

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4046825号
(P4046825)

(45) 発行日 平成20年2月13日(2008.2.13)

(24) 登録日 平成19年11月30日(2007.11.30)

(51) Int.Cl.

F I

G O 6 F 13/28 (2006.01)

G O 6 F 13/28 3 1 O G

G O 6 F 11/30 (2006.01)

G O 6 F 11/30 3 2 O B

請求項の数 4 (全 12 頁)

(21) 出願番号 特願平9-361533
 (22) 出願日 平成9年12月26日(1997.12.26)
 (65) 公開番号 特開平11-194996
 (43) 公開日 平成11年7月21日(1999.7.21)
 審査請求日 平成15年11月28日(2003.11.28)

(73) 特許権者 000001007
 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 100077481
 弁理士 谷 義一
 (74) 代理人 100088915
 弁理士 阿部 和夫
 (72) 発明者 松野 裕之
 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 横山 佳弘

(56) 参考文献 特開平06-309233(JP,A)
 特開平05-233526(JP,A)
 最終頁に続く

(54) 【発明の名称】 データ転送装置

(57) 【特許請求の範囲】

【請求項1】

データを複数に分割することによって、DMA転送を繰り返して行うデータ転送装置であって、

前記データのDMA転送の転送時間を計測するタイマーと、

前記タイマーによって計測したDMA転送の1回当たりの転送時間を転送終了毎に測定する転送時間測定手段と、

該測定された転送時間に基づいて、今回と前回の単位バイト数あたりの転送時間を比較し、前回より今回の単位バイト数あたりの転送時間が短くなった場合は、今回よりデータ転送サイズを大きくしてDMA転送させ、前回より今回の単位バイト数あたりの転送時間が長くなった場合は前回のデータ転送サイズを最適な転送サイズとして決定する転送サイズ決定手段とを具え、該決定されたデータ転送サイズに従ってDMA転送を繰り返して行うことを特徴とするデータ転送装置。

10

【請求項2】

前記データの転送エラーが生じた場合、起動したタイマーのタイムアウトにより異常を検知する手段と、

該異常の検知によって前記データのDMA転送を停止させる手段とをさらに具えたことを特徴とする請求項1記載のデータ転送装置。

【請求項3】

データを複数に分割することに

20

よって、DMA転送を繰り返して行うデータ転送方法であって、

DMA転送の1回当たりの転送時間を転送終了毎に測定する工程と、

該測定された転送時間に基づいて、今回と前回の単位バイト数あたりの転送時間を比較し、前回より今回の単位バイト数あたりの転送時間が短い場合は、データ転送サイズを大きくしてDMA転送させ、前回より今回の単位バイト数あたりの転送時間が長くなった場合は前回のデータ転送サイズを最適な転送サイズとして決定する工程とを具え、該決定されたデータ転送サイズに従ってDMA転送を繰り返して行うことを特徴とするデータ転送方法。

【請求項4】

前記データの転送エラーが生じた場合、起動したタイマーのタイムアウトにより異常を検知する工程と、

該異常の検知によって前記データのDMA転送を停止させる工程とをさらに具えたことを特徴とする請求項3記載のデータ転送方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、大容量のデータをDMA(Direct Memory Access)を用いて転送するデータ転送装置に関する。

【0002】

【従来の技術】

従来から大容量のデータ転送を高速に行うには、DMA転送は必須の技術である。この種のデータ転送を高速に行う技術としては、バスマスタデバイスでバスを継続的に使用するバーストモード方式が一般的である。

【0003】

近年、このバースト転送の性能向上のために、クロックと同期することによって、高速転送を行えるSDRAM(Synchronous DRAM)を用いる技術がある。

【0004】

【発明が解決しようとする課題】

しかしながら、高速転送が可能なSDRAMは、クロックパルスと同期するため、電圧降下などによるハードウェアのわずかなタイミングの狂いが生じた場合でも、データ転送エラーという致命的な欠陥が発生しやすく、データに影響を及ぼすという問題点がある。

【0005】

また、転送エラーを起こした場合、必要なデータサイズの転送が終了しないので、ホストCPUは、終了を認識できない。このような場合は、DMAコントローラを停止させ、SDRAMのモード切り換えを指示する必要がある。このモード切り換え処理を行わず、他のメモリアクセスを実行すると、ローカルバスがロックし、システム全体がハングアップする場合がある。

【0006】

さらに、ホストCPUやDMAコントローラの制限によって、14MBのDMA転送を一度に実行できない場合がほとんどであり、また、バーストモード方式は、転送時間が長くなると、バスを占有する時間が長いため、ホストCPU上で動作するOS(Operating System)やアプリケーションのレスポンスが悪くなる傾向がある。

【0007】

そこで、本発明の目的は、データのDMA転送における総転送時間を短縮させ、最適なバス使用効率を設定することが可能なデータ転送装置を提供することにある。

【0008】

【課題を解決するための手段】

本発明は、データを複数に分割することによって、DMA転送を繰り返して行うデータ転送装置であって、前記データのDMA転送の転送時間を計測するタイマーと、前記タイマーによって計測したDMA転送の1回当たりの転送時間を転送終了毎に測定する転送時

10

20

30

40

50

間測定手段と、該測定された転送時間に基づいて、今回と前回の単位バイト数あたりの転送時間を比較し、前回より今回の単位バイト数あたりの転送時間が短くなった場合は、今回よりデータ転送サイズを大きくしてDMA転送させ、前回より今回の単位バイト数あたりの転送時間が長くなった場合は前回のデータ転送サイズを最適な転送サイズとして決定する転送サイズ決定手段とを具え、該決定されたデータ転送サイズに従ってDMA転送を繰り返して行うことによって、データ転送装置を構成する。

【0010】

前記データの転送エラーが生じた場合、起動したタイマーのタイムアウトにより異常を検知する手段と、該異常の検知によって前記データのDMA転送を停止させる手段とをさらに具えることができる。

10

【0012】

本発明は、データを複数に分割することによって、DMA転送を繰り返して行うデータ転送方法であって、DMA転送の1回当たりの転送時間を転送終了毎に測定する工程と、該測定された転送時間に基づいて、今回と前回の単位バイト数あたりの転送時間を比較し、前回より今回の単位バイト数あたりの転送時間が短い場合は、データ転送サイズを大きくしてDMA転送させ、前回より今回の単位バイト数あたりの転送時間が長くなった場合は前回のデータ転送サイズを最適な転送サイズとして決定する工程とを具え、該決定されたデータ転送サイズに従ってDMA転送を繰り返して行うことによって、データ転送方法を提供する。

【0014】

20

前記データの転送エラーが生じた場合、起動したタイマーのタイムアウトにより異常を検知する工程と、該異常の検知によって前記データのDMA転送を停止させる工程とをさらに具えることができる。

【0018】

前記データの転送エラーが生じた場合、起動させたタイマーのタイムアウトにより異常を検知させ、該異常の検知によって前記データのDMA転送を停止させることができる。

【0019】

【発明の実施の形態】

以下、図面を参照して、本発明の実施の形態を詳細に説明する。

【0020】

30

図2は、大画面固体撮像素子を備えた医療用のX線撮影装置6のシステム構成例を示す。このX線撮影装置6は、LAN100と双方向通信可能に接続されている。このLAN100としては、例えばEthernet等の病院内に張り巡らされたネットワークを想定できる。

【0021】

本例では、X線撮影装置6内におけるデータ転送処理を例に挙げる。

【0022】

X線撮影装置6において、大画面固体撮像素子を含むセンサユニット1と、撮影した画像データを蓄えるキャプチャボード2とが、画像転送専用のバス3を介して接続されている。キャプチャボード2上に転送された画像データは、さらにPCIバス7を通じて、ホストCPUボード4上に転送され、画像処理後、ネットワークインターフェース5を通じて、LAN100に接続された外部装置、例えば、医療用のプリンタ200や、他の診断装置201に転送され、医療用の画像診断に利用される。

40

【0023】

図1は、図2のX線撮影装置6におけるデータ転送処理部の構成を示す。

【0024】

このデータ転送処理部は、キャプチャボード2と、ホストCPUボード4とのシステムによって構成される。

【0025】

キャプチャボード2において、11は、ボード内の処理を統括制御するためのローカルCPUである。12は、ローカルCPU11と、ホストCPU20との通信に使用するため

50

の D P R A M である。13 は、センサユニット 1 からデータが入力され記憶される S D R A M である。14 は、S D R A M 13 のモードを切り替えるモードレジスタである。15 は、S D R A M 13 に入力される基準クロック (C L K) である。16 は、S D R A M 13 からのデータが入力される F I F O メモリである。17 は、P C I バスインターフェースである。この P C I バスインターフェース 17 内には、F I F O メモリ 16 内のデータの D M A 転送制御を行うための D M A コントローラ (D M A C) 18 が設けられている。

【 0 0 2 6 】

ホスト C P U ボード 4 において、20 は、本発明に係る図 7 ~ 図 9 の D M A 処理を実行するためのホスト C P U である。なお、本例では、ホスト C P U 18 上で動作する O S (オペレーティングシステム) として、W i n d o w s N T を採用している。21 は、F I F O メモリ 16 から D M A 転送されたデータが入力されるホストメモリである。22 は、本発明に係る図 7 ~ 図 9 の制御プログラム 22 a が記憶される R A M である。23 は、D M A 処理の際に用いられるマップレジスタである。24 は、P C I バスインターフェースである。25 は、キャッシュメモリである。なお、制御プログラム 22 a は、R A M 22 の他に、R O M (図示せず) に記憶したり、また、別体として、フロッピーディスク等の記憶媒体に記憶してもよい。

【 0 0 2 7 】

以下、データ転送処理部の動作について説明する。

【 0 0 2 8 】

データ転送処理部は、D M A 転送の設定とともにタイマー (制御プログラム 22 a により実行されるソフト的な手段であるが、ハード的な手段でも構成が可能である) を起動し、D M A 終了割り込み毎に転送に要した転送時間と転送バイト数を記憶する。

【 0 0 2 9 】

異常が生じた場合には、タイマーのタイムアウト割り込みが入るので、D M A 転送を停止してバスの開放および S D R A M 13 のモード切り換えをするとともに、転送異常をユーザに通知することができる。

【 0 0 3 0 】

正常時には、次の D M A 設定とタイマーを初期化し、以後すべてのデータ転送を終了するまで繰り返す。

【 0 0 3 1 】

以上の基本シーケンスを用いてデータ転送を繰り返すが、ホスト C P U 20 のクロックバスのメモリコントローラのウェイト時間、メモリの速度や種類などハードウェアによって、D M A 転送の時間が異なる。すなわち、一般には D M A 転送を設定するオーバーヘッドを最小にする方が効率良く転送ができるはずだが、ハードウェアの組み合わせを変えて転送サイクルを計測すると、1 度に転送するデータサイズと割り込み回数の組み合わせにおいて、転送時間が異なる。

【 0 0 3 2 】

(メモリ空間)

次に、D M A 転送に係るメモリ空間を、図 3 に基づいて説明する。

【 0 0 3 3 】

X 線撮影装置 6 を制御するユーザプロセスは、O S 上の仮想メモリ空間で動作しており、撮影データを画像処理するために 14 M B (メガバイト) の画像データをキャプチャボード 2 上の S D R A M 13 からホストメモリ 21 に D M A 転送する。

【 0 0 3 4 】

しかし、S D R A M 13、D M A コントローラ 18、ホストメモリ 21 などは、物理メモリ空間でアドレスを指定して動作しているため、ユーザプロセスへデータ転送を実現するためには、図 3 に示すように、ユーザ仮想メモリ 300 と、物理メモリ 301 と、論理メモリ 302 との間においてメモリアドレスの変換が必要となる。この図 3 は、全転送要求されたバイト数 B r に対して、各アドレスで示す所定サイズのブロックに分割してデータを転送する例である。

10

20

30

40

50

【 0 0 3 5 】

ここで、ホストCPU20の一例としてPentiumを用いた場合には、1度にアクセス可能なサイズは4KBのページサイズという制限があるが、WindowsNTには、ホストメモリ21を論理的にOSの空間内に連続領域に割り付けることで、データ転送を1度のDMA起動に対して行う機能がある。この論理的なアドレス変換を行うのがマッピングレジスタであり、m個使用して、4×mKB単位でバースト転送を可能にしている。

【 0 0 3 6 】

そして、SDRAM13のモードをモードレジスタ14で切り替え、バースト転送を行うために、ローカルCPU11は、FIFOメモリ16にデータを入れる初期化を行う。その後、PCIバスインターフェース17に内蔵されたDMAコントローラ18が、FIFO16からホストメモリ19へデータを順次バースト転送するだけの処理になる。このとき、ローカルCPU11は、そのデータ転送中においても、他のプロセスを実行できる。また、ホストCPU20は、所定のDMAサイズに分割されたデータに対して割り込み処理を行い、次のDMA転送の設定および起動以外は、他のプロセスを実行することができる。

10

【 0 0 3 7 】

(実験例)

次に、センサユニット1で撮影した14MBの画像データをDMA転送するために、分割するサイズについて考える。

【 0 0 3 8 】

図4は、マッピングレジスタ23の個数を変え、1回のDMA転送のサイズを変化させて、14MB(ただし、1MB=1024KB)の画像データ全体の転送時間を計測した一例である。

20

【 0 0 3 9 】

転送サイズを規定するマッピングレジスタ23の個数がm=4のときは、(14×1024)/16KB=896回のDMA起動および割り込み後処理などのI/O処理を行わなければならない。また、m=16のときは、224回のI/O処理になる。

【 0 0 4 0 】

しかしながら、OSの論理的連続メモリ空間にあるDMA転送されたデータを、仮想メモリ空間にあるユーザプロセスに対してデータを渡すためには、DMAコントローラ18内のキャッシュメモリ、および、ホストメモリ21内の論理空間あるいはホストCPUボード4上のキャッシュメモリ25を明示的にメモリフラッシュ(キャッシュメモリ上のデータを吐き出すためのメモリコピー)する必要がある。このメモリフラッシュ中は、次のDMA転送を行えないため、PCIバス7上は遊んでいる状態が継続し、メモリフラッシュ終了後に次のDMA転送を起動できる。

30

【 0 0 4 1 】

図5および図6は、回路上の信号線の状態を示す。図5は、64KB単位でDMA転送を行う場合の例である。30は、DMA転送サイズのレジスタにアクセスする時間を計測したものである。1回の転送について実際にバースト転送している転送時間は31のように短く、30から31を引いた32の部分が、転送終了割り込み後の処理になる。このとき、割り込みの後処理時間比は、

40

【 0 0 4 2 】

【数1】

$$\frac{5.33-1.50}{5.33} \times 100 \div 72\% \quad \dots (1)$$

【 0 0 4 3 】

となる。

【 0 0 4 4 】

また、図6は、16KB単位でDMA転送を行う場合の例である。このとき、割り込みの

50

後処理時間比は、

【 0 0 4 5 】

【 数 2 】

$$\frac{1.10 - 0.38}{1.10} \times 100 \div 65\% \quad \dots (2)$$

【 0 0 4 6 】

となる。この場合、割込み後処理に時間がかかっているが、その割合が図 5 に比べて 7 2 % から約 6 5 % と少なくなっている。この後処理時間のほとんどがメモリフラッシュにか

10

【 0 0 4 7 】

以上の実験結果から、CPUクロック、バスクロック、メモリスピード、メモリアクセスウェイト、キャッシュサイズなどのハードウェア構成によって、1 度に DMA 転送を行う最適なサイズ、すなわち、マップレジスタ 2 3 の個数 m が異なる結果になる。

【 0 0 4 8 】

すなわち、ホスト CPU 2 0 やホストメモリ 2 1 が十分に速い場合、さらにメモリアクセスウェイトが小さい場合やキャッシュメモリ 2 5 が多い場合は、DMA 転送の起動および割込み後処理にかかる CPU オーバヘッドが多くても、PCI バス 7 の使用率を高める方が全体の転送時間を短くすることができる。一方、このオーバヘッドが多くなりすぎると、ホスト CPU 1 8 が遅い場合や、チップセットの性能により、かえって転送時間がかかる結果になる。

20

【 0 0 4 9 】

(DMA 転送処理)

次に、DMA 転送の処理を、図 7 ~ 図 1 1 に基づいて説明する。以下に示す処理は、ホスト CPU 2 0 が制御プログラム 2 2 a を用いて行う制御例である。

【 0 0 5 0 】

本処理では、画像データを複数に分割して DMA 転送を繰り返して行う場合において、自動的に最適な転送サイズに近づける処理を行う。

30

【 0 0 5 1 】

まず、図 7 に示すように、ステップ S 1 0 1 において、ユーザプロセスからの要求により、初期化処理を行う。この場合、係数は、以下のような内容とする。

【 0 0 5 2 】

m : マップレジスタの個数

F : 最適な転送サイズが決定済か否かを示すフラグ

T p : 前回の DMA 転送に要した単位バイト数当たり (ここでは 4 K B) の転送時間

T n : 今回の DMA 転送に要した単位バイト数当たり (ここでは 4 K B) の転送時間

t : 今回の DMA 転送に要した転送時間

40

B n : 今回の DMA 転送に要したバイト数

B t : 現在までの転送済のバイト数

B r : 全転送要求されたバイト数

また、転送先アドレスを計算し、DMA 転送の設定および起動を行う。これにより、タイマーがスタートし、F I F O メモリ 1 6 からホストメモリ 2 1 への画像データの DMA 転送が開始される。

【 0 0 5 3 】

次に、図 8 に示すように、ステップ S 1 0 2 では、DMA 終了割込み処理に入る。そして、ステップ S 1 0 3 では、DMA コントローラ 1 8 が DMA 転送処理終了を示す割り込み信号を発生したかをチェックする。

50

【 0 0 5 4 】

D M A 終了割り込みを発生した場合には、ステップ S 1 0 4 に進む。ステップ S 1 0 4 では、タイマーを停止し、D M A 転送に要した転送時間 t を記憶する。

ステップ S 1 0 5 では、現在までに転送済のバイト数 B_t を算出する。

【 0 0 5 5 】

ステップ S 1 0 6 では、現在までの転送済のバイト数 B_t が、全転送要求されたバイト数 B_r を超えているか否かをチェックする。

【 0 0 5 6 】

ステップ S 1 0 6 において、現在までのバイト数 B_t が全バイト数 B_r を超えた場合は、ステップ S 1 0 7 に進み、S D R A M 1 3 のモード切り換えを行う。そして、ステップ S 1 0 8 にて、D M A 転送を正常な状態で終了する。

10

【 0 0 5 7 】

一方、現在までのバイト数 B_t が全バイト数 B_r を超えていない場合は、図 9 に示すステップ S 1 1 0 に進む。ステップ S 1 1 0 以降の処理は、最適な転送サイズを決定する処理である。

【 0 0 5 8 】

ステップ S 1 1 0 では、画像データの転送サイズが最適な転送サイズになっているか否かをチェックする。最適な転送サイズになっていない場合には、ステップ S 1 1 1 に進む。

【 0 0 5 9 】

ステップ S 1 1 1 では、単位バイト数当たりの転送時間 T_p を算出する。そして、ステップ S 1 1 2 に進み、前回の転送時間が有効か否かをチェックする。

20

【 0 0 6 0 】

ステップ S 1 1 2 において、前回の転送時間が有効でない場合は、ステップ S 1 1 3 に進み、個数 m と、前回の転送時間 T_p との値を入れ替える。一方、前回の転送時間が有効である場合は、ステップ S 1 1 4 に進み、前回の転送時間 T_p と今回の転送時間 T_n とを比較する。

【 0 0 6 1 】

ステップ S 1 1 4 において、今回の転送時間 T_n が前回の転送時間 T_p よりも短い場合は、ステップ S 1 1 3 に進む。一方、今回の転送時間 T_n が前回の転送時間 T_p よりも長い場合は、ステップ S 1 1 5 に進み、個数 m を入れ替えると共に、フラグ F を T R U E とすることにより、最適な転送サイズが決定される。

30

【 0 0 6 2 】

その後、ステップ S 1 1 6 に進む。このステップ S 1 1 6 は、初期ルーチンのステップ S 1 0 1 に対応するものである。この場合、今回の転送バイト数 B_t を新たに算出する。そして、D M A 転送の設定および起動を行い、タイマーをスタートさせて、前回と同様に、F I F O メモリ 1 6 からホストメモリ 2 1 への画像データの D M A 転送を開始する。以後、ステップ S 1 0 2 に進み、同様な処理を繰り返す。

【 0 0 6 3 】

また、ステップ S 1 1 0 において、最適な転送サイズになっていた場合には、ステップ S 1 1 6 に直接進んで処理を続行する。

40

【 0 0 6 4 】

以上述べたように、最適な転送サイズを動的に探しながら、D M A 転送を繰り返して行う。

【 0 0 6 5 】

また、万一、D M A コントローラ 1 8 等のハードウェアの不具合が生じた場合には、ステップ S 1 0 3 において、D M A 転送処理終了を示す割り込みがないと判断して、ステップ S 2 0 1 に進み、D M A タイムアウト処理を実行する。

【 0 0 6 6 】

ステップ S 2 0 1 では、タイマーを停止し、S D R A M 1 3 のモード切り換えを行う。そして、ステップ S 2 0 2 に進んで、ユーザに異常状態を通知し、全ての処理を終了する。

50

ステップS203)。このようなタイムアウト処理により、システム全体を誤動作させることなく、ユーザに異常を通知し、保守点検を促すことができる。

【0067】

また、図10～図11は、上述した最適な転送サイズを決定する処理を模式的に示したものである。

【0068】

図10の例では、初期設定の後、2回目の処理で最適な転送サイズを決定し、以後 $m = 1$ として最後の処理まで転送処理を続行する。

【0069】

図11の例では、初期設定の後、4回目の処理で最適な転送サイズを決定し、以後 $m = 4$ として最後の処理まで転送処理を続行する。

10

【0070】

なお、本発明は、複数の機器から構成されるシステムに適用しても、1つの機器からなる装置に適用してもよい。また、本発明はシステム或いは装置にプログラムを供給することによって達成される場合にも適用できることはいうまでもない。この場合、本発明を達成するためのソフトウェアによって表されるプログラムを格納した記憶媒体を該システム或いは装置に読み出すことによって、そのシステム或いは装置が、本発明の効果を享受することが可能となる。

【0071】

【発明の効果】

20

以上説明したように、本発明によれば、データを複数に分割してDMA転送を繰り返して行う場合において、DMA転送の1回当たりの転送時間および転送バイト数を逐次比較監視しながら、転送サイズを各転送単位で動的に変化させるようにしたので、データのDMA転送の総転送時間を短縮させることができる。

【0072】

また、これにより、ハードウェアの組み合わせが幾通りあっても、何度も実験を繰り返して最適値を求めなくても、自動的に最適値に近似することが可能なため、バス上のデータ転送が多くなっている場合にも、自動的に最適なバス使用効率に近づけることができる。

【図面の簡単な説明】

【図1】本発明の実施の形態であるデータ転送装置の構成を示すブロック図である。

30

【図2】本発明に適用される医療用システムの構成例を示すブロック図である。

【図3】DMA転送に係るメモリ空間を説明する説明図である。

【図4】DMA転送のサイズと転送時間との関係を示す説明図である。

【図5】DMA転送の設定と転送時間との関係を示す説明図である。

【図6】DMA転送の設定と転送時間との関係を示す説明図である。

【図7】DMA転送の制御を示すフローチャートである。

【図8】図7に続くDMA転送の制御を示すフローチャートである。

【図9】図8に続くDMA転送の制御を示すフローチャートである。

【図10】最適な転送サイズを決定する処理を模式的に説明する説明図である。

【図11】最適な転送サイズを決定する処理を模式的に説明する説明図である。

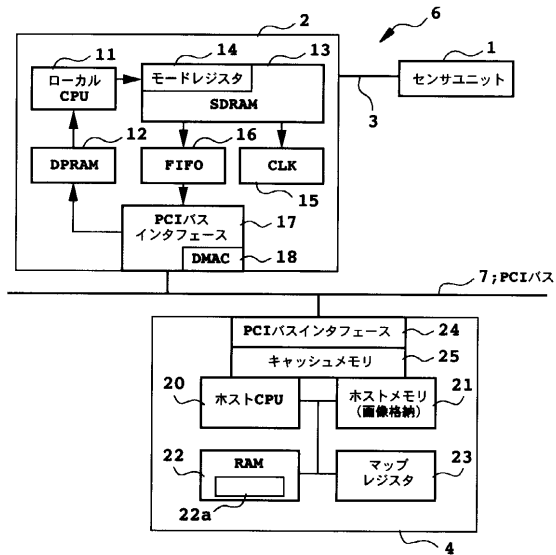
40

【符号の説明】

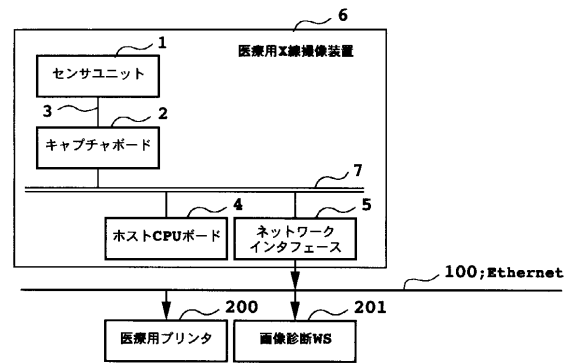
20 ホストCPU

22a 制御プログラム

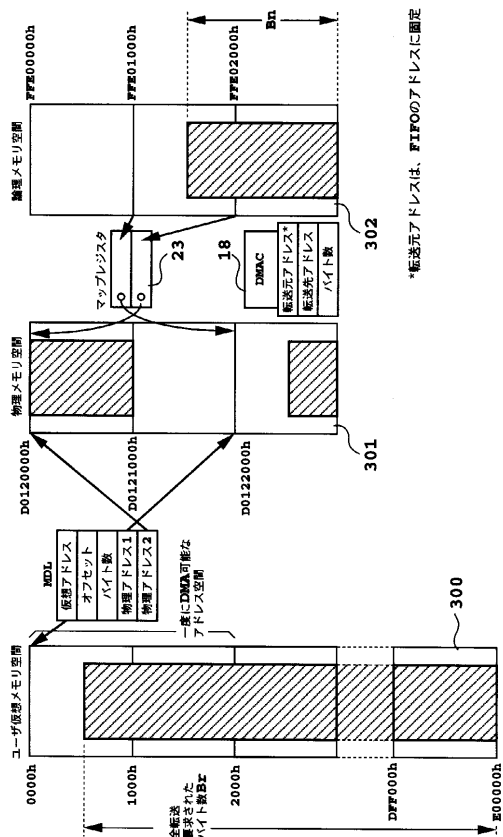
【図1】



【図2】



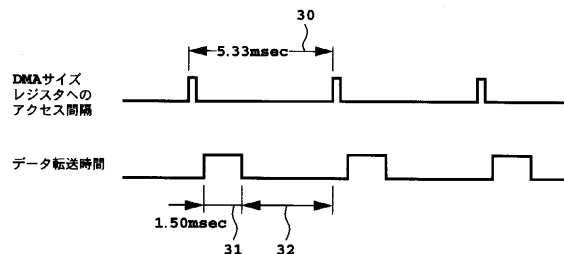
【図3】



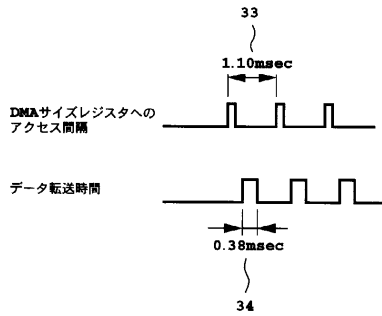
【図4】

1回のDMA転送単位 (KB)	14KBの転送時間 (msec)	1回のDMA平均転送時間 (μsec)	4KBあたりの平均転送時間 (μsec)
4KB	1043ms	291 μsec	291 μsec
8KB	1012ms	564 μsec	282 μsec
16KB	1015ms	1132 μsec	283 μsec
32KB	1080ms	2410 μsec	301 μsec
64KB	1194ms	5330 μsec	333 μsec

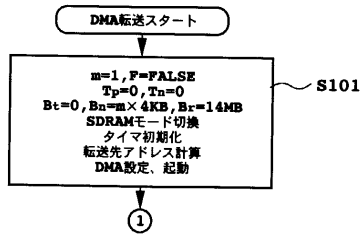
【図5】



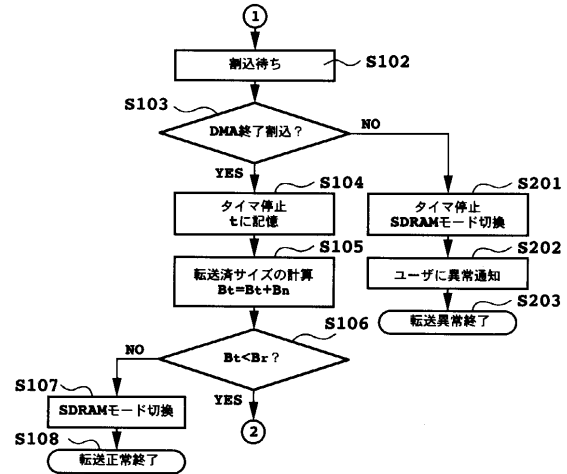
【図 6】



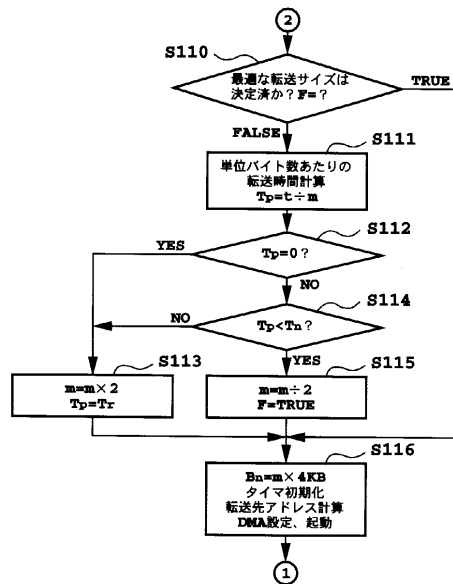
【図 7】



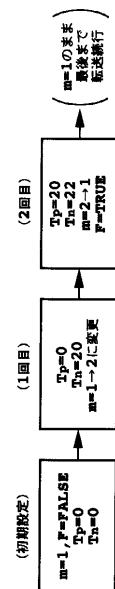
【図 8】

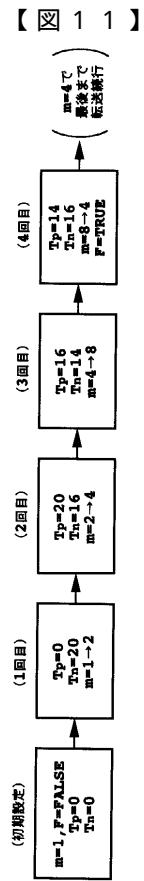


【図 9】



【図 10】





フロントページの続き

(58)調査した分野(Int.Cl. , D B 名)

G06F 13/28

G06F 11/30