

(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2009-205334
(P2009-205334A)

(43) 公開日 平成21年9月10日(2009.9.10)

(51) Int.Cl. F I テーマコード(参考)
G06F 13/38 (2006.01) G06F 13/38 340A 5B077
 G06F 13/38 350

審査請求 未請求 請求項の数 12 O L (全 17 頁)

(21) 出願番号 特願2008-45730(P2008-45730)
 (22) 出願日 平成20年2月27日(2008.2.27)

(71) 出願人 000005108
 株式会社日立製作所
 東京都千代田区丸の内一丁目6番6号
 (74) 代理人 110000350
 ポレール特許業務法人
 (72) 発明者 村上 祥基
 神奈川県秦野市堀山下1番地 株式会社日
 立製作所エンタープライズサーバ事業部内
 (72) 発明者 處 雅尋
 神奈川県秦野市堀山下1番地 株式会社日
 立製作所エンタープライズサーバ事業部内
 Fターム(参考) 5B077 AA24 BA02 BA09 DD02 DD12

(54) 【発明の名称】 性能モニタ回路及び性能モニタ方法

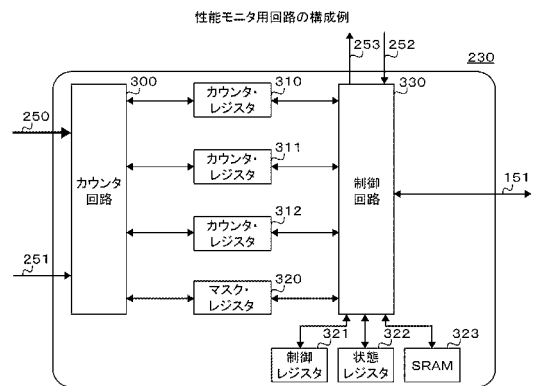
(57) 【要約】

【課題】 サンプリング間隔をクロック・サイクル単位で指定できる性能モニタ回路及びその方法を提供する。

【解決手段】 バッファを備え、受信したPCI Expressのトランザクション層の packets をバッファに格納し、バッファに格納された packets を送信するPCI Expressコンポーネントと接続する性能モニタ回路及びそのモニタ方法である。バッファから送信された packets のデータ量を計測する第1のカウンタ、バッファに packets が存在するクロック・サイクル数を計測する第2のカウンタ、バッファ内に送信する packets が存在するが送信できなかったクロック・サイクル数を計測する第3のカウンタ、及び第1～第3のカウンタによる計測を所定時間間隔で繰り返すように制御する制御回路を有する。

【選択図】 図3

図3



【特許請求の範囲】**【請求項 1】**

バッファを備え、受信した P C I E x p r e s s のトランザクション層のケットを前記バッファに格納し、前記バッファに格納された前記ケットを送信する P C I e x p r e s s コンポーネントと接続し、前記バッファから送信された前記ケットのデータ量を計測する第 1 のカウンタ、前記バッファに前記ケットが存在するクロック・サイクル数を計測する第 2 のカウンタ、前記バッファ内に送信するケットが存在するが送信できなかったクロック・サイクル数を計測する第 3 のカウンタ、及び前記第 1 ~ 第 3 のカウンタによる前記計測を所定時間間隔で繰り返すように制御する制御回路を有することを特徴とする性能モニタ回路。

10

【請求項 2】

前記制御回路が前記所定時間間隔で前記第 1 ~ 第 3 のカウンタによる計測値を格納する記憶回路をさらに有することを特徴とする請求項 1 記載の性能モニタ回路。

【請求項 3】

前記制御回路は、前記所定時間間隔で前記第 1 ~ 第 3 のカウンタによる計測値の前記記憶回路への格納に応答して、前記第 1 ~ 第 3 のカウンタを初期化することを特徴とする請求項 2 記載の性能モニタ回路。

【請求項 4】

前記所定時間及び前記所定時間間隔で繰り返す前記第 1 ~ 第 3 のカウンタによる計測期間を前記制御回路が参照する制御レジスタをさらに有することを特徴とする請求項 3 記載の性能モニタ回路。

20

【請求項 5】

前記第 1 のカウンタが前記ケットの中の所望のケットのデータ量を計測するために参照するマスクレジスタをさらに有することを特徴とする請求項 4 記載の性能モニタ回路。

【請求項 6】

前記バッファはアップストリーム用のバッファとダウンストリーム用のバッファであり、前記アップストリーム用のバッファと前記ダウンストリーム用のバッファとの各々に関して、前記第 1 ~ 第 3 のカウンタを有することを特徴とする請求項 1 記載の性能モニタ回路。

【請求項 7】

バッファを備え、受信した P C I E x p r e s s のトランザクション層のケットを前記バッファに格納し、前記バッファに格納された前記ケットを送信する P C I e x p r e s s コンポーネントと接続し、前記バッファから送信された前記ケットのデータ量を計測し、前記バッファに前記ケットが存在するクロック・サイクル数を計測し、前記バッファ内に送信するケットが存在するが送信できなかったクロック・サイクル数を計測し、前記第 1 ~ 第 3 のカウンタによる前記計測を所定時間間隔で繰り返すことを特徴とする性能モニタ方法。

30

【請求項 8】

前記所定時間間隔で計測した前記データ量、前記バッファに前記ケットが存在するクロック・サイクル数及び前記バッファ内に送信するケットが存在するが送信できなかったクロック・サイクル数を記憶回路に格納することを特徴とする請求項 7 記載の性能モニタ方法。

40

【請求項 9】

前記記憶回路への格納に応答して、前記データ量、前記バッファに前記ケットが存在するクロック・サイクル数及び前記バッファ内に送信するケットが存在するが送信できなかったクロック・サイクル数の計測を初期化することを特徴とする請求項 8 記載の性能モニタ方法。

【請求項 10】

前記所定時間及び前記所定時間間隔で繰り返す計測期間を制御レジスタに設定することを特徴とする請求項 9 記載の性能モニタ方法。

50

【請求項 1 1】

前記第 1 のカウンタが前記パケットの中の所望のパケットのデータ量を計測するためのマスクデータをマスクレジスタに設定することを特徴とする請求項 10 記載の性能モニタ方法。

【請求項 1 2】

前記バッファはアップストリーム用のバッファとダウンストリーム用のバッファであり、前記アップストリーム用のバッファと前記ダウンストリーム用のバッファとの各々に関して、前記データ量、前記バッファに前記パケットが存在するクロック・サイクル数及び前記バッファ内に送信するパケットが存在するが送信できなかったクロック・サイクル数を計測することを特徴とする請求項 7 記載の性能モニタ方法。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、PCI Express コンポーネントに係り、特に PCI (Peripheral component interconnect) Express コンポーネントのデータ転送性能の情報を計測し記録する回路及びその方法に関する。

【背景技術】

【0002】

最近では、非特許文献 1 で定義されるルート・コンプレックス、スイッチ、エンド・ポイントといった PCI Express コンポーネントにおいても性能モニタ機能を持つ必要が出てきた。理由は、現在 PCI の標準化団体である PCI-SIG (PCI Special Interest Group) において、IOV (I/O Virtualization) という名称で、PCI Express デバイスを仮想化するための標準仕様の策定が進んでおり、近い将来 IOV 対応の PCI Express デバイスが実現する可能性があるためである。PCI Express デバイスが仮想化されると、計算機システム設計者は、物理 PCI Express デバイス数を削減することによるコスト低減を目的に、VMM (Virtual Machine Monitor) 上で、1 つの物理 PCI Express デバイスを複数の仮想 PCI Express デバイスに分割し、ゲスト OS と呼ぶ VMM 上で動作する OS のそれぞれに対して割り当てる。

20

【0003】

現在、Intel 社製 CPU が備えている性能モニタ機能 (非特許文献 2 記載) のように、一般的に計算機を構成するコンポーネントが備える性能モニタ機能は、使用者が設定した時間内に発生したイベント数をカウントするのみである。そのため、例えば 10 秒間に発生したイベント数を、1 秒間隔で解析したい場合、性能モニタに対し 1 秒間のイベント数の計測を 10 回指示する必要がある。このような解析をするためには、例としてあげた Intel 社製 CPU には Intel 社が販売している Vtune アナライザーといった、専用の性能解析用ソフトウェアを必要とする。性能モニタ機能の使用者は、性能解析用ソフトウェア上でイベント数の計測を行う時間であるサンプリング期間と、イベント数の計測を行う間隔であるサンプリング間隔を指定する。そうすれば、性能解析ソフトウェアが、指定に従って性能モニタに指示を出し、イベント数の計測を複数回自動で行って

30

40

【0004】

【非特許文献 1】PCI Express Base Specification Revision 2.0, PCI-SIG, December 2006

【非特許文献 2】Intel Itanium Architecture Software Developer's Manual Volume 2: System Architecture Revision 2.2, Intel, January 2006

【発明の開示】

【発明が解決しようとする課題】

50

【0005】

上述のように、計算機システム設計者が物理PCI Expressデバイス数を削減した結果、各ゲストOSが行うデータ転送のQoS (Quality of Service) が保証されないのであれば、計算機システムとしての必要性能を満たせなくなるのでコスト削減は意味をなさない。従って、計算機システム設計者は、ゲストOSのデータ転送性能を解析し、QoSを保証できるように1つの物理PCI Expressデバイスを複数のゲストOSに割り当てる必要がある。そのためには、データ転送経路上に存在するPCI Expressコンポーネントにおいても、データ転送性能を解析するための性能モニタ機能を搭載する必要が出る。

【0006】

また、性能解析ソフトウェアは、ソフトウェアの実行時間の制約から、最短のサンプリング間隔がミリ秒と制限される。近年、CPUはマルチコア化が進み、ますます性能が上がっている。それに伴い、I/Oも転送性能を上げる必要が出てきて、2007年には、1リンクあたり5Gbpsの転送性能を持つPCI Express Gen2対応のPCI Expressコンポーネントが出荷され始めている。この流れは今後も続くと予想され、PCI Expressコンポーネントが単位時間あたりに処理するデータ量は益々増える。すると、これまでミリ秒単位のサンプリング間隔で十分な解析ができる数しか発生していなかったパケットの衝突や輻輳といったデータ転送に関わるイベントが、単位時間に扱うデータ量が増えることにより、ミリ秒単位のサンプリング間隔では十分な解析ができないほど多く発生する可能性が考えられる。よって、今後より細かいサンプリング間隔の指定が必要となると予想される。現在、PCI Expressコンポーネントは1クロック・サイクルが数ナノ秒で動いており、今後サンプリング間隔を最低限ミリ秒より細かいマイクロ秒単位か、動作クロック・サイクルと同じナノ秒単位で指定する必要が出ると考えられる。

【0007】

以上説明した通り、現在PCI Expressコンポーネントの開発をする上で、3つの課題が存在する。

(1) PCI Expressデバイスを仮想化して使用する環境では、ゲストOSのQoSを保証するために、データ転送性能の解析を可能とする性能モニタ機能をPCI Expressコンポーネントに搭載する必要がある。

(2) サンプリング期間とサンプリング間隔をそれぞれ指定して計算機を構成するコンポーネントの性能解析を行う場合、従来の性能モニタ機能では、専用の性能解析ソフトウェアが必要となる。このことは、性能モニタ機能の使用者に、性能解析ソフトウェアの購入コストもしくは開発コストの負担が生じることを意味する。

(3) 性能解析ソフトウェアを使用した性能解析では、現在サンプリング間隔をミリ秒単位より下の単位では指定できない。そのため、例えばPCI Expressコンポーネントのクロック・サイクル単位といった、よりサンプリング間隔の粒度を細かくした性能解析ができない。

【0008】

本発明の目的は、PCI Expressコンポーネントのデータ転送性能の解析を可能とし、性能解析の際に性能解析用ソフトウェアを必要とせず、サンプリング間隔をクロック・サイクル単位で指定できる性能モニタ回路及びその方法を提供することである。

【課題を解決するための手段】

【0009】

本発明の性能モニタ回路及びその方法は、次のような構成である。バッファを備え、受信したPCI Expressのトランザクション層のパケットを前記バッファに格納し、前記バッファに格納された前記パケットを送信するPCI Expressコンポーネントと接続する性能モニタ回路及びその性能モニタ回路による性能モニタ方法である。バッファから送信されたパケットのデータ量を計測する第1のカウンタ、バッファ内にパケットが存在するクロック・サイクル数を計測する第2のカウンタ、バッファ内に送信するパケ

ットが存在するが送信できなかったクロック・サイクル数を計測する第3のカウンタ、及び前記第1～第3のカウンタによる前記計測を所定時間間隔で繰り返すように制御する制御回路を有する。

【0010】

本発明の他の態様は、記憶回路を設け、その記憶回路に制御回路が第1～第3のカウンタによる計測値を格納する。

【0011】

本発明のさらに他の態様は、第1～第3のカウンタによる計測値の記憶回路への格納に応答して、第1～第3のカウンタを初期化する。

【0012】

本発明のさらに他の態様は、所定時間及び所定時間間隔で繰り返す第1～第3のカウンタによる計測期間を制御レジスタに設定する。

【0013】

本発明のさらに他の態様は、所望のパケットのデータ量を計測するためのマスクデータをマスクレジスタに設定する。

【0014】

本発明のさらに他の態様は、アップストリーム用のバッファとダウンストリーム用のバッファとの各々に関して、所定時間内の、バッファから送信されたパケットのデータ量、バッファにパケットが存在するクロック・サイクル数、及びバッファ内に送信するパケットが存在するが送信できなかったクロック・サイクル数を計測する。

【発明の効果】

【0015】

本発明の性能モニタ回路及びその方法によれば、サンプリング間隔をクロック・サイクル単位で指定できる。

【発明を実施するための最良の形態】

【0016】

以下、実施例を用いて本発明を実施するための最良の形態を詳細に説明する。

【実施例1】

【0017】

図1は、実施例1の計算機システムの一例である。本計算機システムは、CPU100とメモリ101、そしてPCI Expressコンポーネントであるルート・コンプレックス110とスイッチ111、エンド・ポイント112とエンド・ポイント113を備える。ルート・コンプレックス110とスイッチ111はPCI Expressリンク140で接続する。そして、スイッチ111とエンド・ポイント112及びエンド・ポイント113は、それぞれPCI Expressリンク141と142で接続する。

【0018】

サービス・プロセッサ120の機能は、計算機システムの各構成要素の電源管理（投入/切断）、構成要素情報の管理、障害発生時の計算機システム内トランザクション情報・処理状態履歴の採取・障害通報、システム内環境（温度・電源）監視である。本実施例では、管理用コンソール130とサービス・プロセッサ120は、インターフェース160を介して通信が可能である。サービス・プロセッサ120と管理用コンソール130間の通信は、例えばLANインターフェースやシリアル・インターフェースで実現すればよい。計算機システムの利用者は、管理用コンソール130を使用して、サービス・プロセッサ120を介し計算機システムの管理/操作を行う。

【0019】

サービス・プロセッサ120は、管理用コンソール180から各PCI Expressコンポーネントにある性能モニタ回路に対する設定値のリード・ライト及び計測結果の記録のリードができるインターフェース150、151、152及び153を備える。各PCI Expressコンポーネントとサービス・プロセッサ120間のインターフェースは、例えばハードウェアのモニタ用標準インターフェース仕様であるI2Cインタ-

10

20

30

40

50

フェースを用いればよい。

【0020】

本実施例の性能モニタ回路は、PCI Expressコンポーネントである、ルート・コンプレックス110、スイッチ111、及びエンド・ポイント112に備える。通常、PCI Expressコンポーネント（ルート・コンプレックス、スイッチ、エンド・ポイント等）は、PCI Expressコンポーネントが受信したTLP（Transaction Layer Packet）を一旦格納し、格納後パケットを他のPCI Expressコンポーネントか、計算機を構成する他のコンポーネント（CPU、メモリ等）に送信するためのトランザクション層に位置するバッファをデータ転送経路上に備える。このバッファに性能モニタ回路を備えることで、PCI Expressコンポーネントのデータ転送性能を解析するために必要な情報を得る。

10

【0021】

以下、スイッチ111に備える性能モニタ回路を使用して、性能モニタ回路の機能を説明する。性能モニタ回路の機能は、PCI Expressコンポーネントの種類により異なるものではない。

【0022】

図2に、本実施例におけるスイッチ111を示す。スイッチ111は、アップストリーム・ポート200とダウンストリーム・ポート201、ダウンストリーム・ポート202を備える。アップストリーム・ポート200はPCI Expressのデータリンク層及び物理ポート層の機能を備え、PCI Expressリンク140を介してルート・コンプレックス110からTLP（Transaction Layer Packet）を受信、もしくはルート・コンプレックス110に対してTLPを送信する。ダウンストリーム・ポート201とダウンストリーム・ポート202も同様にPCI Expressのデータリンク層及び物理ポート層の機能を備え、それぞれPCI Expressリンク141と142を介してエンド・ポイント112とエンド・ポイント113からTLPを受信、もしくはエンド・ポイント112とエンド・ポイント113に対してTLPを送信する。

20

【0023】

各ポートは、常にTLPの宛先であるPCI Expressコンポーネントのルート・コンプレックス110、エンド・ポイント112、及びエンド・ポイント113にあるTLPを受信するバッファの状態を各々監視している。宛先のPCI Expressコンポーネントのバッファに空きが有り、ポートが宛先のPCI Expressコンポーネントに対しTLPを送信することが可能であれば、その情報をバッファ（アップストリーム用のバッファ）210とバッファ（ダウンストリーム用のバッファ）211に教える。その情報を見たバッファ210とバッファ211は、（バッファ211はクロスバ220を介して）送信可能なポートへ、送信可能なPCI Expressコンポーネント宛でのTLPを送信する。TLPを受け取ったポートは、TLPを宛先のPCI Expressコンポーネントに対して送信する。また、各ポートは、バッファ210とバッファ211の状態を監視し、TLPの送信元でもあるPCI ExpressコンポーネントにTLPをスイッチ111が受信可能であるとの情報を送る。その情報を見たTLP送信元であるPCI ExpressコンポーネントはTLPをスイッチ111に対して送信し、各ポートは直接、もしくはクロスバ・スイッチ220を介してバッファ210またはバッファ211に対しTLPを送信する。

30

40

【0024】

バッファ210及びバッファ211はPCI Expressのトランザクション層に位置するバッファで、一旦バッファに対して送られてきたTLPを格納する。そして、TLPを見て宛先を確認し、宛先のPCI Expressコンポーネントが受信可能であれば、直接かクロスバ・スイッチ220を介して各ポートにTLPを送る。ただし、バッファ211が送信するTLPの宛先が性能モニタ回路230であるときは、TLP送信の動作が異なる。性能モニタ回路230宛でのTLPとは、使用者がCPU100上で動作するOSを介して性能モニタ回路230の設定値のリード/ライトや計測結果の記録のり

50

ードを行うときに、CPU 100が発行する。送信するTLPが性能モニタ回路230宛ての場合、性能バッファ211はクロスバ・スイッチ220とバス252を介して性能モニタ回路230に対しTLPを送信する。本実施例では、性能モニタ回路230は毎クロック・サイクルでTLPを処理できるものとし、性能モニタ回路230がTLPを受信可能かどうかは見ていない。もし性能モニタ回路が毎サイクルTLPを受信できないのであれば、バッファ211は、PCI ExpressコンポーネントにTLPを送信するのと同じく、性能モニタ回路230がTLPを受信可能であるか、TLPを送信する前に確認する必要がある。

【0025】

クロスバ・スイッチ220は、ダウンストリーム・ポート201またはダウンストリーム・ポート202、もしくは性能モニタ回路230からバス253を介して送られてきたTLPをバッファ210に対して送る。性能モニタ回路230が送信するTLPとは、性能モニタ回路230の設定値のリード・データや計測結果の記録のリード・データである。また、クロスバ・スイッチ220は、バッファ211から送られてきたTLPを、TLPの宛先を見て、宛先のPCI Expressコンポーネントと接続しているダウンストリーム・ポート201か202、もしくは性能モニタ回路230に対して送信する。

【0026】

性能モニタ回路230は、バス250を介してバッファ211と接続している。バス250を介してバッファ211は、性能モニタ回路230に対して毎クロック・サイクル、バッファ211に送信するTLPが存在することを示す情報と、バッファ211内に送信するTLPが存在するが送信できないことを示す情報を送る。情報は、例えば2つのレベル信号で実現され、1つのレベル信号の'1'は送信するTLPがバッファ211内に存在することを示し、もう1つのレベル信号の'1'はバッファ211内にTLPが存在するが送信できないことを示す。両レベル信号の'0'は、どちらの状態でもないことを示す。また、性能モニタ回路230は、バス251を介し、バッファ211が送信したTLPのコピーを受信する。そして、性能モニタ回路230は、サービスプロセッサ120とのインターフェース151と直接接続する。

【0027】

図3を用いて、本実施例の性能モニタ回路の内部構成及び処理内容について詳細に説明する。

【0028】

毎クロック・サイクル、カウンタ回路300は、バス250を介して送られてくるバッファ211に送信するTLPが存在することを示す情報を見て、バッファ211に送信するTLPが存在するのであれば、カウンタ・レジスタ310の値を+1する。同様に、毎クロック・サイクル、カウンタ回路300は、バス250を介して送られてくるバッファ211内に送信するTLPが存在するが送信できないことを示す情報を見て、バッファ211内に送信するTLPが存在するが送信できないのであれば、カウンタ・レジスタ310の値を+1する。また、毎クロック・サイクル、カウンタ回路300は、バス251を介してバッファ211が送信したTLPのコピーを受信する。カウンタ回路300はTLPを受信した時、最初にマスク・レジスタ320を参照する。

【0029】

マスク・レジスタ320は、スイッチ111の性能モニタ機能の使用者が性能解析のために任意に設定するレジスタであり、バッファ211が転送するTLP数の計測を所定のTLPに限定するとき使用する。

【0030】

マスク・レジスタ320の一例を図4に示す。マスク・レジスタ320は272ビットで構成される。400~402の3つのフィールドは各々1ビットで構成される。フィールド400が“enable”であることは、TLPヘッダを受信した時に、カウンタ回路300がカウンタ・レジスタ312にTLPヘッダの packetsize (Double Word、32bit)単位でプラスすることを示す。同様に、フィールド4

10

20

30

40

50

01が“enable”であることは、TLPデータを受信した時に、カウンタ回路300がカウンタ・レジスタ312にTLPデータのケット・サイズ分をDW単位でプラスすることを示し、フィールド402が“enable”であることは、TLPダイジェストを受信した時に、カウンタ回路300がカウンタ・レジスタ312にTLPダイジェストのデータ・サイズ分をDW単位でプラスすることを示す。

【0031】

フィールド403は13ビットで構成され、将来制御回路を機能拡張する際に使用するために予約しているフィールドであり、特に今回使用するものではない。

【0032】

フィールド404は、PCI Express仕様で定められるTLPヘッダの最大サイズである128ビットで構成され、性能モニタ回路の使用者は、データ量を計測する対象のTLPのTLPヘッダを本フィールドに設定する。フィールド405は、フィールド404と同じサイズである128ビットで構成され、フィールド404の各bitの有効・無効を指定するのに使用する。例えば、フィールド404で設定したTLPヘッダの一部のbitのみでTLPをマスクしたい時は、フィールド405で対応するbitのみを“1”に設定して“enable”とすればよい。カウンタ回路300はフィールド404及び405を見て、マスク条件に合うTLPがバッファ211から送信されたら、カウンタ・レジスタ312にTLPのケット・サイズ分をDW単位でプラスする。

10

【0033】

以上説明したマスク・レジスタ320のマスク機能は次のような二つの場合に使用することが効果的である。

20

【0034】

第一に、TLPは、TLPヘッダ、TLPデータ、TLPダイジェストの3つで構成され、通常、データ転送量はTLPデータの量のみを指す。他の2つはデータ転送のオーバーヘッドである。純粹にデータ転送量を測りたい場合に、マスク・レジスタ320のマスク機能を使用する。

【0035】

第二に、TLPヘッダには、発行元、発行先、トランザクション種別等を示す情報を格納しているので、TLPヘッダをマスクすることで、発行元単位、発行先、トランザクション種別に分類してデータ転送量を計測することができる。

30

【0036】

以上、マスク・レジスタの構成について説明したが、マスク機能及び構成については必要最小限のもののみ説明している。マスク機能及び構成は特に制限するものではなく、例えば性能モニタ回路を搭載するPCI Expressコンポーネントの仕様や、性能モニタ機能の使用者の要求に従い、マスク機能及び構成を拡張しても良い。

【0037】

制御回路330は、バッファ211が送信したTLP数と、バッファ211に送信するTLPが存在するクロック・サイクル数と、バッファ211内に送信するTLPが存在するが送信できないクロック・サイクル数を、制御レジスタ321の設定値に従い計測する。

40

【0038】

バッファ211が送信したTLP数の計測は、バッファが送信したデータ転送量の計測を意味する。毎クロック・サイクル、バッファが送信したTLPのケット・サイズを、TLPの最小のケット・サイズであるDW(Double Word、32bit)を単位としてプラスする。例えば、1クロック・サイクルで3DWのTLPを送信した時は、これまで計測したTLPのケット・サイズの合計に+3する。本計測により、サンプリング期間内の、バッファ211が存在するデータ転送経路のデータ転送量が分かる。

【0039】

バッファ211に送信可能なTLPが存在するクロック・サイクル数の計測は、サンプリング期間の、バッファ211が存在するデータ転送経路の使用率が分かる。

50

【 0 0 4 0 】

バッファ 2 1 1 内に送信するパケットが存在するが送信できなかったクロック・サイクル数の計測は、バッファ 2 1 1 より先のデータ転送経路で T L P の送信が滞っている時間の計測であり、データ転送経路の混雑度の計測を意味する。サンプリング期間のデータ転送量が期待値より低い場合は、原因として、(a) T L P 発行元が発行する T L P のデータ量が少ない、(b) バッファより先のデータ転送経路で T L P の送信が滞っている、の 2 つが考えられる。本計測により、バッファ 2 1 1 より先のデータ転送経路で T L P の送信が滞っている時間を計測でき、データ転送経路の混雑度が分かるので、原因 (a) (b) の区別が可能となる。

【 0 0 4 1 】

また、制御回路 3 3 0 は、バス 2 5 2 を介して性能モニタ回路 2 3 0 宛ての T L P を受信した時、T L P を見て、マスク・レジスタ 3 2 0 と制御レジスタ 3 2 1 のリード/ライトや、カウンタ・レジスタ 3 1 0 ~ 3 1 2、状態レジスタ 3 2 2 と S R A M (記憶回路) 3 2 3 のリードを行う。リード処理後、制御回路 3 3 0 は、バス 2 5 3 を介してリード・データを C P U 1 0 0 宛てに返す。性能モニタ機能の使用者が、管理用コンソール 1 3 0 を介して性能モニタ回路 2 3 0 の設定値のリード/ライトや計測結果の記録のリードを行った時は、インターフェース 1 5 1 から受信したリクエストに従い、マスク・レジスタ 3 2 0 と制御レジスタ 3 2 1 のリード/ライトや、カウンタ・レジスタ 3 1 0 ~ 3 1 2、状態レジスタ 3 2 2 と S R A M 3 2 3 のリードを行う。リード処理後、制御回路 3 3 0 は、インターフェース 1 5 1 を介してリード・データを管理用コンソール 1 3 0 宛てに返す。

【 0 0 4 2 】

制御レジスタ 3 2 1 の一例を図 5 に示す。制御レジスタ 3 3 0 は 1 1 2 ビットで構成され、5 0 0 ~ 5 0 4 の 5 つのフィールドに分かれる。フィールド 5 0 0 は 1 ビットで構成され、“ 1 ” の値で計測開始を示し、“ 0 ” の値で計測停止を示す。性能モニタ機能の使用者は、フィールド 5 0 0 を操作する事で、計測の開始と停止を指示できる。制御回路 3 3 0 は、フィールド 5 0 0 が “ 1 ” に設定されている間は計測を続け、“ 0 ” に設定された時は計測を止める。

【 0 0 4 3 】

フィールド 5 0 1 は 1 ビットで構成され、“ 1 ” の値でカウンタ・レジスタ 3 1 0 ~ 3 1 2 と S R A M 3 2 3、そして状態レジスタ 3 2 2 の初期化を示し、“ 0 ” の値では何もしない。カウンタ・レジスタ 3 1 0 ~ 3 1 2 と S R A M 3 2 3、そして状態レジスタ 3 2 2 の初期化が指示された時、制御回路 3 3 0 は、カウンタ・レジスタ 3 1 0 ~ 3 1 2 に格納している値を “ 0 ” にクリアする。そして、S R A M 3 2 3 の全エントリに格納している値を全て “ 0 ” の値にクリアし、状態レジスタ 3 2 2 に格納している値も全て “ 0 ” にクリアする。各初期化対象において初期化が完了するまで、制御回路 3 3 0 はフィールド 5 0 0 に “ 1 ” が設定されても計測を開始しない。初期化処理が完了した時、フィールド 5 0 1 は “ 0 ” の値に戻る。通常、性能モニタ機能の使用者は、計測開始を指示する前に、フィールド 5 0 1 に “ 1 ” を設定し、カウンタ・レジスタ 3 1 0 ~ 3 1 2 と S R A M 3 2 3、そして状態レジスタ 3 2 2 に格納している値を “ 0 ” にクリアする。

【 0 0 4 4 】

フィールド 5 0 2 は 1 4 ビットで構成され、将来制御回路を機能拡張する際に使用するために予約しているフィールドであり、特に今回使用するものではない。

【 0 0 4 5 】

フィールド 5 0 3 は 4 8 ビットで構成され、サンプリング間隔をクロック・サイクル単位で設定するフィールドである。性能モニタ機能の使用者は、当該フィールドに計測するサンプリング間隔を設定する。制御回路 3 3 0 は、設定したサンプリング間隔分のクロック・サイクルが経過した後、S R A M 3 2 3 に各カウンタ・レジスタ 3 1 0 ~ 3 1 2 の値を格納する。制御回路 3 3 0 は、初期化後、S R A M 3 2 3 のエントリ 0 番から各カウンタ・レジスタ 3 1 0 ~ 3 1 2 の値の格納を開始し、その後エントリ番号を + 1 しながら S R A M 3 2 3 にカウンタ・レジスタ 3 1 0 ~ 3 1 2 の値の格納を繰り返す。

10

20

30

40

50

【 0 0 4 6 】

フィールド 5 0 4 は 4 8 ビットで構成され、サンプリング期間（計測期間）をクロック・サイクル単位で格納するフィールドである。性能モニタ機能の使用者は、当該フィールドに計測するサンプリング期間を設定する。制御回路 3 3 0 は、設定したサンプリング期間分のクロック・サイクルが経過した後、S R A M 3 2 3 にカウンタ・レジスタ 3 1 0 ~ 3 1 2 の値を最後に書き込んだエントリのフラグ・フィールドに“ 1 ”の値をライトする。

【 0 0 4 7 】

状態レジスタ 3 2 2 は、スイッチ 1 1 1 の性能モニタ回路のステータスを示すレジスタである。状態レジスタ 3 2 2 の一例を図 6 に示す。状態レジスタ 3 2 2 は 1 1 2 ビットで構成され、6 0 0 ~ 6 0 6 の 7 つのフィールドに分かれる。

10

【 0 0 4 8 】

フィールド 6 0 0 は 2 ビットで構成され、制御回路 3 3 0 のステータスを示す。“ 0 0 ”で制御回路 3 3 0 が計測していないことを示し、“ 0 1 ”で制御回路 3 3 0 が計測中であることを示す。そして、“ 1 0 ”は計測開始後計測が完了する前に制御レジスタ 3 2 1 のフィールド 5 0 0 が“ 0 ”に設定されて計測が停止したことを示し、“ 1 1 ”は計測が終了したことを示す。制御回路 3 3 0 は、フィールド 5 0 0 に“ 1 ”が設定され計測を開始したら、フィールド 6 0 0 を“ 0 1 ”に設定する。次に、制御回路 3 3 0 は、計測開始後フィールド 5 0 0 に“ 0 ”が設定されたら、フィールド 6 0 0 に“ 1 0 ”を設定する。そして、制御回路 3 3 0 は、フィールド 6 0 6 の値と制御レジスタ 3 2 1 のフィールド 5 0 4 の値が同じになった時、“ 1 1 ”を設定する。

20

【 0 0 4 9 】

フィールド 6 0 1 ~ 6 0 3 はそれぞれ 1 ビットで構成され、カウンタ・レジスタでオーバーフローが発生した時に、制御回路 3 3 0 が“ 1 ”に設定し、オーバーフローを報告する。フィールド 6 0 1 はカウンタ・レジスタ 3 1 0、フィールド 6 0 2 はカウンタ・レジスタ 3 1 1、フィールド 6 0 3 はカウンタ・レジスタ 3 1 2 にそれぞれ対応する。

【 0 0 5 0 】

フィールド 6 0 4 は 1 1 ビットで構成され、将来制御回路を機能拡張する際に使用するために予約しているフィールドであり、特に今回使用するものではない。

【 0 0 5 1 】

フィールド 6 0 5 のインターバル・タイマは 4 8 ビットで構成され、クロック・サイクル単位で計測時間を示す。制御回路 3 3 0 は、計測中であるならば、インターバル・タイマの値を 1 クロック・サイクル毎に + 1 する。ただし、制御レジスタ 3 2 1 のサンプリング間隔の値とインターバル・タイマの値が等しくなった時、制御回路 3 3 0 は、インターバル・タイマを“ 0 ”にクリアし、計測が続くのであれば再び計測時間を記録する。

30

【 0 0 5 2 】

フィールド 6 0 6 のイベント・タイマは 4 8 ビットで構成され、クロック・サイクル単位で計測時間を示す。制御回路 3 3 0 は、計測中であるならば、イベント・タイマの値を 1 クロック・サイクル毎に + 1 する。ただし、インターバル・タイマとは異なり、初期化以外でイベント・タイマはクリアされない。制御レジスタ 3 2 2 のサンプリング期間の値とイベント・タイマの値が等しくなった時、性能モニタ回路 2 3 0 の計測は終了したことを示す。

40

【 0 0 5 3 】

S R A M 3 2 3 の 1 例を図 7 に示す。本実施例では、S R A M のエントリ数を 1 0 2 4 とする。S R A M 3 2 3 は、7 0 1 ~ 7 0 4 の 4 つのフィールドで構成される。フィールド 7 0 0 はエントリ番号を示しているが、実際の S R A M には当該フィールドはなく、説明を容易にするために仮に定めたフィールドである。フィールド 7 0 1 は各エントリに 1 ビットで構成され、“ 1 ”は計測が終了した時に最後に値が格納されたエントリであることを示す。フィールド 7 0 2 ~ 7 0 4 は、各カウンタ・レジスタの値を格納するフィールドであり、各エントリにカウンタ・レジスタと同じビット数で構成される。制御回路 3 3

50

0 は、フィールド 7 0 2 にはカウンタ・レジスタ 3 1 0 の値を、フィールド 7 0 3 にはカウンタ・レジスタ 3 1 1 の値を、フィールド 7 0 4 にはカウンタ・レジスタ 3 1 2 の値をそれぞれ格納する。

【 0 0 5 4 】

以上、制御レジスタ 3 2 1、状態レジスタ 3 2 2、SRAM 3 2 3 の構成を説明したが、用いた例は必要最低限の構成であり、構成及び各フィールドのビット数は、特に規定するものではない。性能モニタ回路を搭載する PCI Express コンポーネントの仕様に応じて、必要なフィールド及びビット数の拡張を行っても構わない。

【 0 0 5 5 】

図 8 に、制御回路 3 3 0 が 1 クロック・サイクル毎に実行する処理フローチャートを示す。図 8 に示すフローチャートでは、性能モニタ機能の使用者が、CPU 1 0 0 もしくは管理用コンソール 1 3 0 を介して、性能モニタ回路 2 3 0 の設定値のリード/ライトや計測結果の記録のリードを行う際の処理については示していないが、容易に推敲されるだろう。図 8 に示す処理と並行して、制御回路 3 3 0 は、使用者の操作に伴う処理を実行できる。また、制御レジスタ 3 2 2 のフィールド 5 0 1 を “ 1 ” に設定することで、性能モニタ機能の使用者が性能モニタ回路 2 3 0 の初期化を指示した場合の処理も図 8 には示していないが、容易に推敲されるだろう。性能モニタ回路 2 3 0 の初期化指示を受けた制御回路 3 3 0 は、初期化処理が完了するまで、図 8 に示す処理は行わない。

10

【 0 0 5 6 】

以下、図 8 に示す処理フローチャートの各ステップについて説明する。ステップ 8 0 0 で、制御回路 3 3 0 は、制御レジスタ 3 2 1 のフィールド 5 0 0 のスタート/ストップが “ 1 ” の “ 計測開始 ” の設定であるか判定する。制御回路 3 3 0 は、“ 計測開始 ” の設定であった場合はステップ 8 0 1 の処理を行い、“ 0 ” で “ 計測停止 ” の設定であった場合はステップ 8 0 2 の処理を行う。ステップ 8 0 1 で、制御回路 3 3 0 は、状態レジスタ 3 2 2 のフィールド 6 0 0 の制御回路ステータスを “ 0 1 ” の “ 計測中 ” に設定する。

20

【 0 0 5 7 】

ステップ 8 0 2 で、制御回路 3 3 0 は、状態レジスタ 3 2 2 フィールド 6 0 0 の制御回路ステータスが “ 0 1 ” の “ 計測中 ” であるか判定する。“ 計測中 ” の設定であった場合、制御回路 3 3 0 は、ステップ 8 0 3 の処理を行い、その他の設定であった場合は当該クロック・サイクルの処理を終了する。ステップ 8 0 3 で、制御回路 3 3 0 は、状態レジスタ 3 2 2 のフィールド 6 0 0 の制御回路ステータスを “ 1 0 ” の “ ポーズ ” に設定し、当該クロック・サイクルの処理を終了する。

30

【 0 0 5 8 】

ステップ 8 0 4 で、制御回路 3 3 0 は、状態レジスタ 3 2 2 のフィールド 6 0 5 のインターバル・タイマと制御レジスタ 3 2 1 のフィールド 5 0 3 のサンプリング間隔の値が等しいかを判定する。制御回路 3 3 0 は、等しかった場合はステップ 8 0 5 の処理を行い、等しくなかった場合はステップ 8 0 6 の処理を行う。

【 0 0 5 9 】

ステップ 8 0 5 で、制御回路 3 3 0 は、カウンタ・レジスタ 3 1 0 ~ 3 1 2 の値を SRAM 3 2 3 に格納する。また、制御回路 3 3 0 は、SRAM 3 2 3 に値を格納するために各カウンタ・レジスタから値をリードしたと同時に、各カウンタ・レジスタの値を “ 0 ” にクリアする。そして、制御回路 3 3 0 は、状態レジスタ 3 2 2 のフィールド 6 0 5 のインターバル・タイマの値を “ 0 ” にクリアする。ステップ 8 0 6 で、制御回路 3 3 0 は、状態レジスタ 3 2 2 のフィールド 6 0 5 のインターバル・タイマの値を + 1 する。

40

【 0 0 6 0 】

ステップ 8 0 7 で、制御回路 3 3 0 は、状態レジスタ 3 2 2 のフィールド 6 0 6 のイベント・タイマと制御レジスタ 3 2 1 のフィールド 5 0 4 のサンプリング期間の値が等しいかを判定する。制御回路 3 3 0 は、等しい場合はステップ 8 0 8 の処理を行い、等しくない場合はステップ 8 0 9 の処理を行う。

【 0 0 6 1 】

50

ステップ 808 で、制御回路 330 は、SRAM 323 の最後に格納したエントリのフィールド 701 のフラグに “ 1 ” をセットする。また、状態レジスタ 322 フィールド 600 の制御回路ステータスを “ 11 ” の “ 計測終了 ” に設定し、当該クロック・サイクルの処理を終了する。ステップ 809 で、制御回路 330 は、状態レジスタ 322 フィールド 606 のイベント・タイマの値を + 1 して、当該クロック・サイクルの処理を終了する。

【 0062 】

本実施例によれば、サンプリング間隔をクロック・サイクル単位で指定できる性能モニタ回路を実現することができる。

【 実施例 2 】

【 0063 】

図 9 に、実施例 2 のスイッチ 111 を示す。本実施例のスイッチ 111 は、実施例 1 にバッファ 210 の性能モニタ機能を追加したものである。バッファ 210 が送信した TLP の数と、バッファ 210 に送信する TLP が存在するクロック・サイクル数と、バッファ 210 内に送信する TLP が存在するが送信できなかったクロック・サイクル数を計測するために、バッファ 211 と同様に、バス 250 と同じ機能を持つバス 950 と、バス 251 と同じ機能を持つバス 951 を備える。性能モニタ回路 930 は、バッファ 210 の上記 3 つの計測を行うために、性能モニタ回路 230 の機能を拡張したものである。

【 0064 】

図 10 に性能モニタ回路 930 の一例を示す。以下、性能モニタ回路 930 について、性能モニタ回路 230 に追加した構成の説明を行う。追加構成以外の性能モニタ回路 930 の構成は、性能モニタ回路 230 の構成と同じである。

【 0065 】

毎クロック・サイクル、カウンタ回路 1000 は、バス 950 を介して送られてくるバッファ 210 に送信する TLP が存在することを示す情報を見て、バッファ 210 に送信する TLP が存在するのであれば、カウンタ・レジスタ 1010 の値を + 1 する。同様に、毎クロック・サイクル、カウンタ回路 1000 は、バス 950 を介して送られてくるバッファ 210 内に送信する TLP が存在するが送信できないことを示す情報を見て、バッファ 210 内に送信する TLP が存在するが送信できないのであれば、カウンタ・レジスタ 1011 の値を + 1 する。また、毎クロック・サイクル、カウンタ回路 1000 は、バス 951 を介してバッファ 210 が送信した TLP のコピーを受信する。カウンタ回路 1000 は TLP を受信した時、最初にマスク・レジスタ 320 を参照し、マスク・レジスタ 320 の設定内容が示す計測対象の TLP であれば、パケット・サイズ分を DW 単位でカウンタ・レジスタ 1012 の値にプラスする。

【 0066 】

制御回路 1030 の構成は、制御回路 330 の構成に、カウンタ・レジスタ 310 ~ 312 に対する処理と同様の処理をカウンタ・レジスタ 1010 ~ 1012 に対して行う構成を追加したものである。また、制御回路 1030 の動作は、SRAM 1023 に対する動作も追加している。

【 0067 】

SRAM 1023 の 1 例を、図 11 に示す。SRAM 323 に対して SRAM 1023 は、フィールド 1102 ~ 1104 の 3 つのフィールドを追加している。フィールド 1102 ~ 1104 は、フィールド 702 ~ 704 と同様に、各エントリにカウンタ・レジスタと同じビット数で構成される。制御回路 1030 は、フィールド 1102 はカウンタ・レジスタ 1010 の値を、フィールド 1103 にはカウンタ・レジスタ 1011 の値を、フィールド 1104 にはカウンタ・レジスタ 1012 の値をそれぞれ格納する。

【 0068 】

制御回路 1030 は、状態レジスタ 322 のフィールド 605 のインターバル・タイマと制御レジスタ 321 フィールド 503 のサンプリング間隔の値が等しいと判定したクロック・サイクルで、カウンタ・レジスタ 310 ~ 312 の値に加え、カウンタ・レジスタ

10

20

30

40

50

1010～1012の値をSRAM1021に格納する。また、制御回路1030は、SRAM1023に値を格納するために各カウンタ・レジスタから値をリードしたと同時に、各カウンタ・レジスタの値を“0”にクリアする。

【0069】

また、状態レジスタ322もフィールドを追加して、状態レジスタ1022とする。図12に状態レジスタ1022の一例を示す。追加したフィールド1201～1203は、それぞれカウンタ・レジスタ1010～1012がオーバーフローした時に、制御回路1030が“1”に設定して、オーバーフローがあったことを報告するためのフィールドである。

【0070】

実施例2のスイッチ111は、データ転送性能を計測するバッファの数が2の例であるが、さらにデータ転送性能を計測するバッファの数が増えても、同様の拡張をバッファ単位に施すことで、性能モニタ回路はバッファが1つである時と同様に複数バッファのデータ転送性能の計測ができる。従って、PCI Expressコンポーネント内で複数バッファが存在する場合でも、性能モニタ回路は、同じサンプリング間隔とサンプリング期間で計測と計測結果の記録を行うことができる。これにより、各バッファの計測結果の時間的な相関を見ることができる。

【0071】

以上の実施形態によれば、性能モニタ回路は、解析の対象となるデータ転送経路の単位時間内のデータ転送量、使用率、混雑度を計測することができ、さらに、発行元や発行先、トランザクション種別等に分けてデータ転送量を計測できる。これにより、計算機システム設計者は、データ転送経路のデータ転送性能を解析するために必要な情報を得ることができる。

【0072】

1つの物理PCI Expressデバイスを、複数の仮想PCI Expressデバイスに分割して複数のゲストOSに割り当てる計算機システム設計者が、どのように性能モニタ回路を使用するのか一例を挙げて説明する。まず、計算機システム設計者は、PCI Expressデバイスに繋がるデータ転送経路上の、各ゲストOSのデータ転送性能を各々計測する。その後、計算機システム設計者は、各ゲストOSのQoSを保証するために、各ゲストOSのデータ転送性能の合計が、データ転送経路が持つデータ転送性能の最大値を越えないようにゲストOSを物理デバイスに割り当てればよい。また、実際に運用を開始した後、各ゲストOSのデータ転送性能が設計通りにならない場合がある。原因としては、先の計測時と異なった挙動をゲストOSがする、もしくはデータ転送経路上にある回路の不良等が考えられる。このような状況において計算機システム設計者は、データ転送経路の使用率及び混雑度等を見て、性能低下を引き起こしている原因を突き止める。その際、データ転送量を計測するTLPをマスクすることは、解析の精度を向上させることに寄与する。

【0073】

先に述べた通り、性能モニタ回路は、PCI Expressコンポーネント内に実装し、毎クロック・サイクル発生するイベント数をカウントするために、PCI Expressコンポーネントと同じクロック・サイクルで動作する。これに伴い、サイクル間隔及びサイクル期間をクロック・サイクル単位で指定可能とし、最低でもサイクル間隔及びサイクル期間を格納するレジスタは、ミリ秒を満たすクロック・サイクル数を格納できる大きさとする。同様に、イベント数をカウントするのに使用するカウンタ・レジスタも、最低でもミリ秒間で発生するイベント数をカウントするのに十分な大きさとする。以上により、ナノ秒～ミリ秒単位でのサンプリング間隔及びサンプリング期間の指定を実現する。

【0074】

ただし、サイクル間隔及びサイクル期間を格納するレジスタやイベント数をカウントするのに使用するカウンタ・レジスタは、特に大きさを規定するものではなく、サイクル間

10

20

30

40

50

隔及びサイクル期間を1秒間指定できるようにする等、必要に応じて拡張もしくは削減して構わない。

【0075】

次に、性能モニタ回路は、使用者が指定するサンプリング間隔に従って、サンプリング期間内に上記3つの計測と計測結果の記録を複数回に分割して実行することができる、例えば、サンプリング期間を10秒としてサンプリング間隔も同じく10秒とした時は記録を1回だけ行い、サンプリング間隔を1秒とした時は1秒間の計測と計測結果の記録を10回繰り返す。

【0076】

さらに、性能モニタ回路は、PCI Expressコンポーネント内で複数バッファが存在した場合に、同じサンプリング間隔とサンプリング期間で上記3つの計測と計測結果の記録を行うことができる。これにより、各バッファの計測結果の時間的な相関を見ることができる。

10

【0077】

以上に説明した通り、本発明を実施するための形態によれば、サンプリング期間とサンプリング間隔をそれぞれ指定してPCI Expressコンポーネントの性能解析を行う場合にも、性能解析用ソフトウェアを用いずにPCI Expressコンポーネントのデータ転送性能の解析が可能となる。またPCI Expressコンポーネント内に性能モニタ回路を実装するので、PCI Expressコンポーネントと同じクロック・サイクルで動作可能となり、性能解析の際にサンプリング間隔をミリ秒単位より細かいクロック・サイクル単位で指定できる。結果、従来技術と比べてより細かい粒度の性能解析ができる。

20

【図面の簡単な説明】

【0078】

【図1】実施例1の計算機システムの一例である。

【図2】実施例1におけるスイッチを示す。

【図3】実施例1の性能モニタ回路の構成例を示す。

【図4】実施例1のマスク・レジスタの一例を示す。

【図5】実施例1の制御レジスタの一例を示す。

【図6】実施例1の状態レジスタ322の一例を示す。

30

【図7】実施例1のSRAM323の一例を示す。

【図8】実施例1の制御回路の処理フローチャートの一例を示す。

【図9】実施例2におけるスイッチを示す。

【図10】実施例2の性能モニタ回路の構成例を示す。

【図11】実施例2のSRAMの一例を示す。

【図12】実施例2の状態レジスタの一例を示す。

【符号の説明】

【0079】

100：CPU、101：メモリ、110：ルート・コンプレックス、111：スイッチ、112～113：エンド・ポイント、120：サービス・プロセッサ、130：管理用コンソール、140～142：PCI Expressリンク、150～153：サービス・プロセッサとPCI Expressコンポーネント間のインターフェース、160：管理用コンソールとサービス・プロセッサ間のインターフェース、200：アップストリーム・ポート、201～202：ダウンストリーム・ポート、210～211：バッファ、220：クロスバ・スイッチ、230、930：性能モニタ回路、250、950：データ転送性能計測のための情報を性能モニタ回路に渡すバス、251、951：バッファが転送するTLPのコピーを性能モニタ回路に渡すバス、252～253：性能モニタ回路の設定及び計測結果を読み出すためのバス、300：カウンタ回路、310～312、1010～1012：カウンタ・レジスタ、320：マスク・レジスタ、321：制御レジスタ、322、1022：状態レジスタ、323、1023：SRAM、330、1

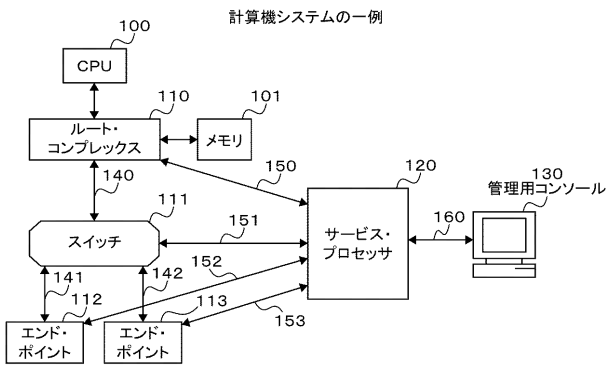
40

50

030 : 性能モニタ回路。

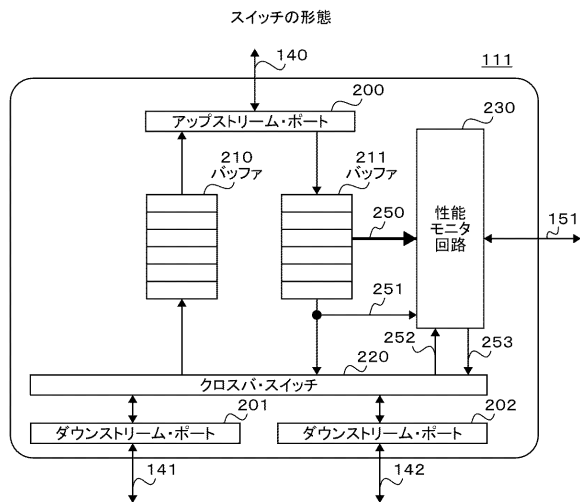
【 図 1 】

図 1



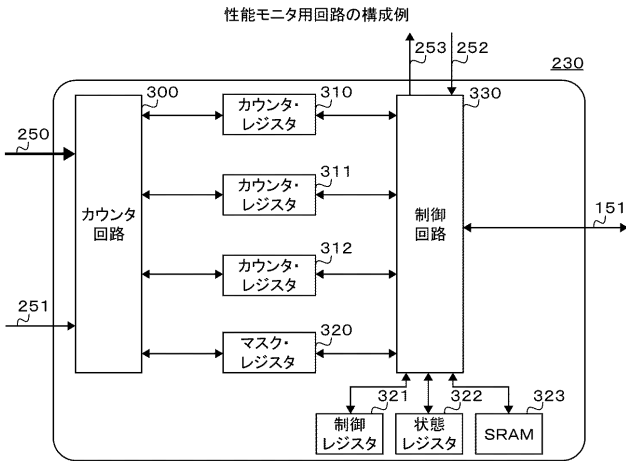
【 図 2 】

図 2



【 図 3 】

図 3



【 図 4 】

図 4

マスクレジスタの一例

ビット位置	内容
0	TLPヘッダ採取 0: disable, 1: enable
1	TLPデータ採取 0: disable, 1: enable
2	TLPダイジェスト採取 0: disable, 1: enable
15:3	予約フィールド
143:16	計測対象のTLPヘッダ・マスク
271:144	TLPヘッダ・マスクのenable/disable 0: disable, 1: enable

【 図 7 】

図 7

SRAMの一例

エントリ番号	フラグ	カウンタレジスタ310	カウンタレジスタ311	カウンタレジスタ312
0	0	0000_0100	0000_0000_0100	0000_0000_0100
1	0	0000_0200	0000_0000_0000	0000_0000_0000
2	0	0000_0100	0000_0000_0100	0000_0000_0100
...
1023	1	0000_0100	0000_0000_0100	0000_0000_0100

【 図 5 】

図 5

制御レジスタの一例

ビット位置	内容
0	スタート/ストップ 0: 計測停止, 1: 計測開始
1	初期化 0: 初期化せず, 1: 初期化実行
15:2	予約フィールド
63:16	サンプリング間隔(クロック・サイクル単位)
111:64	サンプリング期間(クロック・サイクル単位)

【 図 6 】

図 6

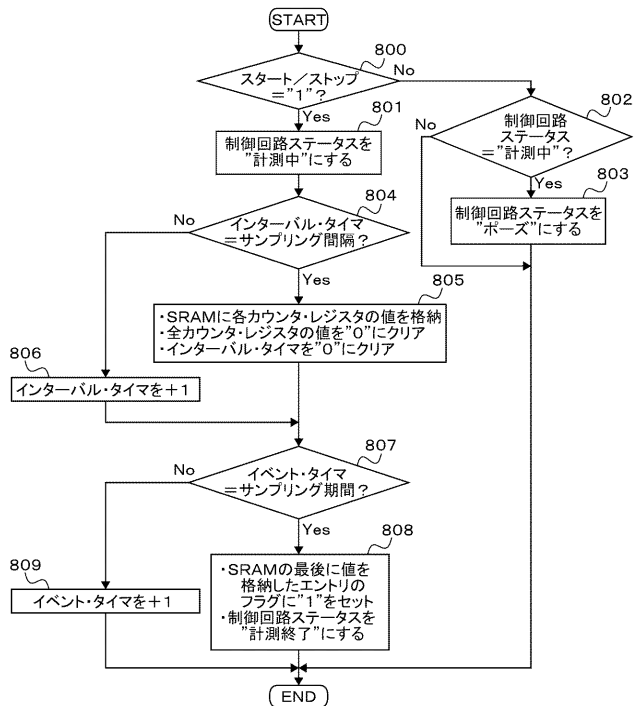
状態レジスタの一例

ビット位置	内容
1:0	制御回路ステータス 00: 計測停止 01: 計測中 10: ポーズ 11: 計測終了
2	カウンタレジスタ310がオーバーフロー 0: オーバーフローせず, 1: オーバーフロー発生
3	カウンタレジスタ311がオーバーフロー 0: オーバーフローせず, 1: オーバーフロー発生
4	カウンタレジスタ312がオーバーフロー 0: オーバーフローせず, 1: オーバーフロー発生
15:5	予約フィールド
63:16	インターバル・タイマ
111:64	イベント・タイマ

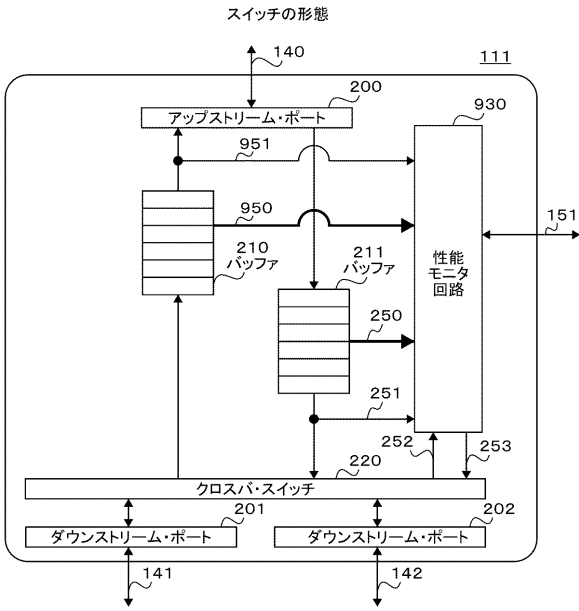
【 図 8 】

図 8

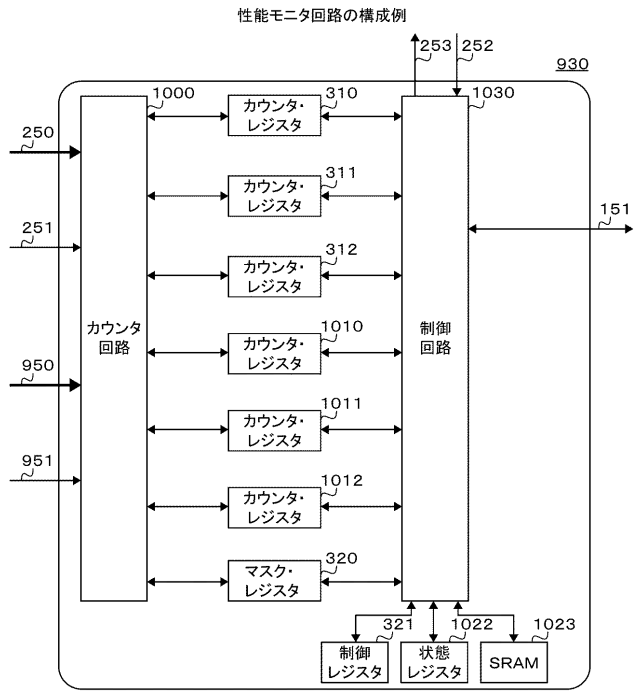
制御回路の処理フローチャートの一例



【 図 9 】



【 図 10 】



【 図 1 1 】

図 1 1

SRAMの一例

700	エンドリ 番号	701	カウンタ・ レジスタ310	702	カウンタ・ レジスタ311	703	カウンタ・ レジスタ312	704	カウンタ・ レジスタ1010	705	カウンタ・ レジスタ1011	706	カウンタ・ レジスタ1012
0	0	0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100
1	0	0000.0200	0000.0000.0000	0000.0000.0000	0000.0000.0000	0000.0000.0000	0000.0000.0000	0000.0000.0000	0000.0000.0000	0000.0000.0000	0000.0000.0000	0000.0000.0000	
2	0	0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	
...	
1023	1	0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	0000.0000.0100	

【 図 1 2 】

図 1 2

状態レジスタの一例

ビット位置	内容	
1:0	制御回路ステータス 00:停止 01:計測中 10:ポーズ 11:計測終了	600
2	カウンタ・レジスタ310がオーバーフロー 0:オーバーフローせず, 1:オーバーフロー発生	601
3	カウンタ・レジスタ311がオーバーフロー 0:オーバーフローせず, 1:オーバーフロー発生	602
4	カウンタ・レジスタ312がオーバーフロー 0:オーバーフローせず, 1:オーバーフロー発生	603
5	カウンタ・レジスタ1010がオーバーフロー 0:オーバーフローせず, 1:オーバーフロー発生	1201
6	カウンタ・レジスタ1011がオーバーフロー 0:オーバーフローせず, 1:オーバーフロー発生	1202
7	カウンタ・レジスタ1012がオーバーフロー 0:オーバーフローせず, 1:オーバーフロー発生	1203
15:8	予約フィールド	604
63:16	インターバル・タイマ	605
95:64	イベント・タイマ	606