

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5205603号  
(P5205603)

(45) 発行日 平成25年6月5日(2013.6.5)

(24) 登録日 平成25年3月1日(2013.3.1)

(51) Int.Cl. F I  
**G 0 6 F 13/28 (2006.01)** G O 6 F 13/28 3 1 O B  
**G 0 6 F 13/38 (2006.01)** G O 6 F 13/28 3 1 O J  
 G O 6 F 13/38 3 1 O F

請求項の数 10 (全 20 頁)

(21) 出願番号 特願2010-226063 (P2010-226063)  
 (22) 出願日 平成22年10月5日 (2010.10.5)  
 (65) 公開番号 特開2011-81801 (P2011-81801A)  
 (43) 公開日 平成23年4月21日 (2011.4.21)  
 審査請求日 平成22年12月3日 (2010.12.3)  
 (31) 優先権主張番号 0957033  
 (32) 優先日 平成21年10月8日 (2009.10.8)  
 (33) 優先権主張国 フランス (FR)

(73) 特許権者 510191207  
 コミサリアト ア レネルジー アトミック  
 ー エ オ エネルギーズ アルタナティ  
 ヴズ  
 フランス国 75015 パリ, リュー  
 ルブラン パティモン <<ル ポノン  
 デー>> 25  
 (74) 代理人 100091683  
 弁理士 ▲吉▼川 俊雄  
 (72) 発明者 イヴ ドゥロン  
 フランス 38330 サン-イスマエ,  
 シュマン デ プロンテ 41

最終頁に続く

(54) 【発明の名称】 複数のソースを有する直接アクセスメモリコントローラ、それに対応する方法及びコンピュータプログラム

(57) 【特許請求の範囲】

【請求項1】

いくつかのバッファメモリ(16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>)を通して、データをいくつかのデータソース(12<sub>1</sub>、・・・、12<sub>i</sub>、・・・、12<sub>n</sub>)から、それらデータの少なくとも1つのアドレス(14)に転送する直接メモリアクセスコントローラ(10、20)であって、前記バッファメモリ(16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>)に記憶されているデータを読み取り、前記アドレス(14)にシーケンスで転送するように設計された読み取り管理モジュール(30)と、各バッファメモリにそれぞれ関連付けられた読み取りポインタ(PL1、PL2)を記憶する記憶モジュール(又は記憶手段)(38)と、を備え、各前記読み取りポインタは、データを読み取ることができる関連付けられた前記バッファメモリの基本ロケーションを示し、前記バッファメモリ(16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>)のそれぞれに各データソース(12<sub>1</sub>、・・・、12<sub>i</sub>、・・・、12<sub>n</sub>)が関連付けられ、前記バッファメモリ毎に、前記コントローラ(10、20)は、ファームウェア(40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>)を実行して、データを読み取り、前記バッファメモリに関連付けられた前記読み取りポインタを更新する手段を備え、かつ、前記バッファメモリにそれぞれ関連付けられた前記データソースから発せられたデータの所定の順序に応じて、ファームウェアの実行を互いに同期させる手段(30、40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>)を備え、前記所定の順序は、データを前記アドレス(14)に転送するために、前記バッファメモリにそれぞれ関連付けられた前記データソースから発せられたデータの所定の順序により形

10

20

成されるように、データシーケンス内で要求されることを特徴とする、直接メモリアクセスコントローラ(10、20)。

【請求項2】

前記同期手段(30、40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>)が、少なくとも1つの読み取り・更新権を有するファームウェアから別のファームウェアに送信する手段を備え、現在実行中のあるファームウェア(40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>)が読み取り・書き込み権を保持することにより、前記ファームウェアは、関連付けられた前記バッファメモリ(16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>)内の少なくとも1回の読み取り及び対応する前記読み取りポイント(PL1、PL2)の少なくとも1回の更新を実行することが可能である、請求項1に記載の直接メモリアクセスコントローラ(10、20)。

10

【請求項3】

各バッファメモリ(16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>)は、所定数の連続した基本メモリロケーションを含み、且つ循環式のものであり、任意の1つの前記循環バッファメモリの各基本ロケーションは、この循環バッファメモリの最下位アドレス(PB1、PB2)と最上位アドレス(PS1、PS2)との間のアドレスにより識別され、前記記憶手段(38)は、各循環バッファメモリに関連付けられ、データを書き込むことができる、関連付けられた前記バッファメモリの基本ロケーションをそれぞれ示す書き込みポイント(PE1、PE2)を備え、前記コントローラは、ソース(12<sub>1</sub>、・・・、12<sub>i</sub>、・・・、12<sub>n</sub>)から受信したデータを、対応する前記循環バッファメモリ内の、対応する前記書き込みポイント(PE1、PE2)で示される前記アドレスに書き込むように設計された書き込み管理モジュール(24)並びに各循環バッファメモリの前記読み取りポイント(PL1、PL2)及び前記書き込みポイント(PE1、PE2)により示される前記基本ロケーションの相対位置に応じて、各循環バッファメモリでの読み取り、書き込みのそれぞれを阻止する手段(24、30、40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>)も備える、請求項1又は2に記載の直接メモリアクセスコントローラ(10、20)。

20

【請求項4】

任意の1つの前記循環バッファメモリ(16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>)内の前記読み取り阻止手段(30、40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>)は、読み取りポイント(PL1、PL2)への更新が、読み取りポイント(PL1、PL2)が指すアドレスが、書き込みポイント(PE1、PE2)が指すアドレスに達するか、又はそれを超えることを意味する場合、この循環バッファメモリ内のいかなるデータ読み取りも阻止するように設計され、任意の1つの前記循環バッファメモリ(16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>)内の前記書き込み阻止手段(24、40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>)は、書き込みポイント(PE1、PE2)への更新が、書き込みポイント(PE1、PE2)が指すアドレスが、読み取りポイント(PL1、PL2)が指すアドレスに達するか、又はそれを超えることを意味する場合、この循環バッファメモリ内のいかなるデータ書き込みも阻止するように設計される、請求項3に記載の直接メモリアクセスコントローラ(10、20)。

30

【請求項5】

循環バッファメモリ(16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>)毎に、前記記憶手段(38)は、前記循環バッファメモリに関連付けられた前記読み取りポイント(PL1、PL2)とは異なる「作業ポイント」(PT1、PT2)と呼ばれる読み取りポイントも備え、この作業ポイント(PT1、PT2)は、データを読み取らなければならない、関連付けられた前記バッファメモリの基本ロケーションを示し、この循環バッファメモリに関連付けられた前記ファームウェアは、所定の更新論理を使用して、この循環バッファメモリでの読み取りの都度、前記作業ポイントを更新するようにも設計される、請求項3又は4に記載の直接メモリアクセスコントローラ(10、20)。

40

【請求項6】

各循環バッファメモリの前記作業ポイント(PT1、PT2)の前記所定の更新論理は

50

、データが書き込まれた順序とは異なる順序で、含まれるデータを読み取るように、この循環バッファメモリにおける少なくとも1つのアドレスジャンプを含む、請求項5に記載の直接メモリアクセスコントローラ(10、20)。

【請求項7】

請求項1～6のいずれか一項に記載の直接メモリアクセスコントローラ(10、20)により、所定数の連続した基本メモリロケーションをそれぞれ含むいくつかのバッファメモリ(16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>)を通して、データをいくつかのデータソース(12<sub>1</sub>、・・・、12<sub>i</sub>、・・・、12<sub>n</sub>)からそれらデータの少なくとも1つのアドレス(14)に転送する方法であって、任意の1つの前記ソースから受信された各データは、書き込み管理モジュール(24)により、このソースに関連付けられたバッファメモリに送信され(106)、前記アドレス(14)に送信される各データは、前記コントローラにより前記バッファメモリのうちの1つにおいて読み取られ(118、220)、該方法が、

- バッファメモリ毎に、前記コントローラが、ファームウェア(40<sub>1</sub>、40<sub>2</sub>)を実行して、このバッファメモリから前記データを読み取り、データを読み取ることができるこのバッファメモリ内の基本ロケーションを示す読み取りポインタを更新するステップ(112、212)と、

- 前記バッファメモリにそれぞれ関連付けられた前記データソースから導出されるデータの所定の順序に応じて、前記ファームウェア(40<sub>1</sub>、40<sub>2</sub>)の実行を互いに同期させるステップ(122、124、214、224)と、を含む、

前記所定の順序は、データを前記アドレス(14)に転送するために、前記バッファメモリにそれぞれ関連付けられた前記データソースから発せられたデータの所定の順序により形成されるように、データシーケンス内で要求されることを特徴とする、方法。

【請求項8】

前記同期ステップ(122、124、214、224)は、現在実行中の異なる前記ファームウェア間での少なくとも1つの読み取り・更新権のプログラムされた循環により行われ、現在実行中の前記ファームウェアのうちの1つが前記読み取り・更新権を一時的に保持することにより、前記ファームウェアは、関連付けられた前記バッファメモリ内の少なくとも1回の読み取り及び対応する前記読み取りポインタの少なくとも1回の更新を実行することができる、請求項7に記載のデータ転送方法。

【請求項9】

現在実行中の異なる前記ファームウェア間での少なくとも1つの読み取り・更新権のプログラムされた循環は、以下：

- 各ファームウェアの実行が、このファームウェア内にプログラムされた命令に従って前記読み取り・更新権を待つ少なくとも1つのステップ(124、214)を含み、

- 各ファームウェアの実行が、少なくとも1回の読み取り及びこのファームウェアに関連付けられた前記読み取りポインタの少なくとも1回の更新を実行した後、このファームウェアにプログラムされた命令に従って、前記読み取り・更新権を現在実行中の別のファームウェアに送信する少なくとも1つのステップ(122、224)を含む

ように行われる、請求項8に記載のデータ転送方法。

【請求項10】

コンピュータにより読み取り可能であり、且つ/又は直接メモリアクセスコントローラ(10、20)により実行可能な、通信ネットワークからダウンロード可能であり、且つ/又は媒体に記録可能なコンピュータプログラムであって、前記プログラムが直接メモリアクセスコントローラにより実行された場合、請求項7～9のいずれか一項に記載のデータ転送方法のステップを実行するプログラムコード命令を含むことを特徴とする、コンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

10

20

30

40

50

本発明は、複数のソースを使用する直接メモリアクセスコントローラ、データをいくつかのデータソースからこのコントローラにより使用されるそれらデータの少なくとも1つのアドレスに転送する方法、及びそれに対応するコンピュータプログラムに関する。

【背景技術】

【0002】

そのようなコントローラは、DMA（直接メモリアクセス）コントローラと呼ばれ、通常、例えば、集積回路等の中央演算処理装置を有するデータ処理装置に使用される。コントローラは、転送の開始及び終了を除いて中央演算処理装置による介入なしで、1つ又はいくつかの中央演算処理装置により処理又は供給されたデータを、通信ポート、ハードディスク、又は任意のメモリ等の少なくとも1つの周辺機器から、又はそのような周辺機器に転送し、例えば、この周辺機器と装置内の1つ又はいくつかのローカルメモリとの間でデータを転送する。

10

【0003】

例えば、DMAコントローラは、高速周辺機器への繰り返しアクセスが、中央演算処理装置により行われている処理をブロックしない場合でも少なくとも遅くさせる危険性があるシステムにおいて非常に有用であることができる。コントローラの存在により、中央演算処理装置により実行されるソフトウェアアプリケーションの処理時間が最適化され、DMAコントローラは、ローカルメモリからのデータ転送及びローカルメモリへのデータ転送の管理を任せられる。

【0004】

20

本発明は特に、所定数の連続した基本メモリロケーションをそれぞれ含むいくつかのバッファメモリを通してデータをいくつかのデータソースからそれらデータの少なくとも1つのアドレスに転送する直接メモリアクセスコントローラに適用可能であり、アクセスコントローラは、バッファメモリに記憶されているデータを読み取り、順番にアドレスに転送するように設計された読み取り管理モジュールと、それぞれ各バッファメモリに関連付けられた読み取りポインタ記憶手段とを備え、各読み取りポインタは、データを読み取ることができる、各読み取りポインタが関連付けられたバッファメモリの基本ロケーションを示す。

【0005】

例えば、そのようなDMAコントローラは、（特許文献1）として公開された特許に記載されている。この特許には、データが明らかに互いに同期されていない異なるソースから発せられた場合でのデータがDMAコントローラからアドレスに送信される順番についての一般には解決されていない問題が述べられている。

30

【0006】

この文献において与えられる解決策は、データが書き込まれるいくつかのバッファメモリを提供することからなり、それにより、DMAコントローラは、バッファメモリ内のデータを読み取ってアドレスに転送することができ、それらバッファメモリの順序付きリストがDMAコントローラに供給され、バッファメモリのうちの1つのすべてのデータが、DMAコントローラにより読み取られてアドレスに転送された場合、順序付きリスト内の次のバッファメモリの読み取りが開始される。したがって、それらデータの送信元である中央演算処理装置によるいかなる動作もなく、データを要求される所定の順番でアドレスに転送することが可能である。

40

【0007】

しかし、この解決策は、DMAコントローラにバッファの順序付きリストの管理を余儀なくさせる。さらに、多数の小さなバッファメモリがこのリスト内に存在する場合であっても、特定の限度を超えて正確にいくつかのソースを同期させることはできない。例えば、2つの異なるソースから導出されたデータをこのようにしてインタレースさせて送信することは不合理である。

【先行技術文献】

【特許文献】

50

【 0 0 0 8 】

【特許文献 1】米国特許第 5, 9 0 1, 2 9 1 号明細書

【発明の概要】

【発明が解決しようとする課題】

【 0 0 0 9 】

したがって、上述した問題及び制約を解消する直接メモリアクセスコントローラを提供することが望ましいであろう。

【課題を解決するための手段】

【 0 0 1 0 】

したがって、本発明の主旨は、所定数の連続した基本メモリロケーションをそれぞれ含むいくつかのバッファメモリを通してデータをいくつかのデータソースからそれらデータの少なくとも 1 つのアドレスに転送する直接メモリアクセスコントローラであり、アクセスコントローラは、バッファメモリに記憶されているデータを読み取って順番にアドレスに転送するように設計された読み取り管理モジュールと、それぞれ各バッファメモリに関連付けられた読み取りポインタ記憶手段とを備え、各読み取りポインタは、データを読み取ることができる、各読み取りポインタが関連付けられたバッファメモリの基本ロケーションを示し、バッファメモリのそれぞれには各データソースが関連付けられ、バッファメモリ毎に、コントローラは、ファームウェアを実行してデータを読み取り、このバッファメモリに関連付けられた読み取りポインタを更新する手段を備え、アドレスに転送すべきデータシーケンス内で必要とされるバッファメモリから送信されたデータの所定の順番に応じて、ファームウェアの実行を同期させる手段を備える。

10

20

【 0 0 1 1 】

ファームウェアは、集積回路、より正確にはこの構成要素の揮発性又は不揮発性のメモリ等のハードウェア構成要素内のオンボードソフトウェアを意味する。

【 0 0 1 2 】

したがって、データを読み取り、読み取りポインタを更新するファームウェアを、それ自体が特定のソースに関連付けられた各バッファメモリに関連付けることにより、且つファームウェアの実行を同期させて、異なるバッファメモリ（ひいては異なるソース）を送信元とするデータをアドレスに転送しなければならない順番を決定論的に定義することにより、DMA コントローラをデータソースに同期させる必要なく、且つバッファメモリを互いに並べる必要なく、転送シーケンスの詳細を管理することができる。

30

【 0 0 1 3 】

任意的に、同期手段は、少なくとも 1 つの読み取り・更新権をあるファームウェアから別のファームウェアに送信する手段を備え、現在実行中のあるファームウェアが読み取り・更新権を一時的に保持することにより、そのファームウェアは、関連付けられたバッファメモリ内の少なくとも 1 回のデータ読み取り及び対応する読み取りポインタの少なくとも 1 回の更新を実行することができる。

【 0 0 1 4 】

これもまた任意的に、各バッファメモリは循環式であり、任意の 1 つの循環バッファメモリの各基本ロケーションは、この循環バッファメモリの最下位アドレスと最上位アドレスとの間のアドレスにより識別され、記憶手段は、各循環バッファメモリに関連付けられた書き込みポインタを備え、各書き込みポインタは、データを書き込むことができる、関連付けられたバッファメモリの基本ロケーションを示し、コントローラは、ソースから受信したデータを対応する循環バッファメモリ内の、対応する書き込みポインタにより示されるアドレスに書き込むように設計された書き込み管理モジュール、並びに各循環バッファメモリの上記読み取りポインタ及び上記書き込みポインタにより示される基本ロケーションの相対位置に応じて、各循環バッファメモリ内の読み取り、書き込みのそれぞれを阻止する手段も備える。

40

【 0 0 1 5 】

したがって、各循環バッファメモリは、対応するソースとアドレスとの間で 1 つの中間

50

読み取り/書き込み阻止機能を実行する。この機能は、各循環バッファメモリに対して書き込みアクセス及び読み取りアクセスを行うことができる直接メモリアクセスコントローラにより各循環バッファメモリに関連付けられた読み取りポインタ及び書き込みポインタにより管理される。このようにして、中央演算処理装置に同期する必要なく、入力データフロー（ソースにより生成されたフロー）を出力データフロー（アドレスにより生成されたフロー）に同期させることが可能である。

【0016】

これもまた任意的に、任意の1つの循環バッファメモリ内の読み取り阻止手段は、読み取りポインタへの更新が、書き込みポインタが指すアドレスに到達するか、又は書き込みポインタが指すアドレスを超えることを意味する場合、この循環バッファメモリ内のいかなるデータ読み取りも阻止するように設計され、任意の1つの循環バッファメモリ内の書き込み阻止手段は、書き込みポインタへの更新が、書き込みポインタが指すアドレスが、読み取りポインタが指すアドレスに達するか、又はそれを超えることを意味する場合、この循環バッファメモリ内のいかなるデータ書き込みも阻止するように設計される。

10

【0017】

これもまた任意的に、循環バッファメモリ毎に、記憶手段は、循環バッファメモリに関連付けられた上記読み取りポインタと異なる「作業ポインタ」と呼ばれる読み取りポインタも備え、この作業ポインタは、データを読み取らなければならない、関連付けられたバッファメモリの基本ロケーションを示し、この循環バッファメモリに関連付けられたファームウェアも、所定の更新論理を使用してこの循環バッファメモリの読み取り毎に作業ポインタを更新するように設計される。

20

【0018】

これもまた任意的に、各循環バッファメモリの作業ポインタの所定の更新論理は、データが書き込まれた順番とは異なる順番で、含まれるデータを読み取るようなこの循環バッファメモリ内の少なくとも1つのアドレスジャンプを含む。

【0019】

本発明の別の目的は、上に定義されたような直接メモリアクセスコントローラにより、所定数の連続した基本メモリロケーションをそれぞれ備えるいくつかのバッファメモリを通していくつかのデータソースからそれらデータの少なくとも1つのアドレスにデータを転送する方法であり、任意の1つのソースから受信された各データは、書き込み管理モジュールにより、このソースに関連付けられたバッファメモリに送信され、アドレスに送信されるべき各データは、コントローラによりバッファメモリの1つにおいて読み取られ、この方法は、

30

- バッファメモリ毎に、ファームウェアのコントローラが、このバッファメモリからのデータの読み取り及びデータを読み取ることができるこのバッファメモリ内の基本ロケーションを示す読み取りポインタの更新を実行するステップと、

- アドレスに転送すべきデータシーケンス内で要求されるバッファメモリから導出されるデータの所定の順番に応じて、ファームウェアの実行を同期させるステップとを含む。

【0020】

40

任意的に、同期ステップは、現在実行中の異なるファームウェア間での少なくとも1つの読み取り・更新権のプログラムされた循環により行われ、現在実行中のファームウェアの1つが読み取り・更新権を一時的に保持することにより、そのファームウェアは、関連付けられたバッファメモリにおける少なくとも1回のデータ読み取り及び対応する読み取りポインタの少なくとも1回の更新を実行することが可能になる。

【0021】

これもまた任意的に、現在実行中の異なるファームウェア間での読み取り・更新権のプログラムされた循環は、以下のように行われる。

- 各ファームウェアの実行が、このファームウェア内にプログラムされた命令に従って読み取り・更新権を待つ少なくとも1つのステップを含み、

50

- 各ファームウェアの実行が、少なくとも1回の読み取り及びこのファームウェアに関連付けられた読み取りポイントの少なくとも1回の更新を実行した後、このファームウェア内にプログラムされた命令に従って、読み取り・更新権を現在実行中の別のファームウェアに送信する少なくとも1つのステップを含む。

【0022】

最後に、本発明の別の目的は、コンピュータにより読み取り可能であり、且つ/又は直接メモリアクセスコントローラにより実行可能な、通信ネットワークからダウンロード可能であり、且つ/又は媒体に記録可能なコンピュータプログラムであって、上記プログラムが直接メモリアクセスコントローラにより実行された場合、上に定義されたようなデータ転送方法のステップを実行するプログラムコード命令を含む、コンピュータプログラム

10

【0023】

本発明は、添付図面を参照して単なる例として与えられる以下の説明を読んだ後、よりよく理解されるであろう。

【図面の簡単な説明】

【0024】

【図1】本発明の一実施形態による直接メモリアクセスコントローラの一般的な構造を概略的に示す。

【図2】図1の直接メモリアクセスコントローラが、2つのソースから導出されるインタレースしたデータ転送アプリケーション内で機能する有利な方法を概略的に示す。

20

【図3A】図1の直接メモリアクセスコントローラにより管理される循環バッファメモリ構造を概略的に示す。

【図3B】図1の直接メモリアクセスコントローラにより管理される循環バッファメモリ構造を概略的に示す。

【図4】本発明の一実施形態によるデータ転送方法の連続ステップを示す。

【発明を実施するための形態】

【0025】

図1に示される直接メモリアクセスコントローラ又はDMAコントローラ10は、いくつかのデータソース12<sub>1</sub>、・・・、12<sub>i</sub>、・・・、12<sub>n</sub>から、所定数の連続した基本メモリロケーションをそれぞれ含む複数のバッファメモリ16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>を通して、これらデータの少なくとも1つのアドレス14に転送する機能を実行する。

30

【0026】

本発明が使用可能になる前は、DMAコントローラ10は、読み取る際、及び任意的に書き込む際に各バッファメモリ16<sub>i</sub>にアクセス可能でなければならない。しかし、図1に示される例では、書き込み及び読み取りの際に、DMAコントローラ10とインターフェースする書き込み・読み取りデータアクセスコマンドモジュール18及びバッファメモリ16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>を統合した記憶媒体を通して、各バッファメモリ16<sub>i</sub>の基本ロケーションにアクセスすることが可能である。一変形実施形態では、参照20として記されたDMAコントローラが、バッファメモリ16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>を統合した記憶媒体及び読み取りアクセスコマンドモジュール18を含むことに留意する。実際には、DMAコントローラは、スタンドアロンチップ、例えば、CMOS型集積回路上のシステムの形態であり得る。

40

【0027】

非限定的な実施例として、NoC(ネットワークオンチップ)と呼ばれる1つの可能なアーキテクチャによれば、DMAコントローラ10又は20は、パケット交換ネットワーク内でリンクを通して互いに通信する機能要素で構成されるものとして考えられる。

【0028】

このアーキテクチャによれば、DMAコントローラ10又は20は、ソース12<sub>1</sub>、・・・、12<sub>i</sub>、・・・、12<sub>n</sub>からアドレス14にデータを転送する以下の要素：

50

- ソース  $12_1, \dots, 12_i, \dots, 12_n$  から発せられたデータの受信リンク 22、

- コントローラ 10 又は 20 が、バッファメモリ  $16_1, \dots, 16_i, \dots, 16_n$  へのデータ書き込みを管理するために使用するモジュール 24 であって、リンク 22 から受信されるデータによりアクティブ化される、モジュール 24、

- データがバッファメモリ  $16_1, \dots, 16_i, \dots, 16_n$  に送信され、それらデータが書き込み・読み取りアクセスコマンドモジュール 18 を通してそれらメモリの基本ロケーション内に書き込まれるリンク 26、

- データがバッファメモリ  $16_1, \dots, 16_i, \dots, 16_n$  から受信され、書き込み・読み取りアクセスコマンドモジュール 18 を通してそれらメモリの基本ロケーションのそれらデータが読み取られるリンク 28、

- コントローラ 10 又は 20 が、リンク 28 により供給されたデータを受信可能なバッファメモリ  $16_1, \dots, 16_i, \dots, 16_n$  内のデータの読み取りを管理するために使用するモジュール 30、及び

- 読み取られたデータがアドレス 14 に送信されるリンク 32 を備える。

#### 【0029】

これら要素を使用して、DMA コントローラ 10 又は 20 は、ソース  $12_i$  から受信された各データに対応するバッファメモリ  $16_i$  に送信し、アドレス 14 に送信される各データは、DMA コントローラ 10 又は 20 により対応するバッファメモリ  $16_i$  において読み取られる。したがって、バッファメモリ  $16_1, \dots, 16_i, \dots, 16_n$  は、ソース  $12_1, \dots, 12_i, \dots, 12_n$  とアドレス 14 との間の中間バッファ機能を遂行する。

#### 【0030】

示されていない本発明の別の可能な実施形態では、DMA コントローラ 10 がバッファメモリ  $16_1, \dots, 16_i, \dots, 16_n$  を統合せず、それらバッファメモリ内にデータ書き込み管理モジュール 24 を有さない場合、換言すれば、それらバッファメモリに書き込みアクセスすることができない場合、例えば、ソース  $12_1, \dots, 12_i, \dots, 12_n$  がそれらバッファメモリを直接形成することに留意する。

#### 【0031】

バッファメモリ  $16_1, \dots, 16_i, \dots, 16_n$  は、それ自体既知の異なる様式で実施してもよいが、好ましい一実施形態では、循環バッファメモリの形態である。

#### 【0032】

その場合、各バッファメモリ  $16_i$  は、最下位アドレスと最上位アドレスとの間のメモリ領域を占める。バッファメモリに関連付けられた読み取りポインタ又は書き込みポインタは、このバッファメモリの最下位アドレスと最上位アドレスとの間でインクリメントされ、最上位アドレスに達した後、インクリメントの追加により最下位アドレスに戻る。これは、バッファメモリを「循環バッファメモリ」として適格なものにする。

#### 【0033】

例えば、FIFO (先入れ先出し) 式のこの循環バッファメモリ構成の利点は、このメモリに関連付けられた書き込みポインタが、記憶されてはいるがまだ読み取られていないデータを含む基本ロケーションに達しない限り、換言すれば、メモリが満杯ではない限り、そのようなメモリバッファ内にデータをまるで無期限に書き込み可能であるかのようにすべてのことが行われるため、追加のデータが受信された場合に、メモリ再割り振り制約が回避されることである。

#### 【0034】

この場合、データ受信リンク 22 は、任意的に、例えば、対応する循環バッファメモリ  $16_i$  が、満杯であるために書き込み阻止されている間、ソース  $12_i$  から受信されたデータを一時的に記憶可能な FIFO 式入力バッファメモリ 34 を備える。

#### 【0035】

10

20

30

40

50

これもまた任意的に、読み取られたデータを送信するリンク 32 は、例えば、出力ネットワークが一時的に飽和している間、又はアドレス 14 が読み取り阻止されている間、読み取られたデータを少なくとも 1 つの循環バッファメモリ  $16_1$ 、 $\dots$ 、 $16_i$ 、 $\dots$ 、 $16_n$  に一時的に記憶可能な F I F O 式出力バッファメモリ 36 を備える。

【0036】

DMA コントローラ 10 又は 20 は、アドレス 14 への入力データフローをバッファメモリに分散できるようにする、循環バッファメモリ記述子  $16_1$ 、 $\dots$ 、 $16_i$ 、 $\dots$ 、 $16_n$  を記憶する手段 38 も備える。

【0037】

循環バッファメモリ  $16_i$  毎に、記述子は、記憶媒体内の循環メモリ  $16_i$  の最下位アドレスを指す少なくとも 1 つの最下位ポインタ  $P B_i$ 、記憶媒体内の循環バッファメモリ  $16_i$  の最上位アドレスを指す最上位ポインタ  $P S_i$ 、DMA コントローラ 10 又は 20 が受信した次のデータを書き込むことができる循環バッファメモリ  $16_i$  内の基本ロケーションのアドレスを指す書き込みポインタ  $P E_i$ 、アドレス 14 に送信するために次のデータを読み取ることができる循環バッファメモリ  $16_i$  内の基本ロケーションのアドレスを指す読み取りポインタ  $P L_i$  を含む。

【0038】

各バッファメモリ  $16_i$  は、サイズがポインタ  $P B_i$  及び  $P S_i$  により示される最下位アドレスと最上位アドレスとの間に含まれる基本ロケーション数により定義される循環メモリであり、書き込みポインタ  $P E_i$  及び読み取りポインタ  $P L_i$  が指すことができるこのメモリの各基本ロケーションは、最下位アドレスと最上位アドレスとの間のアドレスにより識別される。

【0039】

最下位ポインタ  $P B_i$  及び最上位ポインタ  $P S_i$  は事前に決定され、一定のままであるのに対して、書き込みポインタ  $P E_i$  及び読み取りポインタ  $P L_i$  は可変である。ポインタは、循環バッファメモリ  $16_i$  内で、書き込み動作又は読み取り動作後にアドレスジャンプありで、又はアドレスジャンプなしでインクリメントすることにより更新することができる。

【0040】

上述したように、各バッファメモリ  $16_i$  の循環は、書き込みポインタ  $P E_i$  又は読み取りポインタ  $P L_i$  がメモリの最上位アドレスに達した場合、このポインタを基本ロケーション 1 つ分だけインクリメントさせると、メモリの最下位アドレスを指すことになることに依存する。

【0041】

したがって、アドレス 14 に送信するために、DMA コントローラ 10 又は 20 による読み取りを待っている間、各循環バッファメモリ  $16_i$  に一時的に記憶されるデータは、このメモリ  $16_i$  の読み取りポインタアドレスと最下位から最上位の方向での書き込みポインタ  $P E_i$  の先行アドレスとの間に配置される。他方、循環バッファメモリ  $16_i$  の書き込みポインタ  $P E_i$  のアドレスと最下位から最上位への方向での読み取りポインタ  $P L_i$  のアドレスの先行アドレスとの間に配置されるメモリ領域は、自由に書き込むことができる領域である。 $P B_i$ 、 $P S_i$ 、 $P E_i$ 、及び  $P L_i$  の各記述子に関連する各循環バッファ  $16_i$  のこの構造を、詳細に後述する図 3 A 及び図 3 B に示す。

【0042】

例えば、より正確には、各書き込みポインタ  $P E_i$  は、対応する循環バッファメモリ  $16_i$  での書き込み動作後に、最下位から最上位への方向に基本ロケーションをインクリメントすることにより更新することができる。

【0043】

他方、読み取りポインタ  $P L_1$ 、 $\dots$ 、 $P L_i$ 、 $\dots$ 、 $P L_n$  は、ファームウェア  $40_1$ 、 $\dots$ 、 $40_i$ 、 $\dots$ 、 $40_n$  のそれぞれの実行により更新される。例えば、これらファームウェア  $40_1$ 、 $\dots$ 、 $40_i$ 、 $\dots$ 、 $40_n$  は、循環バッファメモリ

10

20

30

40

50

16<sub>1</sub>、・・・、16<sub>i</sub>、・・・、16<sub>n</sub>と同じ記憶媒体に記憶され、各循環バッファメモリに関連付けられる。ファームウェアは、読み取り管理モジュール30により実行される。

【0044】

本発明によれば、各ファームウェア40<sub>i</sub>は、データを読み取り、関連付けられた循環バッファメモリ16<sub>i</sub>の読み取りポインタPL<sub>i</sub>を更新するようにプログラムされる。また本発明によれば、DMAコントローラ10又は20には、アドレス14に転送されるデータシーケンス内で要求される、ソース12<sub>1</sub>、・・・、12<sub>i</sub>、・・・、12<sub>n</sub>から導出されたデータの所定の順序に応じて、ファームウェア40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>の実行を同期させる手段が提供される。

10

【0045】

より正確には、同期手段は、例えば、少なくとも1つの読み取り・更新権を有するファームウェア40<sub>i</sub>から他のファームウェアに送信する手段を備えることができ、現在実行中のファームウェア40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>の1つが読み取り・更新権を一時的に保持することにより、そのファームウェアは、関連付けられた循環バッファメモリ16<sub>i</sub>内で少なくとも1つのデータ読み取り及び対応する読み取りポインタPL<sub>i</sub>の少なくとも1つの更新を実行することができる。

【0046】

単純な一実施形態では、この読み取り・更新権は一意であり、トークンと呼ばれ、図2ではTと記される。2つのオブジェクトが同時に実行されることを回避するためのオブジェクトからオブジェクトに循環する単一のトークンという概念は、コンピュータ科学において周知であり、詳細に説明しない。これは、特に基本のトークン待ち・送信関数(elementary waiting and token function)を使用して、プログラミングによりシミュレートされる。

20

【0047】

この場合、現在実行中の異なるファームウェア40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>間のトークンTの循環は、以下のように各ファームウェアにプログラムすることができる。

- 各ファームウェア40<sub>i</sub>が、トークンTを待つ少なくとも1つの命令を含み、
- 各ファームウェア40<sub>i</sub>が、循環バッファメモリ16<sub>i</sub>内で少なくとも1回の読み取り及び読み取りポインタPL<sub>i</sub>の少なくとも1つの更新が実行された後、トークンTを現在実行中の別ファームウェア40<sub>j</sub>に送信する少なくとも1つの命令を含む。

30

【0048】

したがって、同期手段は、読み取り管理モジュール30と、モジュール30が実行できるファームウェア40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>に統合されたトークンを待ち送信するこれら命令との組み合わせから生まれる。

【0049】

しかし、異なるファームウェア間で循環する読み取り・更新権を実施する同期プリミティブとして単一のトークンTを使用することは、ファームウェアプログラム(ひいてはソース)の数が増大し、出力データのインタレースがより複雑になった場合、確實ではない。このトークンを使用して読み取りがどのように同期させるかを不注意にプログラミングすると、異なるファームウェア間でのブロック又は実行衝突が生じる恐れがある。

40

【0050】

これは、セマフォ、特に2つのファームウェアプログラムに1つのセマフォを有するミューテックス(相互排除)セマフォとして適格なバイナリセマフォを使用して読み取り・更新権を実装することにより補正することができる。したがって、C<sup>2</sup><sub>N</sub>セマフォが、N個のソースを有するDMAコントローラに定義される。すべてのファームウェアの衝突の管理は、各セマフォが特に、識別された2つのファームウェアプログラム間で交換されるため、単純化される。明らかに、トークン及びバイナリセマフォの概念は、2つのソースを有するDMAコントローラに当てはまる。

50

## 【0051】

単純であるが限定的ではない特定の適用例での図1の直接メモリアクセスコントローラ10又は20の有利な動作を図2に示す。この適用例は、関連付けられた読み取りポインタのデータ読み取り・更新ファームウェア40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>を同期させる原理を単に実証するためだけに選択されたものである。この適用例によれば、第1のデータソース12<sub>1</sub>が、データシーケンスd<sub>1,1</sub>、d<sub>1,2</sub>、d<sub>2,1</sub>、d<sub>2,2</sub>、d<sub>3,1</sub>、d<sub>3,2</sub>、・・・を提供し、DMAコントローラ10又は20に提供し、DMAコントローラ10又は20は、それらデータを循環バッファメモリ16<sub>1</sub>に一時的に記憶する。書き込みポインタPE1のアドレス更新によるアドレスのインクリメントにより、それらソース12<sub>1</sub>により提供されるシーケンスの順序で循環バッファメモリ16<sub>1</sub>に記憶される。第2のデータソース12<sub>2</sub>は、データシーケンスd<sub>1,1</sub>、d<sub>1,2</sub>、d<sub>2,1</sub>、d<sub>2,2</sub>、d<sub>3,1</sub>、d<sub>3,2</sub>、・・・を提供し、DMAコントローラ10又は20に提供し、DMAコントローラ10又は20は、それらデータを循環バッファメモリ16<sub>2</sub>に一時的に記憶する。同様に、書き込みポインタPE2のアドレス更新によるアドレスのインクリメントにより、それらデータは、ソース12<sub>2</sub>により供給されたシーケンスの順序で循環バッファメモリ16<sub>2</sub>に記憶される。次に、DMAコントローラは、2つのソース12<sub>1</sub>及び12<sub>2</sub>から導出されたデータを、2つのシーケンスをインタレースさせて1つのシーケンスd<sub>1,1</sub>、d<sub>1,2</sub>、d<sub>2,1</sub>、d<sub>2,2</sub>、d<sub>3,1</sub>、d<sub>3,2</sub>、・・・を提供して、アドレス14に送信しなければならない。

10

## 【0052】

この出力シーケンスが提供される前、データ送信でのソース12<sub>1</sub>又は12<sub>2</sub>のうちの一方のその他に対する遅延に関係なく、バッファメモリ16<sub>1</sub>及びバッファメモリ16<sub>2</sub>のそれぞれに関連付けられたファームウェア40<sub>1</sub>及び40<sub>2</sub>は、以下のように定義される。

20

ファームウェア40<sub>1</sub>

```
For i in 0 to n (メモリ161において読み取るべきデータ数)
  PL1 += k ; (所定のインクリメントを使用してポインタPL1を更新する)
  Dest = *PL1 ; (データを読み取り、アドレス14に送信する)
  Send T to 402 ; (トークン/ミュテックスをファームウェア402
に送信する)
```

```
  Wait T ; (トークン/ミュテックスを待つ)
```

30

```
End for
```

ファームウェア40<sub>2</sub>

```
For i in 0 to n (メモリ162において読み取るべきデータ数)
  PL2 += k ; (所定のインクリメントを使用してポインタPL2を更新する)
  Dest = *PL2 ; (データを読み取り、アドレス14に送信する)
  Send T to 401 ; (トークン/ミュテックスをファームウェア401
に送信する)
```

```
  Wait T ; (トークン/ミュテックスを待つ)
```

```
End for
```

## 【0053】

インクリメントkは1に等しくてよいが、ソース12<sub>1</sub>及び12<sub>2</sub>のいずれかにより提供されるすべてのデータのすべてを送信する必要があるわけではない場合、1よりも大きくてもよい。

40

## 【0054】

したがって、ソースが互いに同期されない場合であっても、データが読み取られて出力シーケンスd<sub>1,1</sub>、d<sub>1,2</sub>、d<sub>2,1</sub>、d<sub>2,2</sub>、d<sub>3,1</sub>、d<sub>3,2</sub>、・・・を提供する都度、トークン又はミュテックスTは、ファームウェア40<sub>1</sub>と40<sub>2</sub>との間で循環する。さらに、要求される出力シーケンスのこの決定論的な供給は、DMAコントローラ10又は20が任意のマイクロプロセッサと同期されずに行われる。

## 【0055】

50

データインタレースはCDMA（符号分割多元接続）技術を使用して、又はパイロットデータを無視しながら有用データを読み取ることで、2つのMIMO（多入力多出力）データフレームを組み合わせることにより行われるため、DMAコントローラを使用するファームウェア40<sub>1</sub>、・・・、40<sub>i</sub>、・・・、40<sub>n</sub>のこの同期の多くの実際用途がある。

【0056】

したがって、DMAコントローラが、データソースがその他のソースから独立してデータを送信する速度に関係なく、アドレス14にデータをデータソースの所定の順序で単純且つ効率的に提供可能なことは極めて明白である。

【0057】

いくつかのソースから導出されたデータがアドレスに提供される順序を管理するこの機能と組み合わせて、本発明の特定の一実施形態では、DMAコントローラが入力フロー同期機能及び出力フロー同期機能も実行し得ることが可能である。

【0058】

この特定の実施形態の実施を可能にするにはまず、DMAコントローラは、上記定義した書き込み管理モジュール24を備えなければならない。

【0059】

この場合、データ書き込み管理モジュール24は、特に書き込みポインタPE<sub>i</sub>及び読み取りポインタPL<sub>i</sub>の相対値に依存して、任意の1つの循環バッファメモリ16<sub>i</sub>へのデータの書き込みを許可又は阻止するように設計される。データ書き込み管理モジュール24は、書き込みポインタPE<sub>i</sub>を基本ロケーション1つだけインクリメントすることにより、シーケンサを使用して書き込みポインタPE<sub>i</sub>を自動的に更新するようにも設計される。より正確には、本発明のこの特定の実施形態の単純な一変形では、循環バッファメモリ16<sub>i</sub>内に書き込むべきデータは、書き込みポインタPE<sub>i</sub>により示されるアドレスに書き込まなければならない、このデータの書き込みは、書き込み管理モジュール24が以前に行った書き込みポインタPE<sub>i</sub>への更新が、書き込みポインタPE<sub>i</sub>が指すアドレスが、読み取りポインタPL<sub>i</sub>が指すアドレスに達するか、又はそれを超えるようなものである（メモリが満杯である）場合、阻止されなければならない。

【0060】

同様に、データ読み取り管理モジュール30は、特に書き込みポインタPE<sub>i</sub>及び読み取りポインタPL<sub>i</sub>の相対値に応じて、循環バッファメモリ16<sub>i</sub>内のデータの読み取りを許可又は阻止するように設計される。データ読み取り管理モジュール30は、各ファームウェア40<sub>i</sub>を実行することにより、各読み取りポインタPL<sub>i</sub>を更新するようにも設計される。より正確には、本発明のこの特定の実施形態の単純な一変形では、循環バッファメモリ16<sub>i</sub>内で読み取られるべきデータは、読み取りポインタPL<sub>i</sub>により示されるアドレスで読み取られなければならない、読み取り管理モジュール30により行われた読み取りポインタPL<sub>i</sub>への更新が、読み取りポインタPL<sub>i</sub>が指すアドレスが、書き込みポインタPE<sub>i</sub>が指すアドレスに達するか、又はそれを超えるようなものである（メモリが空である）場合、データの読み取りは、阻止されなければならない。

【0061】

特に、任意の循環バッファメモリ16<sub>i</sub>での読み出し動作及び書き込み動作の阻止及び非阻止が上述したように機能可能なように、書き込み管理モジュール24は、書き込みポインタPE<sub>i</sub>の各更新後に、

- 書き込みポインタPE<sub>i</sub>が読み取りポインタPL<sub>i</sub>に達した場合（メモリが満杯であることを示す）、循環バッファメモリ16<sub>i</sub>へのすべてのデータの書き込みを阻止すること、及び

- 以前、この読み取りが阻止され、書き込みポインタPE<sub>i</sub>が読み取りポインタPL<sub>i</sub>を再び超える場合、循環バッファメモリ16<sub>i</sub>内のデータ読み取りを阻止しないことを行うようにプログラムすることができる。

【0062】

10

20

30

40

50

同様に、読み取り管理モジュール30は、読み取りポインタ $PL_i$ の各更新後に、

- 読み取りポインタ $PL_i$ が書き込みポインタ $PE_i$ に達した場合（メモリが空であることを示す）、循環バッファメモリ $16_i$ 内のすべてのデータ読み取りを阻止すること、及び

- 以前、この書き込みが阻止され、読み取りポインタ $PL_i$ が読み取りポインタ $PE_i$ を再び超えた場合、循環バッファメモリ $16_i$ へのデータの書き込みを阻止しないことを行うようにプログラムすることができる。

#### 【0063】

このようにして、書き込み管理モジュール24は、メモリ $16_i$ が満杯の場合、書き込み阻止機能を実行し、メモリ $16_i$ が空の場合、読み取り非阻止機能を実行し、再び満杯にする。読み取り管理モジュール30は、メモリ $16_i$ が空の場合、読み取り阻止機能を実行し、満杯のメモリ $16_i$ が再び空になった場合、書き込み非阻止機能を実行する。

10

#### 【0064】

上述した書き込み管理モジュール24及び読み取り管理モジュール30の詳細な動作は、書き込みポインタ $PE_1$ 、 $\dots$ 、 $PE_i$ 、 $\dots$ 、 $PE_n$ 及び読み取りポインタ $PL_1$ 、 $\dots$ 、 $PL_i$ 、 $\dots$ 、 $PL_n$ を使用して循環バッファメモリ $16_1$ 、 $\dots$ 、 $16_i$ 、 $\dots$ 、 $16_n$ 内の書き込み及び読み取り阻止手段の単なる一例である。しかし、当業者は、書き込みポインタ $PE_1$ 、 $\dots$ 、 $PE_i$ 、 $\dots$ 、 $PE_n$ 及び読み取りポインタ $PL_1$ 、 $\dots$ 、 $PL_i$ 、 $\dots$ 、 $PL_n$ の値に応じて、循環バッファメモリ $16_1$ 、 $\dots$ 、 $16_i$ 、 $\dots$ 、 $16_n$ に読み書きを阻止させる他の実施態様又は変形も考案可能なことを理解するであろう。

20

#### 【0065】

これら特徴により、DMAコントローラ10又は20は、ソース $12_1$ 、 $\dots$ 、 $12_i$ 、 $\dots$ 、 $12_n$ からの入力データフロー及びアドレス14への出力フローを同期することができる。

#### 【0066】

結果として、例えば、1つ又は複数のマイクロプロセッサにより処理又は使用されたデータの一時的なローカル記憶領域であり得るいくつかのデータソース $12_1$ 、 $\dots$ 、 $12_i$ 、 $\dots$ 、 $12_n$ と、例えば、周辺機器又は任意の機能オペレータであり得るアドレス14と、変形実施形態20に準拠し、循環バッファメモリ $16_1$ 、 $\dots$ 、 $16_i$ 、 $\dots$ 、 $16_n$ を含むDMAコントローラとを含む集積回路において、マイクロプロセッサは、それらソースとアドレス14との間でDMAコントローラにより行われる転送から独立して、データをソース $12_1$ 、 $\dots$ 、 $12_i$ 、 $\dots$ 、 $12_n$ に転送することができる。

30

#### 【0067】

このDMAコントローラ同期機能と併せて、本発明の別の実施形態では、第1の変形によれば、各循環バッファメモリ $16_i$ に「作業ポインタ」と呼ばれる新しい読み取りポインタを定義することが可能である。この作業ポインタ $PT_i$ は、前に定義した読み取りポインタ $PL_i$ と異なる。読み取りポインタのように、作業ポインタは記憶手段38に記憶される。作業ポインタは変数であり、実行される各読み取り動作後にファームウェア $40_i$ により更新される。

40

#### 【0068】

2つのソース $12_1$ 及び $12_2$ を有する単純な実施形態を示す図3A及び図3Bでは、2つの作業ポインタ $PT_1$ 及び $PT_2$ が、書き込みポインタ $PE_1$ 、 $PE_2$ 及び読み取りポインタ $PL_1$ 及び $PL_2$ に追加される。より正確には、図3Aは循環バッファメモリ構造 $16_1$ を表し、図3Bも同様に、循環バッファメモリ $16_2$ を示す。

#### 【0069】

したがって、各ファームウェア $40_i$ は、作業ポインタ $PT_i$ を更新する論理が、循環バッファメモリ $16_i$ において少なくとも1つのアドレスジャンプを含み、それにより、生成したシーケンスとは異なる順序でソース $12_i$ から導出されたデータを読み取ること

50

が可能なように定義することができる。この新しいポインタ  $P T_i$  及び図 3 A 及び図 3 B に示すように、各循環バッファ  $16_i$  内の順方向又は逆方向へのアドレスジャンプによる更新を行う可能性を導入することにより、このバッファメモリは、単純な FIFO 式リストのようには使用されない。作業ポインタ  $P T_1$ 、 $\dots$ 、 $P T_i$ 、 $\dots$ 、 $P T_n$  により、DMA コントローラ  $10$  又は  $20$  は、ソース  $12_1$ 、 $\dots$ 、 $12_i$ 、 $\dots$ 、 $12_n$  自体から受信された各データシーケンスを並び替えることができる。

【0070】

この第 1 の変形では、任意の 1 つの循環バッファ  $16_i$  の読み取りポインタ  $P L_i$  は、データを書き込むためにメモリ領域を解放する読み取りポインタである。換言すれば、この読み取りポインタが更新される都度、前に指されていたアドレスのロケーションと、読み取りポインタ  $P L_i$  により指される新たなアドレスのロケーションの先行ロケーションとの間のメモリ領域は、書き込みのために自動的に解放される。これは、上述したように、書き込みポインタ  $P E_i$  のアドレスと、最下位から最上位への方向での読み取りポインタ  $P L_i$  のアドレスに先行するアドレスとの間の循環バッファメモリ  $16_i$  内に配置されるメモリ領域が、原理上、自由に書き込むことができる領域としてみなされることによる。

10

【0071】

任意の 1 つの循環バッファメモリ  $16_i$  の作業ポインタ  $P T_i$  はそれ自体、読み取られたデータを保持する読み取りポインタである。換言すれば、データをアドレス  $14$  に送信できるように、データを循環バッファメモリ  $16_i$  において読み取る必要がある場合、データは、作業ポインタ  $P T_i$  により示されるアドレスから読み取られる。しかし、この読み取り後の作業ポインタ  $P T_i$  の更新は、書き込みのために対応する基本ロケーションを解放しない。読み取りでの作業ポインタ  $P T_i$  の変位は、アドレスの昇順でソース  $12_i$  から発せられたデータシーケンスに従う必要はない。変位は、任意の順序で、好ましくは読み取りポインタ  $P L_i$  と書き込みポインタ  $P E_i$  との間でファームウェア  $40_i$  により自由に事前に決定される。特に、用途に応じて、作業ポインタ  $P T_i$  により最初に読み取られたデータを再び読み取ることができ、データを、循環バッファメモリ  $16_i$  に記憶された順序とは異なる順序で読み取ることができる。

20

【0072】

したがって、各作業ポインタ  $P T_i$  が読み取りポインタであるこの変形実施形態では、各ファームウェア  $40_i$  が、読み取り管理モジュール  $30$  により実行されて、循環バッファメモリ  $16_i$  に関連付けられた 2 つの読み取りポインタを更新する命令であって、

30

- 作業ポインタ  $P T_i$  の目標とする用途に応じて、順方向及び / 又は逆方向のアドレスジャンプを含み得る命令、

- 作業ポインタ  $P T_i$  にプログラムされた変位に応じて、読み取りポインタ  $P L_i$  のみに順方向アドレスジャンプを含み得る命令

を含むことが分かる。

【0073】

他方、ファームウェア  $40_1$ 、 $\dots$ 、 $40_i$ 、 $\dots$ 、 $40_n$  のいずれも書き込みポインタ  $P E_1$ 、 $\dots$ 、 $P E_i$ 、 $\dots$ 、 $P E_n$  を更新する命令を含まず、書き込みポインタ  $P E_1$ 、 $\dots$ 、 $P E_i$ 、 $\dots$ 、 $P E_n$  は、例えば、循環バッファメモリ  $16_1$ 、 $\dots$ 、 $16_i$ 、 $\dots$ 、 $16_n$  のうちの 1 つでの各データ書き込み動作においてアドレスジャンプなしでインクリメントされるようにシーケンサにより更新され得る。

40

【0074】

任意の作業ポインタ  $P T_i$  の変位が、図 3 A に示すように、ポインタが読み取るべきデータを実際に含む循環バッファメモリゾーン  $16_i$  を指すようにプログラムされなければならないことにも留意する。作業ポインタのうちの一つが、更新後 (図 3 B に示す) にこのゾーンの外部を指す場合、目標とする用途に応じて、エラーメッセージを生成することができるか、又は書き込みポインタ  $P E_i$  が追いつくまでこの位置に凍結させることができる。

50

## 【 0 0 7 5 】

上記と対称をなす一変形実施形態では、データシーケンスを並べ替えるために、各循環バッファメモリ  $16_i$  の書き込みポインタとして新しい作業ポインタ  $PT_i$  を定義することも可能であろう。

## 【 0 0 7 6 】

この第2の変形では、各循環バッファメモリ  $16_i$  の書き込みポインタ  $PE_i$  は、循環バッファメモリ  $16_i$  内にデータを書き込むメモリ領域が確保された書き込みポインタである。換言すれば、書き込みポインタが更新される都度、前に指していたアドレスのロケーションと書き込みポインタ  $PE_i$  により新たに指されるアドレスの先行ロケーションとの間のメモリ領域は、自動的に読み取りに確保される。作業ポインタ  $PT_i$  は、任意の所定の順序で循環バッファメモリ  $16_i$  内のポインタ  $PE_i$  により確保されるメモリ領域内にデータを書き込む効果的な書き込みポインタである。

10

## 【 0 0 7 7 】

しかし、この第2の変形は、循環バッファメモリ  $16_1$ 、 $\dots$ 、 $16_i$ 、 $\dots$ 、 $16_n$  内の書き込み末端に閉塞を生成される恐れがあるため、前の変形よりも魅力が劣る。DMAコントローラによるデータを転送する用途では、書き込み末端の上流よりも循環バッファメモリ内の読み取り末端の下流において閉塞を何とかすることが好ましい。

## 【 0 0 7 8 】

これより、DMAコントローラ管理ブロックが循環バッファメモリ  $16_1$  及び  $16_2$  に読み書きし、作業ポインタ  $PT_1$  及び  $PT_2$  が、各循環バッファメモリで読み出されたデータを保持する読み取りポインタとして導入される上述した第1の変形実施形態を参照して、2つのソース  $12_1$  及び  $12_2$  から導出されたデータを転送する方法であって、転送されるデータはインタレースされ、DMAコントローラ  $10$  又は  $20$  はそのデータを使用可能である、方法の例の詳細を説明する。この方法の連続ステップを図4に示す。

20

## 【 0 0 7 9 】

この転送方法は、3つ以上のソースに一般化するすることができ、この特定の例は、データインタレース用途のみならず、単純な説明及び例を提供するために、図4を参照して与えられる。

## 【 0 0 8 0 】

第1のステップ  $100$  において、コントローラ  $10$  又は  $20$  は、循環バッファメモリ  $16_i$  (この例では、 $i = 1$  又は  $2$ ) のうちの1つへのデータ読み取り又は書き込み命令を待つ。

30

## 【 0 0 8 1 】

このステップ  $100$  の後に、ステップ  $102$  が続き、循環バッファメモリ  $16_i$  のうちの1つでのデータ書き込み命令を受けて、書き込みを開始する。書き込み管理モジュール  $24$  は、この書き込み開始ステップ中にアクティブ化される。非限定的な実施形態例として、書き込み管理モジュール  $24$  は、記憶手段  $38$  から、このバッファメモリへの書き込みが阻止されるか否かを示す循環バッファメモリ  $16_i$  の第1の記述子の値を抽出する。

## 【 0 0 8 2 】

次に、この第1の記述子の値に応じて、テストステップ  $104$  が行われて、循環バッファメモリ  $16_i$  内で考慮されるデータの書き込みを続けるか否かが判断される。第1の記述子により、書き込みが阻止されることが示される場合、ステップ  $100$  が繰り返され、その他の場合、ステップ  $106$  が使用される。

40

## 【 0 0 8 3 】

ステップ  $106$  において、書き込み管理モジュール  $24$  は、書き込みポインタ  $PE_i$  が指すアドレスを記憶手段  $38$  から抽出する。次に、書き込み管理モジュール  $24$  は、考慮されるデータを循環バッファメモリ  $16_i$  の対応する基本ロケーションに書き込む。

## 【 0 0 8 4 】

次に、更新ステップ  $108$  において、書き込みポインタ  $PE_i$  が指すアドレスが、循環バッファメモリ  $16_i$  内で基本ロケーション1つ分だけインクリメントされる。

50

## 【 0 0 8 5 】

最後に、ステップ 1 1 0 において、書き込み管理モジュール 2 4 は、読み取りポインタ P L i が指すアドレスを記憶手段 3 8 から抽出し、次に、可能な場合には、以下の状況において、第 1 の書き込み阻止記述子の値及び循環バッファメモリ 1 6<sub>i</sub> の第 2 の読み取り阻止記述子の値を更新する。

- ステップ 1 0 8 での更新後、書き込みポインタ P E i が指すアドレスが、読み取りポインタ P L i が指すアドレスに達する場合（メモリ 1 6<sub>i</sub> が満杯であることを示す）、第 1 の記述子は、書き込みが循環バッファメモリ 1 6<sub>i</sub> において阻止されることを示すように変更される。

- ステップ 1 0 8 での更新後、書き込みポインタ P E i が指すアドレスが、読み取りポインタ P L i が指すアドレスと異なり、第 2 の記述子が、読み取りが阻止されることを示す場合、第 2 の記述子は、読み取りが循環バッファメモリ 1 6<sub>i</sub> において許可されることを示すように変更される。

10

## 【 0 0 8 6 】

ステップ 1 1 0 の後、ステップ 1 0 0 が繰り返される。

## 【 0 0 8 7 】

ステップ 1 0 0 の後には、ステップ 1 1 2 も続き、ファームウェア 4 0<sub>i</sub> がアクティブ化された後、循環バッファメモリ 1 6<sub>i</sub> でのデータシーケンスの読み取りが開始される。このアクティブ化ステップは、読み取り管理モジュール 3 0 により実行される。非限定的な実施形態例として、読み取り管理モジュール 3 0 は、記憶手段 3 8 から、読み取りがこのバッファメモリにおいて阻止されるか否かを示す循環バッファメモリ 1 6<sub>i</sub> の第 2 の記述子の値を抽出する。

20

## 【 0 0 8 8 】

次に、テストステップ 1 1 4 において、読み取り管理モジュール 3 0 は、アドレス 1 4 がデータを受信可能な状態にあるか否かを判断する。このテストは、いくつかの方法で行うことができる。DMA コントローラ 1 0 又は 2 0 が出力バッファ 3 6 を備える場合、出力バッファ 3 6 が満杯ではない限り、データを転送できると考えることができる。デフォルトにより、又は補足として、従来の通信プロトコルにより、アドレス 1 4 は DMA コントローラ 1 0 又は 2 0 にアドレス 1 4 の所定の形式でのメッセージによるデータ受信能力について通知することが可能であり得る。データをアドレス 1 4 に転送することが不可能な場合、ファームウェア 4 0<sub>i</sub> の実行は、ステップ 1 1 4 において阻止されたままである。次のステップ 1 1 6 は、出力バッファ 3 6 が空になりつつあるため、又は DMA コントローラが、アドレスがデータを受信可能なことを通知するメッセージを受信したために、アドレスが再びデータを受信可能な状態になるまで開始されない。

30

## 【 0 0 8 9 】

ステップ 1 1 6 は、別のテストステップである。このステップ中、第 2 の記述子の値に応じて、循環バッファメモリ 1 6<sub>i</sub> でのデータ読み取りを続けるか否かが判断される。このメモリ 1 6<sub>i</sub> の第 2 の記述子が、読み取りが阻止されることを示す場合、ファームウェア 4 0<sub>i</sub> の実行は、ステップ 1 1 6 において阻止されたままであり、その他の場合、次のステップ 1 1 8 が行われる。循環バッファメモリ 1 6<sub>i</sub> 内に書き込まれたデータが第 2 の記述子の値を変更しない限り（ステップ 1 1 0 ）、シーケンスはステップ 1 1 6 において阻止状態を保つ。

40

## 【 0 0 9 0 】

ステップ 1 1 8 は、循環バッファメモリ 1 6<sub>i</sub> が読み取られ、作業ポインタ P T 1 が更新され、場合によっては、読み取りポインタ P L 1 がファームウェア 4 0<sub>i</sub> の命令に従って更新されるステップである。

## 【 0 0 9 1 】

次に、ステップ 1 2 0 中、読み取り管理モジュール 3 0 はおそらく、以下の状況において、循環バッファメモリ 1 6<sub>i</sub> での第 1 の書き込み阻止記述子の値及び第 2 の読み取り阻止記述子を更新する。

50

- 読み取りポインタ P L 1 が指すアドレスが、ステップ 1 1 8 での更新後に、書き込みポインタ P E 1 が指すアドレスに達する場合（メモリ 1 6<sub>1</sub> が空であることを示す）、第 2 の記述子は、循環バッファメモリ 1 6<sub>1</sub> において読み取りが阻止されることを示すように変更される。

- 読み取りポインタ P L 1 が指すアドレスが、ステップ 1 1 8 での更新後に、書き込みポインタ P E 1 が指すアドレスと異なり、第 1 の記述子が、書き込みが阻止されることを示す場合、第 1 の記述子は、循環バッファメモリ 1 6<sub>1</sub> での書き込み許可を示すように変更される。

【 0 0 9 2 】

ステップ 1 2 0 の後に、ステップ 1 2 2 が続き、トークン T（又はミューテックス）をファームウェア 4 0<sub>2</sub> に送信する。

10

【 0 0 9 3 】

ステップ 1 2 2 の後に、ステップ 1 2 4 が続き、トークン T を待つ。ファームウェア 4 0<sub>1</sub> は、現在実行中のトークンを別のファームウェア、実際にはファームウェア 4 0<sub>2</sub> から受信しない限り、このステップにおいて阻止状態を保つ。トークン T を受信するとすぐに、ソース 1 2<sub>1</sub> から発せられたデータシーケンスを読み取るカウンタがインクリメントされ、シーケンス全体が読み取られるまで、ステップ 1 1 4 が再び実行される。要求されるシーケンス全体が読み取られた場合、ファームウェア 4 0<sub>1</sub> の実行を終了する（F）。

【 0 0 9 4 】

ステップ 1 0 0 の後には、ステップ 2 1 2 も続き、ファームウェア 4 0<sub>2</sub> がアクティブ化された後、循環バッファメモリ 1 6<sub>2</sub> でのデータシーケンスの読み取りが始動する。このアクティブ化ステップは、読み取り管理モジュール 3 0 により実行される。非限定的な実施形態例として、読み取り管理モジュール 3 0 は、記憶手段 3 8 から、このバッファメモリにおいて読み取りが阻止されるか否かを示す循環バッファメモリ 1 6<sub>2</sub> の第 2 の記述子の値を抽出する。

20

【 0 0 9 5 】

ステップ 2 1 2 の後に、ステップ 2 1 4 が続き、トークン T を待つ。ファームウェア 4 0<sub>2</sub> は、このステップにおいて、現在実行中のトークンを別のファームウェア、実際にはファームウェア 4 0<sub>1</sub> から受信しない限り、このステップにおいて阻止状態を保つ。トークン T を受信するとすぐに、テストステップ 2 1 6 が開始される。

30

【 0 0 9 6 】

このテストステップ 2 1 6 中、読み取り管理モジュール 3 0 は、アドレス 1 4 がデータを受信可能な状態であるか否かを判断する。このステップは、上述したステップ 1 1 4 と同じである。ファームウェア 4 0<sub>2</sub> の実行は、データをアドレス 1 4 に転送することができない限り、このステップにおいて阻止状態を保つ。

【 0 0 9 7 】

次のステップ 2 1 8 は別のテストステップである。このステップ中、循環バッファメモリ 1 6<sub>2</sub> の第 2 の記述子の値に応じて、データ読み取りを続けるか否かが判断される。このメモリ 1 6<sub>2</sub> の第 2 の記述子が、読み取りが阻止されることを示す場合、ファームウェア 4 0<sub>2</sub> の実行はステップ 2 1 8 において阻止状態を保ち、その他の場合、次のステップ 2 2 0 が開始される。循環バッファメモリ 1 6<sub>2</sub> へのデータの書き込みが、第 2 の記述子の値を変更しない限り（ステップ 1 1 0）、シーケンスはステップ 2 1 8 において阻止状態を保つ。

40

【 0 0 9 8 】

ステップ 2 2 0 は、循環バッファメモリ 1 6<sub>2</sub> において読み取り、作業ポインタ P T 2 を更新し、場合によっては、ファームウェア 4 0<sub>2</sub> の命令に従って読み取りポインタ P L 2 を更新するステップである。

【 0 0 9 9 】

次に、ステップ 2 2 2 中、読み取り管理モジュール 3 0 は、以下の状況において、循環バッファメモリ 1 6<sub>2</sub> の第 1 の書き込み阻止記述子の値及び第 2 の読み取り阻止記述子の

50

値を更新し得る。

- 読み取りポインタ P L 2 が指すアドレスが、ステップ 2 2 0 での更新後、書き込みポインタ P E 2 が指すアドレスに達する場合（メモリ 1 6<sub>2</sub> が空であることを示す）、第 2 の記述子が、循環バッファメモリ 1 6<sub>2</sub> において読み取りが阻止されることを示すように変更される。

- 読み取りポインタ P L 2 が指すアドレスが、ステップ 2 2 0 での更新後、書き込みポインタ P E 2 が指すアドレスと異なり、第 1 の記述子が、書き込みが阻止されることを示す場合、第 1 の記述子は、循環バッファメモリ 1 6<sub>2</sub> において書き込みが許可されることを示すように変更される。

【 0 1 0 0 】

ステップ 2 2 2 の後に、ステップ 2 2 4 が続き、トークン T がファームウェア 4 0<sub>1</sub> に送信される。トークン T が送信されるとすぐに、ソース 1 2<sub>2</sub> から導出されたデータシーケンスの読み取りカウンタがインクリメントされ、シーケンス全体が読み取られるまで、実行は待つステップ 2 1 4 に戻る。要求されるシーケンス全体が読み取られた場合、ファームウェア 4 0<sub>2</sub> の実行は終了する（F）。

【 0 1 0 1 】

明らかなことに、循環バッファメモリでの読み取り及び同じ循環バッファメモリでの書き込みを行うファームウェアの実行は、DMA コントローラ 1 0 又は 2 0 により独立して並列に管理することができる。これは、読み取りファームウェア間でのトークン T（又はより一般的にはセマフォ）の循環により決定されるアドレスに送信されるデータシーケンスを何等変更しない。

【 0 1 0 2 】

したがって、上述した直接メモリアクセスコントローラ及びそのデータ転送動作により、いくつかのソースから少なくとも 1 つのアドレスへのデータ転送の管理が、DMA コントローラがアドレスにデータを送信する所定の順序に関して、容易になることがかなり明らかである。

【 0 1 0 3 】

本発明が考案された実施形態に限定されないことにも留意する。特に、上記では、説明を簡単にするために、転送が単一のアドレスに対して行われる適用例を使用することが考案されたが、本発明の原理は、いくつかのアドレスがある場合でも等しく適用可能である。

【 0 1 0 4 】

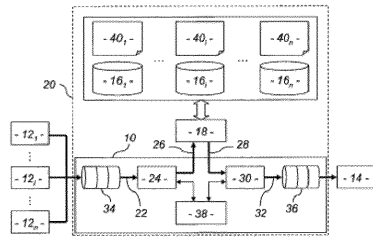
より一般的には、当業者は、上記に開示される情報を考慮した後、上述した実施形態に対していくつかの変更を行い得ることを認識するであろう。以下の特許請求の範囲内で使用される用語は、特許請求の範囲をこの説明に提示される実施形態に限定するものとして解釈されてはならず、当業者が一般的な知識を適用して、ここで開示された情報を利用することにより構築され、予測され得る、特許請求の範囲が包含を目的とするすべての均等物を含むものと解釈されなければならない。

10

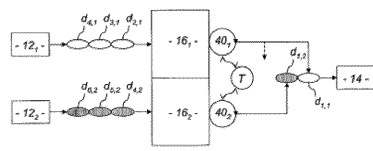
20

30

【 図 1 】



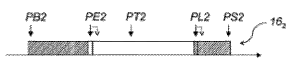
【 図 2 】



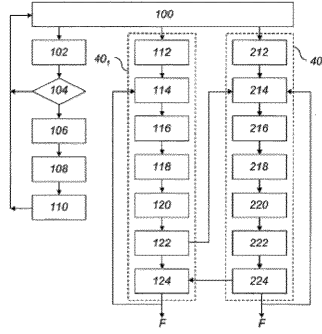
【 図 3 A 】



【 図 3 B 】



【 図 4 】



---

フロントページの続き

(72)発明者 クリスティアン ベルナル  
フランス 38960 サン - エティエンヌ ドゥ クロセ , シュマン ドゥ ラルニ 120

審査官 横山 佳弘

(56)参考文献 特開2001 - 034588 (JP, A)  
特開平05 - 289847 (JP, A)  
国際公開第2007 / 099659 (WO, A1)  
特開2000 - 305891 (JP, A)  
特開2003 - 087338 (JP, A)  
特開2006 - 238452 (JP, A)

(58)調査した分野(Int.Cl. , DB名)  
G06F 13 / 28  
G06F 13 / 38  
G06F 13 / 26  
G06F 5 / 06