーク対応機器と接続され、そのホスト機器206から印刷データを受信して印刷するネットワークプリンタとしても機能している。パネルI/F部208は、操作パネル209との間での通信制御を行う。操作パネル209は、液晶表示部や各種操作ボタン等を配しており、ユーザにより操作されて各種設定を行うUIとして機能している。

40

【0018】

リーダI/F部210は、スキャナ211との間の通信制御を行い、スキャナ211でスキャンした原稿の画像データを入力してコピー機能を実現する。FAXI/F部212は、通信回線を介してFAX装置213との間の通信制御を行い、例えば電話回線に接続されたFAX装置213との間でFAXデータの送受信処理を実行する。外部高速I/F部214は、外部ボード215とPCI-Express等の高速インターフェイスで接続され、印刷データの送受信処理を行う。画像処理部216は、ネットワークI/F部205、リーダI/F部210、FAXI/F部212、外部高速I/F部214を介して取り込んだ画像データに対して、ディザ処理やエッジ処理等の画像処理を行う。尚、この画像処理部

50

216による処理は、画像データの圧縮及び伸張処理も含む。

【0019】

HDD制御部217は、HDD（ハードディスク装置）218に対するATA規格（Parallel-ATAやSerial-ATA等）でデータの入出力制御を行う。HDD218は不揮発性の大容量記憶装置であり、ファイルの保存や印刷データの一次格納場所として使用される。ビデオ制御部219は、プリンタ部220との間のコマンド／ステータスの通信や、RAM204に生成された印刷データをプリンタ部220に送信する。プリンタ部220は主にビデオ制御部219からのコマンド情報に従って印刷データに基づく印刷を行って用紙に画像を印刷する。高速バスI/F部221は、画処理チップ222を接続してデータの入出力を制御する。例えば、図2の点線で囲んだ制御部224を、CPUコアを含むメインチップとした場合、内部の画像処理部216を補助する処理や他の画像処理を行うチップ222を高速バスI/F部221に接続する。これにより、その画処理チップ222による処理が可能となる。システムバス223は、制御バス及びデータバスと任意ブロック間のローカルなバス及び信号線を便宜的にまとめて表現したものである。

10

【0020】

図3は、本実施の形態に係る制御部の一部を更に具体的に示した図である。図3の点線より内側が制御部（基板）224を示している。尚、図3において図2と共通する部分は同じ記号で示している。

【0021】

メインチップ301はメイン制御チップで、複数のIPモジュールで構成されるSoC（System on a Chip）である。このメインチップ301で示す内部構成はSoCの一例を示している。以下、このメインチップ301内部構成について簡単に説明する。

20

【0022】

B2R303は、ブロック化されたデータをラスタデータに変換する。R2B304は逆に、ラスタデータをブロックデータに変換する。B2R303及びR2B304は、ブロック単位でデータの受け渡しを行うパケット処理に用いられる。JPEG-E305は、JPEG（joint photographic experts group）形式での静止画像データを圧縮する。また及びJPEG-D306は、JPEGで圧縮された画像データを伸張する。メモリ制御部202は、ROM203、RAM204等のメモリとの間のデータのやり取りを制御する。画像処理部216は、ディザ処理、スクリーン処理、スムージング処理など画像に対する各種画像処理を行う。外部高速インタフェース部214は、例えばPCIe I/Fで、PCI-Express規格の高速シリアルバス制御を行い、シリアルバス及びコネクタを介して外部ボード215と接続されている。HDD制御部217は、Serial-ATA規格の高速シリアルバス制御を行い、専用ケーブルを介して、HDD218と接続されている。FAX I/F212は、通信回線を介してFAX装置315と接続されており、FAX装置213との間の通信制御を行う。高速バスI/F221は、基板上の他のチップとバス接続してデータの入出力制御を行い、Pチップ317、Iチップ319、Sチップ320等のサブチップ群と接続されている。

30

【0023】

Pチップ317は、メインチップ301から供給される印刷データを、更に画像処理（加工整形）して最終的な印刷データを作成し、その印刷データをプリンタ部220に送信して印刷する。Iチップ319は、メインチップ301の画像処理部216を補助及び拡張する各種画像処理を行う。Sチップ320は、スキャナ211で読み込まれた画像データを画像処理（加工整形）を行う。その他422は、その他の省略した機能ブロックを示す。バススイッチ323は、上述した各部同士をバス接続するセレクト機能をも有し、このバススイッチ323によって、所望の処理を実行すべく各部同士の適切な接続が選択される。

40

【0024】

図4は、FAXデータを受信する際のメインチップ301でのデータの流れを説明する図である。ここでは各矢印に添付された括弧付きの数字の順でデータの受け渡しが行われ

50

る。以下、FAX装置の受信データを印刷するまでのデータフローを説明する。尚、図3と共通する部分は同じ記号で示している。

【0025】

FAX装置213で受信されたFAXデータは、逐次RAM204にバッファリングされる(矢印(1))。また同時に、RAM204にバッファリングされたデータは、HDD218へ転送されてスプールされる(矢印(2))。こうして受信した全てのFAXデータをHDD218にスプールするまで処理が継続される。こうしてHDD218へのスプールが完了すると、HDD218から再びRAM204に、そのFAXデータがページ単位で読み出され(矢印(3))、印刷データの作成が開始される。RAM204に読み出されたFAXデータは、CPU201によって所定の加工が施される(矢印(4))。こうして加工されたデータは画像処理部216に転送される(矢印(5))。画像処理部216は、その受け取ったデータに対して所定の画像処理を行い、再びRAM204に書き戻す(矢印(6))。この書き戻されたデータはページ単位で、外部の画処理チップであるPチップ317に転送され、所定の画像処理が施された後、プリンタ部220に送られて印刷される(矢印(7))。こうして上記の矢印(3)~(7)で示す処理は、HDD218にスプールされたページ分の画像データに対する処理が完了するまで繰り返される。

10

【0026】

ここで、図4の211, 214, 215, 303~306, 319, 320, 322で示す部分は、何も処理をしていない。また、上述したFAX受信データ処理では、区間(1)~(2)と区間(3)~(7)の処理は、時間的に分離していることがわかる。即ち、区間(1)~(2)でFAX受信データをHDD218にスプールしている間は、同時に区間(3)~(7)で動作する回路による処理の必要がない。

20

【0027】

更に、画像処理部216は各種画像処理機能を含んでいるが、FAXデータに対する処理ブロック以外(例えば、ディザ処理やスクリーン処理、スムージング処理等)は動作させる必要がない。

【0028】

詳細は後述するが、更に突き詰めると、FAXデータに画像処理を行う区間(5)~(6)でも、画像処理部216による各処理の間にもデータの受け渡しのための待ち時間が発生する区間がミクロ的に見ると多量に含まれる。その理由の一つとしては、全てを最適化してパイプライン処理化を行うことは困難であることが挙げられる。

30

【0029】

そこで本実施の形態では、現在の処理に関係しない全ての回路、更には、処理中の回路であっても、時間的に動作する必要のない部分へのクロックの供給を止めるとともに、クロックの供給を遮断したい任意の範囲に対して適応可能なクロック制御方法を提供する。これにより、スタンバイ時のみばかりでなく、印刷時も含んで消費電力を削減することができる。また、各システムに応じた最適な適応範囲で、比較的容易にクロック制御を行うことを特徴としている。

【0030】

以下、図を参照して本実施の形態の詳細を説明する。

40

【0031】

図5は、本実施の形態の基本的な構成である処理ブロック間の関係を示した図である。ここでは例えば、図4の画像処理部216の内部を想定している。各処理ブロックA501~ブロックD504は、最適な処理単位にブロック化されている。ここで各ブロックは、50万~100万ゲート規模の複雑な処理を行うものと想定しても良く、或は数百~数千ゲート規模で比較的単純な処理を行うものと考えてもよい。

【0032】

各ブロックには、共通にリセット信号(System_Reset)とクロック信号(CLK)が入力されている。また、各ブロックに接続されているPre_CLK_EN*_*信号は、ブロック内のクロックのオン/オフを制御するイネーブル信号である(オン:発振許可)。EN*の*の

50

部分に数字をいれた E N 1 及び E N 2 は、機能別のグループを表している。

【 0 0 3 3 】

実線のイネーブル信号で接続されるブロック A 5 0 1、ブロック B 5 0 2、ブロック C 5 0 3 は連携して機能 1 の処理を行う。ここで、ブロック A 5 0 1 と接続される前段のブロックと、ブロック C 5 0 3 に接続される後段のブロックは省略している。図示しないが、この機能 1 を実現するのに必要なブロックが前後に接続されていることを前提として考えてもらえればよい。尚、図 5 において、各処理ブロックに入力されるクロックイネーブル信号 (Pre_CLK_EN1_A(B,C)) は所定時間幅の信号で、そのブロックにおける機能 1 の処理の開始を指示している。尚、このクロックイネーブル信号の時間幅は、各ブロックに対して共通に固定でも良く、或はそのブロックの処理に応じて、それぞれ異なる時間幅としても良い。

10

【 0 0 3 4 】

同様に、一点破線のイネーブル信号で接続されるブロック A 5 0 1、ブロック D 5 0 4 は、連携して機能 2 の処理を行う。この回路でブロック A 5 0 1 は、機能 1 及び機能 2 に共通なブロックである。また各ブロックに入力されるクロックイネーブル信号 (Pre_CLK_EN2_A(D)) は、そのブロックにおける機能 2 の処理の開始を指示している。

【 0 0 3 5 】

また各ブロック間を太線矢印で接続している信号線は、データバス (Data_BUS) を示す。本実施の形態では、クロックイネーブル信号とデータの流れが本質であるため、その他の制御信号は省略してある。また、ここでは図示しないが、当然ながら各ブロック間には、データを受け渡すためのタイミングを計る制御信号が接続されている。図 5 に示すデータの流れは、各ブロック間に接続されるクロックイネーブル信号を伴って単純に左から右に流れて行く例で示している。

20

【 0 0 3 6 】

次にクロックイネーブル信号の生成タイミングについて説明する。

【 0 0 3 7 】

図 6 は、図 5 の各ブロック A ~ B のそれぞれに含まれるクロック制御回路の状態遷移を示した図である。ここで基本ステートは 3 つの状態構成される。

【 0 0 3 8 】

状態 S 0 (6 0 1) はスタンバイ状態 (電力消費量が減少した状態) を示し、ここではクロックがオフの状態である。

30

【 0 0 3 9 】

状態 S 1 (6 0 2) は初期化 (Init) 状態を示し、リセット信号がイネーブルになった時の状態である。ここでリセット信号とは、システムリセットや各ブロックの内部初期化を行うローカルなリセット信号 (図 5 の例では、System-Reset 信号) も含む。

【 0 0 4 0 】

状態 S 2 (6 0 3) は実行中 (Active) を示し、該当ブロックにクロック信号が供給されて処理中 (アクティブ) の状態である。

【 0 0 4 1 】

状態 S 0 と状態 S 1 との間の遷移条件は、リセット信号で規定される。ブロックに対するリセット信号がイネーブルになると内部信号リセット = 1 となり、状態 S 0 から状態 S 1 に遷移する。状態 S 1 では、そのブロック内の全てのクロックゲートがオープンとなり内部初期化処理が実行される。尚、ここでは、リセット期間中にも内部初期化に十分な期間で、動作クロック信号が供給されることを前提としている。リセット信号がディスイネーブルになると内部信号リセット = 0 となり、状態 S 1 から状態 S 0 に遷移する。状態 S 0 では、全てのクロックゲートがクローズして、動作クロック信号が供給されない停止状態 (待機状態) となる。この停止状態 (待機状態) では、消費電力量が減少される。

40

【 0 0 4 2 】

状態 S 0 と状態 S 2 との間の遷移条件は、CLK_EN 信号で規定される。ブロックに対するクロックイネーブル信号がイネーブルになると内部信号 CLK_EN = 1 となり、状態 S 0 から

50

状態 S 2 に遷移する。状態 S 2 では、イネーブルになったクロックイネーブル信号に対応する部分のクロックゲートがオープンとなって動作クロック信号が供給され所定の処理が実行される。ここで、クロックイネーブル信号に対応することの意味を説明する。

【0043】

図 5 で説明したように、ブロック A 5 0 1 は、機能 1 と機能 2 の共通ブロックである。然しながら、このことは機能 1 及び機能 2 の両方にブロック A 5 0 1 の回路全てを利用することを意味しない。ブロック A 5 0 1 に入力される Pre_CLK_EN1 に対応する回路を動作させる時、Pre_CLK_EN2 に対応する、機能 1 には不必要な部分の回路の動作を停止させても支障はない。即ち、図 5 のブロック A 5 0 1 ～ブロック D 5 0 4 の処理単位に分割したのと同様に、ブロック A 5 0 1 の内部も機能 1 及び機能 2 の処理内容に応じて、更に最適化した処理単位に分けても良いことを意味している。但し、処理単位に分けた各ブロックには、図 6 に示すクロック制御回路（ステートマシン）が必要である。

【0044】

状態 S 2 において全ての処理単位が終了し、次段のブロックに対してデータを送信完了した時点で内部信号 CLK_EN = 0 となる。これにより、状態 S 2 から状態 S 0 に遷移して再び待機状態（クロックがオフ状態）になる。

【0045】

図 1 は、本実施の形態に係る図 5 に示す処理ブロックのそれぞれのクロック信号及びデータバスの状態の一例を示すタイミングチャートである。ここでは説明の便宜上、信号名を図 5 とは一部変更している。以下、各信号について説明する。

【0046】

CLK 信号は、外部より定常的に供給されているクロック信号である。Pre_CLK_EN_Block* 信号は、前段ブロックから該当ブロック（Block*）に供給されるクロックイネーブル信号である。このクロックイネーブル信号がハイレベルの間、そのブロックの内部の動作クロック信号 CLK_Block* が生成される。ここで Block* は、図 5 のブロック A 5 0 1、B 5 0 2、C 5 0 3、D 5 0 4 に対応している。CLK_Block* 信号は、実際に、そのブロック（Block*）の動作に使用される動作クロック信号を示す。IN_CLK_EN_Block* 信号は、ブロック（Block*）のクロックイネーブル信号で、この信号と Pre_CLK_EN_Block* 信号のいずれかがハイレベルの間、CLK 信号がそのブロックに入力されて動作クロック信号となる（図 8）。IN_Data_Block* 信号は、前段ブロックから、Block* で表されるブロックへ入力されるデータである。OUT_Data_Block* 信号は、Block* で表されるブロックから後段のブロックへ出力するデータを示す。入力データ IN_Data_Block* 及び出力データ OUT_Data_Block* の有効期間を、図 1 では白抜きで表している。尚、この白抜き以外の入力データと出力データ（処理済みデータ）の区間は不定期間である。

【0047】

次にブロック B 5 0 2 を例に、動作クロック信号の生成タイミング及びデータの受け渡しタイミングについて説明する。

【0048】

ブロック A 5 0 1 は、出力データ OUT_Data_BlockA の出力準備が整う直前にブロック B 5 0 2 へのクロックイネーブル信号 Pre_CLK_EN_BlockB をイネーブルにする（Pre_CLK_EN_BlockB = 1）。ここで、クロックイネーブル信号 ***_CLK_EN_BlockB は、CLK 信号の立下りエッジに同期して立ち上がり、その他の内部処理も全て立ち上がりエッジに同期している。当然ながら、必ずしも立下りエッジを使用する必要はなく、全てを立ち上がり（又は立下り）エッジに同期させるように設計してもよい。

【0049】

ブロック B 5 0 2 の内部では、クロックイネーブル信号によって動作クロック信号 CLK_BlockB の発生を開始する。そしてブロック A 5 0 1 とブロック B 5 0 2 との間のデータ受け渡し制御信号のハンドシェイク後、ブロック B 5 0 2 は入力データ IN_Data_BlockB を受け取ると同時に、ブロック B 5 0 2 での処理を開始する。

【0050】

10

20

30

40

50

またブロック B 5 0 2 では、クロックイネーブル信号 Pre_CLK_EN_BlockB が立ち上がった後、所定時間遅れて、ブロック B 5 0 2 の内部のクロックイネーブル信号 IN_CLK_EN_BlockB をハイレベルにしている。

【0051】

ここで、ブロック B 5 0 2 の動作クロック信号 (CLK_BlockB) が出力される時間について説明する。

【0052】

ブロック B 5 0 2 の動作クロック信号 CLK_BlockB は、Pre_CLK_EN_BlockB がハイレベルになると出力を開始し、IN_CLK_EN_BlockB がロウレベルになると、その出力を停止する。即ち、IN_CLK_EN_BlockB = 1 の条件が成立している間は、ブロック B 5 0 2 での処理が実行中であることを示している。そして、この処理によりブロック B 5 0 2 で処理・生成された処理済みデータ OUT_Data_BlockB が、次段のブロック C 5 0 3 へ転送される。この処理済みデータ OUT_Data_BlockB の転送が完了する、少なくとも一つの処理単位期間の間、ブロック B 5 0 2 の内部クロック信号 CLK_BlockB は発振を継続している。これを 1 単位処理期間とする (図 1 の 1 0 3 で図示)。尚、図 1 では、ブロック C 5 0 3 での 1 単位処理期間を 1 0 3 で示しているが、ブロック A 5 0 1, ブロック B 5 0 2 の 1 単位処理期間も同様とする。

【0053】

こうしてブロック B 5 0 2 の 1 単位処理期間が終了した直後 (1 0 2) の動作クロック信号 CLK_BlockB の立ち上げエッジで、ブロック A 5 0 1 からのクロックイネーブル信号 Pre_CLK_EN_BlockB の有無を判定する。このクロックイネーブル信号がハイレベルの場合には、引き続き次の 1 単位処理期間、動作クロック信号 CLK_BlockB の発生が延長される。一方、ロウレベルである場合は、この 1 単位処理期間 (図 1 の例では、1 0 2 の次のクロック信号立下りエッジ) で動作クロック信号 CLK_BlockB の発生を終了してブロック B 5 0 2 の動作クロック信号は停止状態 (待機状態) になる。

【0054】

以下同様に、クロックイネーブル信号とデータがブロック B 5 0 2 からブロック C 5 0 3 へ、またブロック C 5 0 3 から次のブロックへと順次渡されて行く。即ち、各ブロックから出力されるデータにクロック信号が付随して転送されることにより、あるブロックにおける動作クロック信号の発生が次のブロックに伝播される。従って、有効データが存在しない期間では、各ブロックの動作クロック信号は自動的に発振を停止していることになる。こうして各ブロックにおいて消費電力を抑えることが可能になる。

【0055】

上述した本実施の形態の特徴を 2 点挙げると以下のようなになる。

【0056】

(1) 第 1 の点は、図 6 を参照して説明したように、本実施の形態は簡単なクロック制御回路で実現可能である。昨今の A S I C 設計は、H D L (ハードウェア記述言語) を用いた設計が主流であるため、適応したいブロックに対してクロック制御回路 (ステートマシン) をコピー & ペーストで簡単に作成することができる。

【0057】

但し、図 1 でブロック A 5 0 1 からブロック B 5 0 2 へデータを受け渡す期間で、特にクロックイネーブル信号 Pre_CLK_EN_BlockB をハイレベルにしてから、ブロック B 5 0 2 がデータを確実に受取れるように区間 1 0 1 を確保する必要がある。この区間 1 0 1 は、高速性を要求される場合には、各ブロック間で最適化する必要がある。また処理速度に余裕があるブロックに対しては、区間 1 0 1 を、データを受け取れる十分な時間として一律に固定にしてもよい。

【0058】

(2) 第 2 の点を図 7 のタイミングチャートを用いて他のタイミング例で説明する。ブロック構成、信号及び基本的な制御方法は、図 1 で説明した場合と同じであるため省略する。

10

20

30

40

50

【0059】

図7は、本実施の形態に係る図5に示す処理ブロックのそれぞれのクロック信号及びデータバスの状態の他の例を示すタイミングチャートである。ここでは説明の便宜上、信号名を図5とは一部変更している。以下、各信号について説明する。

【0060】

図7では、ブロックA501の入力データIN_Data_BlockAの内容によって次のブロックへのデータ出力タイミングが変更される例を示している。

【0061】

図7の例では、ブロックA501は、入力データに対して2つのデータ(Data1, Data2)を出力する。この時、前段から入力されるデータの内容に応じて出力データOUT_Data_BlockAのData1とData2の出力間隔700が不定となる。 10

【0062】

これらデータ(Data1, Data2)を後段のブロックB502及びブロックC503が処理する時間(固定)は図に示す通りである。この場合、ブロックA501からのデータが出力された時、ブロックC503での処理タイミング701で注意を要する。このタイミング701では、ブロックC503がData1に対する1単位処理を完了している。この時、ブロックBからのクロックイネーブル信号Pre_CLK_EN_BlockCがロウレベルであるため、ブロックC503は継続データなしと判定して内部クロックイネーブル信号IN_CLK_EN_BlockC=0にする。しかしながら、その半クロック後に再びPre_CLK_EN_BlockCがハイレベルとなってデータData2が供給される。従って、このデータData2を処理するために、タイミング701の後、破線で示したクロック信号が喪失しないように考慮して設計しなければならない。図7の例では、クロックイネーブル信号をCLK信号の立下りエッジに同期して切り換え、その他の信号をCLK信号の立ち上がりエッジに同期させている。これにより、タイミング701の後でクロック信号の喪失が発生しなくなる。 20

【0063】

図7の例において、図中の区間700が、図7の場合よりも長い場合は特に問題は発生しない。また逆に、区間700が、図7の場合よりも短い場合は、ブロックC503がブロックB502からのData2を連続的に処理できないため、ブロックC503での待ち時間が発生することになるが、これも特に問題はない。 30

【0064】

図8は、本実施の形態に係るゲーテッドクロック部(クロック発生部)の回路構成例を示す図である。 30

【0065】

OR回路801には、前段のブロックからのクロックイネーブル信号Pre_CLK_EN_Block*と該当ブロック内部で生成されるクロックイネーブル信号IN_CLK_EN_Block*が入力される。ここで、IN_CLK_EN_Block*信号は、図6で説明したクロック制御回路(ステートマシン)からの内部信号である。クロックイネーブル信号IN_CLK_EN_Block*の発生要因には、リセット信号(初期化時)とPre_CLK_EN_Block*によるものとがあるが、いずれも同期化された信号であることを前提としている。このOR回路801の出力信号と、外部クロック信号Ex_CLK(CLK)との論理積の結果がAND回路802から出力され、これが動作クロック信号CLK_block*となる。 40

【0066】

ここまでは、図5の処理ブロックを用いて本実施の形態の基本的な部分を説明したが、更に図9～図12を参照して他の接続パターンについて説明する。基本的な構造及び信号名等は、図5と同様であるため省略する。

【0067】

図9は、本実施の形態の図5に示す処理ブロック間の変形例を示す図である。

【0068】

機能1(function1)において、ブロックA501からブロックB502へデータが供給され、ブロックB502での処理結果を再びブロックA501への入力データとして返 50

している。この構成例では、ブロック A 5 0 1 からブロック B 5 0 2 へのクロックイネーブル信号 Pre_CLK_EN1_B と、ブロック B 5 0 2 からブロック A 5 0 1 へのクロックイネーブル信号 Pre_CLK_EN1_A とが接続される。また機能 2 では、ブロック A 5 0 1 からブロック C 5 0 3 へデータとクロックイネーブル信号 Pre_CLK_EN2_C が供給されている。

【0069】

この時、図 5 と同様に、ブロック A 5 0 1 が Pre_CLK_EN1_A に関する回路と、Pre_CLK_EN2_A に関する回路とを有し、それらが独立に動作する場合は、それぞれ独立に必要な場合にのみクロック信号を供給すれば良いことは言うまでもない。

【0070】

図 10 は、本実施の形態の図 5 に示す処理ブロック間の他の変形例を示す図である。

【0071】

ブロック A 5 0 1 ~ ブロック C 5 0 3 が共通バスデータバス 1 0 0 1 に接続されている。但し、各ブロックに接続されるバス 1 0 0 2 ~ 1 0 0 4 は、3 ステートバス、或はリードバスとライトバスが独立に分離して接続されるバスのどちらでも良いが、煩雑となるためまとめて記載している（後述する図 11、図 12 も同様）。本実施の形態では、クロックイネーブル信号が、各ブロックのセレクト信号と考えることもできる。従って、図 10 の例では、データを送信したいブロックに対するクロックイネーブル信号 Pre_CLK_EN1_(A, B, C) をイネーブル (= 1) にして、データバス 1 0 0 1 に対してデータを出力すればよい。これにより、図 10 の破線矢印で示すように、選択されたブロックが、そのデータバス上のデータを入力データとして取り込む。

【0072】

図 11 は、基本的に図 9 と同じ処理ブロックで構成され、ブロック A 5 0 1 からブロック B 5 0 2 へ入力したデータを、ブロック B 5 0 2 が処理した後に再びブロック A 5 0 1 に戻す構成図である。

【0073】

ブロック A 5 0 1 ~ ブロック C 5 0 3 のデータバスへの接続は、図 10 と同様にデータバス 1 0 0 1 に共通で接続されている。ブロック A 5 0 1 がブロック B 5 0 2 から受け取ったデータはブロック A 5 0 1 で処理され、次段のブロック C 5 0 3 への入力データとして渡される。図 11 の破線矢印はデータの流れを示している。各出力データに先立って、ブロック間に接続されているクロックイネーブル信号 Pre_CLK_EN1_** をイネーブル (= 1) にすることは前述の場合と同じである。

【0074】

図 12 は、図 10 と同様のブロック配列において、データバス 1 0 0 1 に流れるデータ形式がパケットデータ 1 2 0 1 である点が前述の図面との構成とは異なっている。

【0075】

このパケットデータ 1 2 0 1 は、データ部 1 2 0 2 とブロック ID 1 2 0 3 とで構成されている。このパケットデータ 1 2 0 1 を受信した各ブロックは、そのブロック ID 1 2 0 3 が自ブロックの ID と一致した場合にのみ、そのパケットデータを入力する。図 12 の破線矢印は、データの流れを示す。当然ながら、この場合にもパケットデータの出力に先立ってブロック間に接続されているクロックイネーブル信号 Pre_CLK_EN1_** 又は Pre_CLK_EN2_** をイネーブル (= 1) とする。

【0076】

ここまでの説明では、チップ (ASIC) 内部について本実施の形態を適用する例を示した。ここからは、コントローラ基板上の各チップ間に本実施の形態を適用した場合の例を説明する。

【0077】

図 13 は、本実施の形態において、メインチップに 2 つのサブチップが接続されている構成例を示す図である。

【0078】

ここでは、外部からの起点となる制御信号及び入力データは、全てメインチップ 1 3 0

10

20

30

40

50

1が受け取り、このメインチップ1301が処理したデータを更にサブチップ1302及びサブチップ1303へと渡して、印刷データを加工することを想定している。これらサブチップ1302、1303に続くチップは、他のチップとして省略している。また外部からの起点となる信号とは、同じコントローラ基板上の他の回路や図3の外部ボード215、スキャナ211、FAX装置213などを示している。外部からの起点となる制御信号及び入力データを、以後起点信号と呼ぶこととする。図13では、起点信号をInput1~Input3で表している。

【0079】

更に各チップ内部を示すブロックは、処理単位に、更にブロックに分割されており、その分割したブロックのサイズは処理機能に応じて適当でよい。また図に示すブロックはトップブロックを示し、当然ながら各ブロック内部は更に階層的にブロックに分割して本実施の形態を適用しても構わない。また各ブロック内部に示したG1、G2、G3、G4の記号は、機能別のグループを示す。従って、図13の例では、4つの機能を持つことが分かる。また各処理ブロックへのリセット信号やクロック信号、ブロック間の制御信号やデータバス等は、説明の本質とは関係ないため全て省略し、ブロック間を繋ぐ信号線はクロックイネーブル信号のみを明示的に記載している。特にチップ間を結ぶクロックイネーブル信号は、信号名も記載している。(CLK_EN1~4: 数字は各機能に対応)

更に外部から起点信号(Input1~Input3)を受けるメインチップ1301のブロックは、ブロック1310~1312で示している。また、斜線を付したブロックの内部では、クロック信号の発振が停止している。また白抜きのブロックは、処理中のブロックを表している。

【0080】

図13の例では、G1ブロック1310に、外部よりInput1の起点信号が入力されることにより、メインチップ1301の機能1(G1)の処理が開始される。このため、各メインチップ1301のG1ブロックは全て白抜きで示している。上述したように、データに伴ってクロック信号の発振がオン/オフを繰り返しながら末端のG1ブロック1313に到達する。

【0081】

この末端のG1ブロック1313の先は、メインチップ1301の外部に出て、クロックイネーブル信号CLK_EN1がサブチップ1302のG1ブロック1314及びサブチップ1303のG1/G2ブロック1315(図5で説明したように、機能1と機能2の共通ブロックであることを意味する)に入力される。これによって末端のG1ブロック1313から出力されたデータは、G1ブロック1314とG1/G2ブロック1315へ渡される。ここで例えば、G1ブロック1313は、連続的にデータを出力し、その奇数データをG1ブロック1314が、偶数データをG1/G2ブロック1315が受け取る場合などが考えられる。

【0082】

こうしてデータを受け取ったサブチップ1302及びサブチップ1303内部では、メインチップ1301と同様に、クロックイネーブル信号を伝播してデータを処理する。各サブチップ内の末端G1ブロック1316とG1ブロック1317に到達すると、各クロックイネーブル信号は、チップ外部に出力され他のチップ等へ伝達されて行く。

【0083】

繰り返しになるが、処理中として白抜きで活性化していることを表すブロックも、処理すべきデータが無い区間(データが通過した後)は、クロック信号の発振は停止している点が本実施の形態の特徴である。

【0084】

更にサブチップ1302及びサブチップ1303で図示しているように、サブチップ1302の末端G1ブロック1316、G2ブロック1318からチップ外へ接続されるクロックイネーブル信号は他のチップに接続されている。またサブチップ1302の末端G4ブロック1319からチップ外へ接続されるクロックイネーブル信号は、コントローラ

基板上の他の回路に接続される。更に、サブチップ 1303 の末端 G1 ブロック 1317 からチップ外へ接続されるクロックイネーブル信号は、他のボードに接続される。ここでは図示しないが、本実施の形態の適応範囲は、説明してきたチップ内部及びチップ間に留まらず、基板上の他の回路、外部ボード、外部装置にまで同じ原理で適応することが可能である。また本実施の形態では、印刷装置の制御システムを例に説明したが、一般的に任意のデジタル回路に対して適応可能であることは言うまでもない。

【0085】

なお、各機能別 (G1, G2, G3, G4) にメインチップ 1301 から始まり、サブチップ 1302 及びサブチップ 1303 へとクロックイネーブル信号のチェーンが繋がっているが、途中のサブチップ内でグループの番号が変わる箇所は、処理を共有していることを意味している。

10

【0086】

図 14 は、本実施の形態に係るチップ間のクロックイネーブル信号を接続する他の例を説明する図である。

【0087】

図に示すように、メインチップ 1401 の終端ブロック (G1, G2, G3, G4) からのクロックイネーブル信号をエンコード (Pre_CLK_EN ENC) 1403 して送信する。一方、受信側のサブチップ 1402 は、そのクロックイネーブル信号を受信してデコード (Pre_CLK_EN ENC) 1404 することにより、サブチップ 1402 内部の先頭ブロック (G1, G2, G3, G4) にクロックイネーブル信号を分配している。図 11 では、機能 2 に対応するブロックのみが処理を実行しており、その他の機能を実行するブロックの動作が停止している。

20

【0088】

図 15 は、図 13 と構成は同じで、スタンバイ時の状態を表している。

【0089】

基点信号 Input1~Input1 を受けるメインチップ 1301 の G1 ブロック 1310、G2 ブロック 1311、G3 ブロック 1312 のみが、常にクロック信号が発振している状態 (白抜きブロック) である。そして、その他のブロック (メイン/サブチップのいずれでも) は、クロック信号の発振が停止している状態 (斜線付きブロック) であることがわかる。即ち、スタンバイ状態で、直に基点信号により動作可能な状態であっても、消費電力を最小限に抑えることができる。

30

【0090】

以上、ここまでの説明で分かるように本実施の形態では、処理単位である各ブロックは自己の処理に専念し、相手とのハンドシェクには関知しない構造を基本としている。このため、各ブロックをクロックイネーブル信号で数珠繋ぎに接続 (ディジーチェーン) するだけで良く、非常にシンプルな構造にできる。従って、本実施の形態の適用範囲は、チップ内部のみに留まらず、他のチップ、オンボードの回路、外部ボード、外部装置間にも適用可能である。

【0091】

更にスタンバイ時のみならず、印刷時等の処理時実行中であっても、処理すべきデータが存在している区間のみクロック信号が生成され、その他の区間では、クロック信号が供給されない状態にできる。これにより、印刷装置の待機電力及び稼動電力を含めて消費電力を抑えることが可能となる。

40

【0092】

一般的には、クロック信号を停止するよりも電源をオフの方が電力消費を抑えられることは言うまでもないが、この場合には電源オンする度に初期化処理が必要となり、この処理時間が処理パフォーマンスとのトレードオフとなる。また ASIC 内部の回路の小さなブロック単位で、電源供給をオン/オフすることは技術的にも難しい。即ち、本実施の形態では、トータルの消費電力の抑制と処理パフォーマンスとのバランスを取った方法であると言える。

50

【0093】

また、クロック信号の供給停止処理をソフト的に処理する方法は一般的に行われる方法であるが、この場合にクロック信号をオン／オフ制御できる時間幅は、約数ms～数十ms程度である。

【0094】

これに対して本実施の形態では、簡単な構成のクロック制御回路を使用し、それを含む各処理単位のブロック間をクロックイネーブル信号によってディジーチェーンで接続することで処理を行うことができる。このため、クロック信号を制御する時間幅は、クロック信号の周期のオーダ（ns（10の－9乗秒）やps（10の－12乗秒））で処理可能であり、電力削減の効率を最大限に引き出すことが可能となる。

10

【0095】

このような本実施の形態によれば、回路構成が簡単で、且つ処理ブロック間をクロックイネーブル信号でディジーチェーンに接続するだけで、チップ内部のみに留まらず、他のチップ、オンボードの回路、外部ボード、外部装置間など広範囲に適用可能であり、システム全体として消費電力を抑えることが可能となる。

【図面の簡単な説明】

【0096】

【図1】本実施の形態に係る図5に示す処理ブロックのそれぞれのクロック信号及びデータバスの状態の一例を示すタイミングチャートである。

【図2】本発明の実施の形態に係る多機能処理装置（MFP）の制御部の構成例を示すブロック図である。

20

【図3】本実施の形態に係る制御部の一部を更に具体的に示した図である。

【図4】FAXデータを受信する際のメインチップでのデータの流れを説明する図である。

【図5】本実施の形態の基本的な構成である処理ブロック間の関係を示した図である。

【図6】図5の各ブロックA～Bのそれぞれに含まれるクロック制御回路の状態遷移を示した図である。

【図7】本実施の形態に係る図5に示す処理ブロックのそれぞれのクロック信号及びデータバスの状態の他の例を示すタイミングチャートである。

【図8】本実施の形態に係るゲーテッドクロック部の回路構成例を示す図である。

30

【図9】本実施の形態の図5に示す処理ブロック間の変形例を示す図である。

【図10】本実施の形態の図5に示す処理ブロック間の他の変形例を示す図である。

【図11】基本的に図9と同じ処理ブロックで構成され、ブロックA501からブロックB502へ入力したデータを、ブロックB502が処理した後に再びブロックA501に戻す構成図である。

【図12】図10と同様のブロック配列を示し、データバスに流れるデータ形式がパケットデータである例を示す図である。

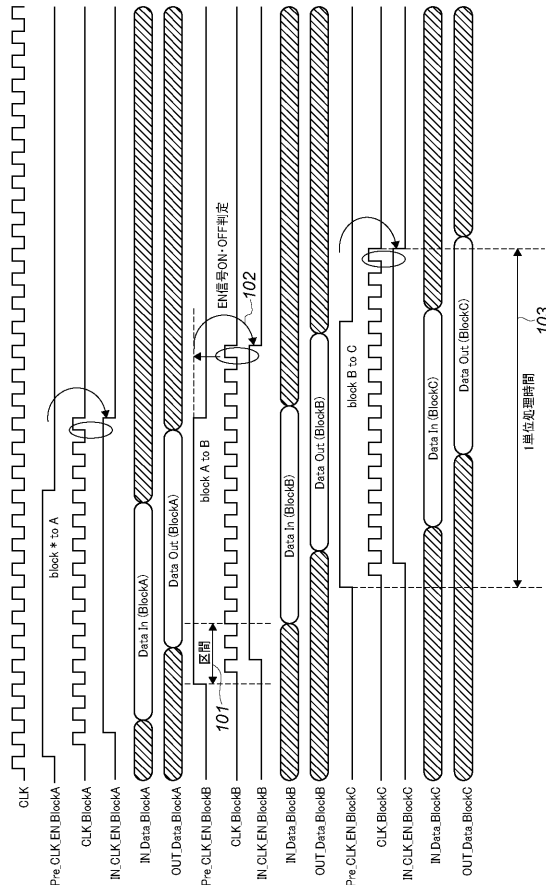
【図13】本実施の形態において、メインチップに2つのサブチップが接続されている構成例を示す図である。

【図14】本実施の形態に係るチップ間のクロックイネーブル信号を接続する他の例を説明する図である。

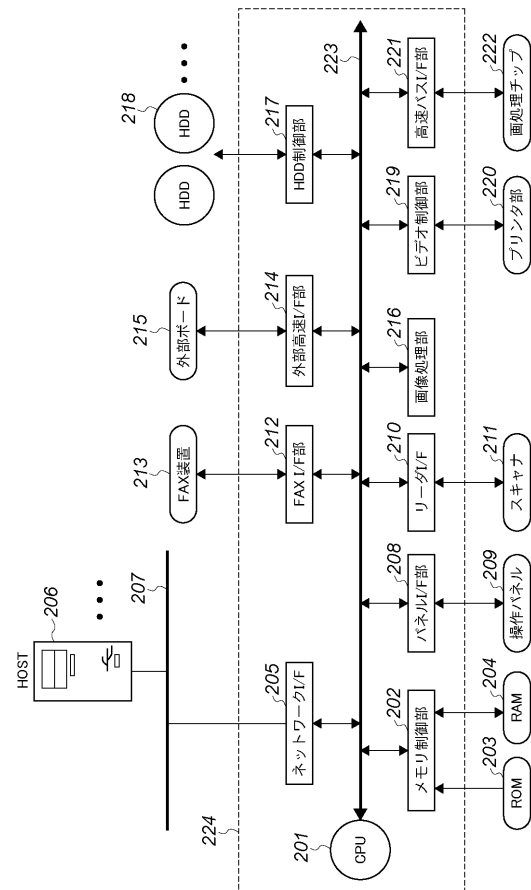
40

【図15】図13と構成は同じで、スタンバイ時の状態を表した図である。

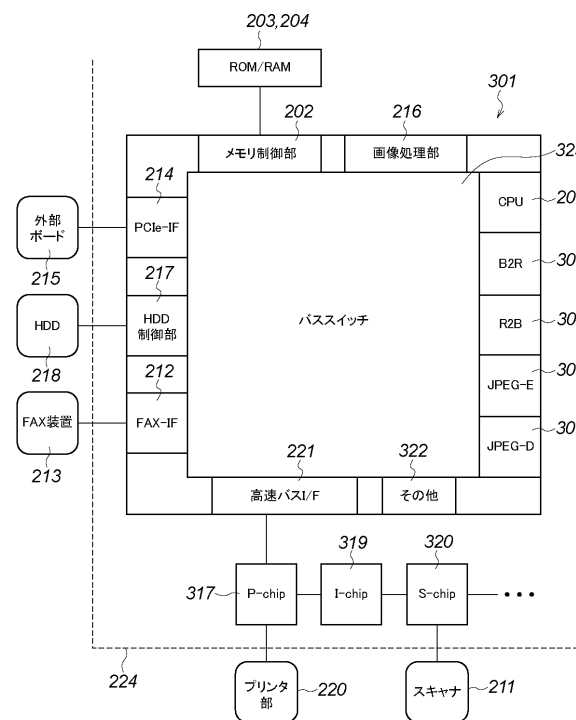
【図 1】



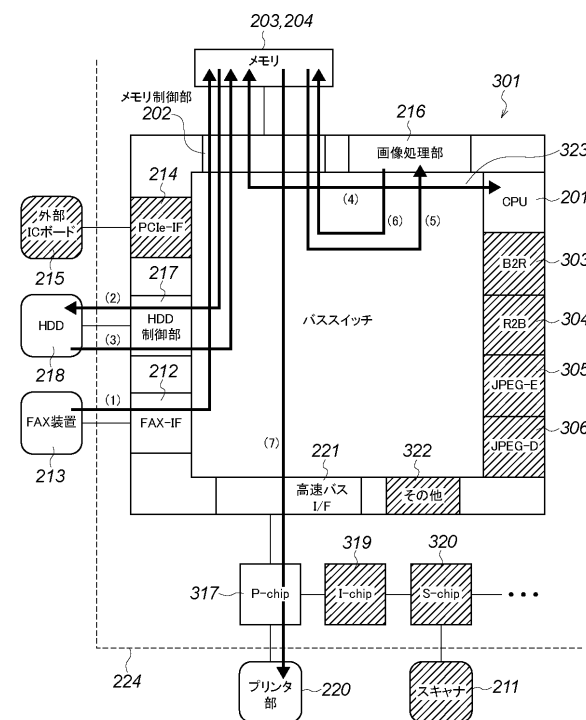
【図 2】



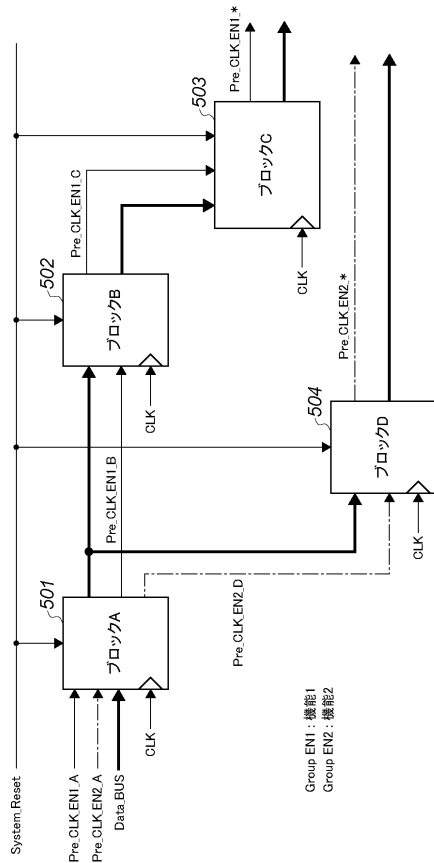
【図 3】



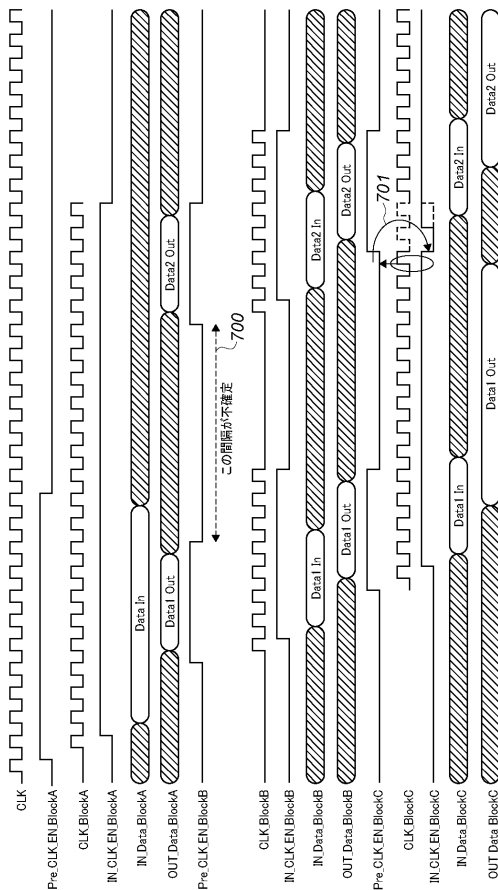
【図 4】



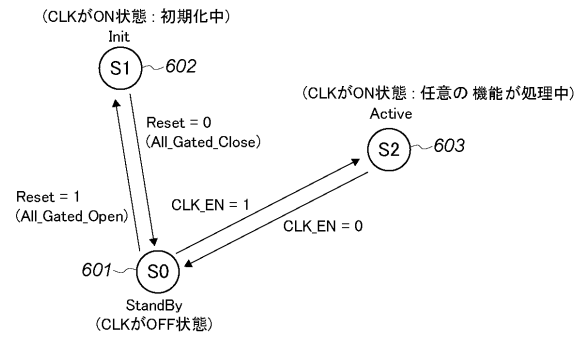
【図 5】



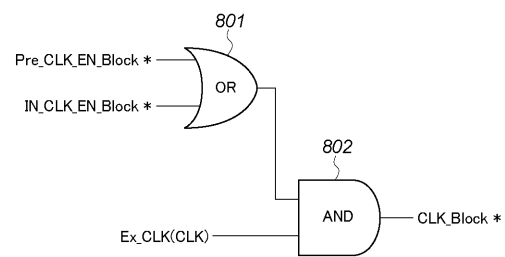
【図 7】



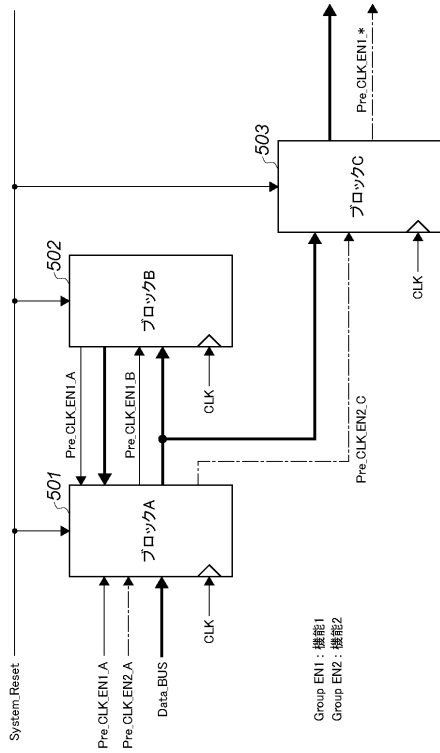
【図 6】



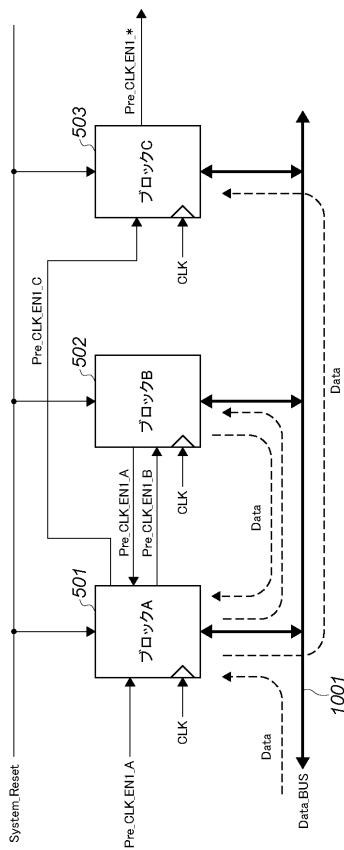
【図 8】



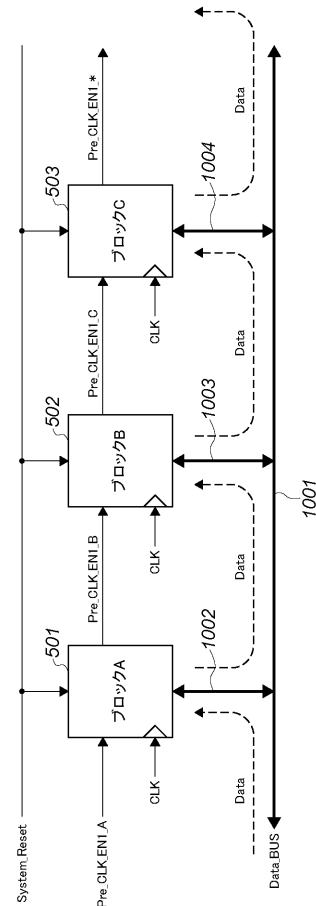
【図 9】



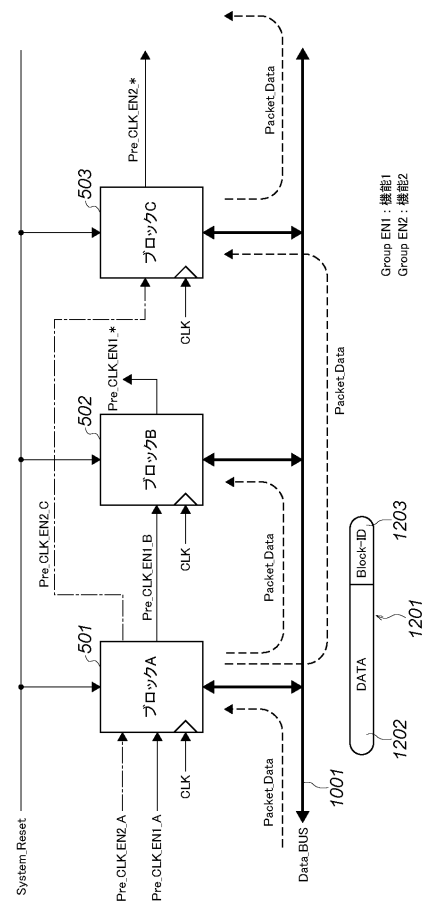
【図 11】



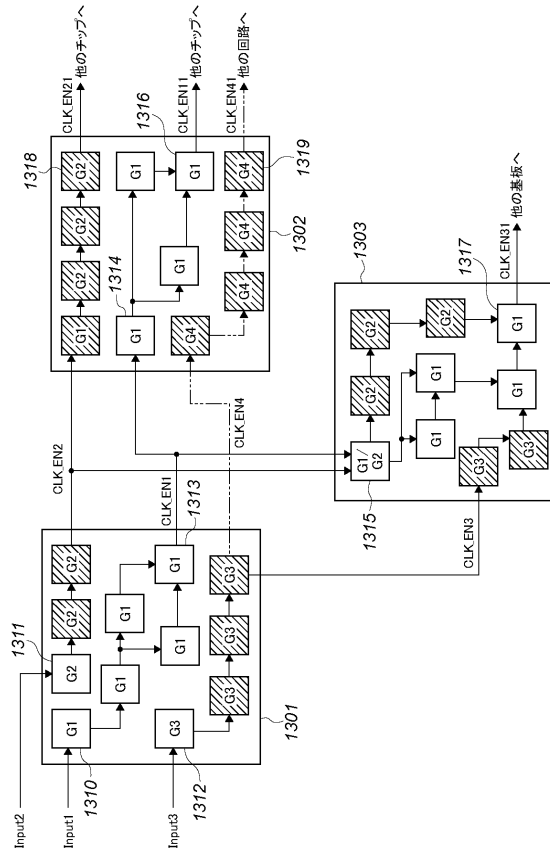
【図 10】



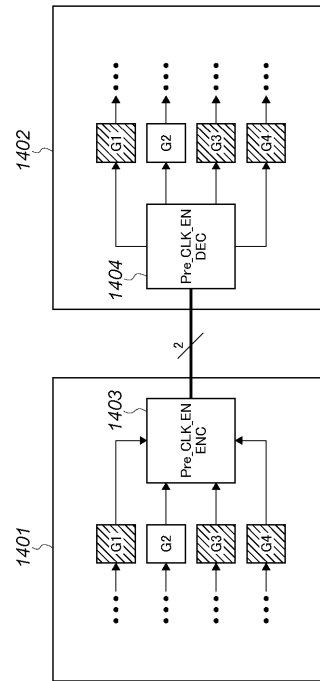
【図 12】



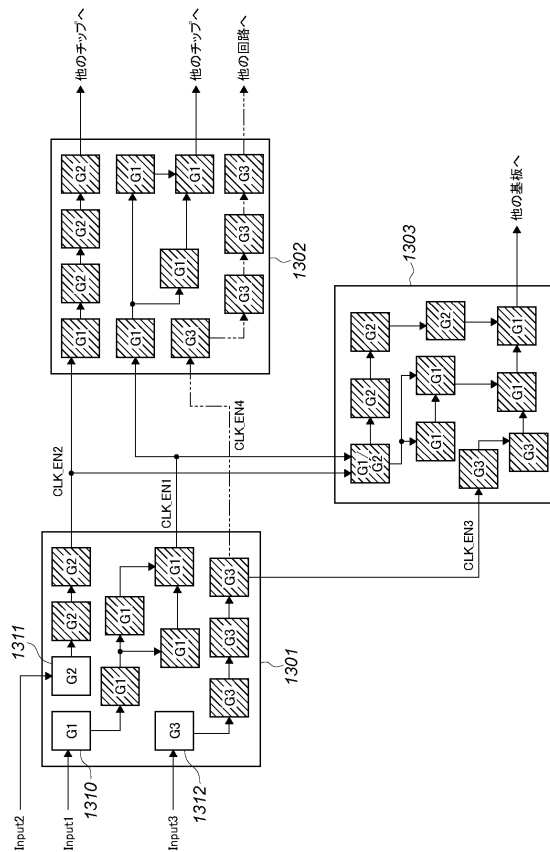
【図 13】



【図 14】



【図 15】



フロントページの続き

(72)発明者 松本 昭浩

東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社内

F ターム(参考) 2C061 AP01 AP03 AP04 AP07 HH11 HJ10 HT11 HT13

5B079 AA04 BA12 BC01

5C062 AA02 AA05 AB17 AB22 AB38 AB47 AB49 AC02 AC04