(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3982077号
(P3982077)
(45)発行日　平成19年9月26日(2007.9.26)　(24)登録日　平成19年7月13日(2007.7.13)

(51)Int.Cl.　F I
G06F 15/80 (2006.01)　G06F 15/80

請求項の数 7　(全 12 頁)

(21)出願番号　特願平10-239314
(22)出願日　平成10年8月26日(1998.8.26)
(65)公開番号　特開2000-67008(P2000-67008A)
(43)公開日　平成12年3月3日(2000.3.3)
審査請求日　平成17年8月5日(2005.8.5)

(73)特許権者　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100092174
弁理士　平戸　哲夫
(72)発明者　中原　誠
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

審査官　鳥居　稔

最終頁に続く

(54)【発明の名称】マルチプロセッサシステム

(57)【特許請求の範囲】

【請求項１】

データ入力端子とデータ出力端子とを有し、データメモリに接続され、入力ポートを介して逐次入力される入力データの受け渡し先を判断し、各入力データを所定の受け渡し先に受け渡すようにプログラミングされる第１のプロセッサと、

データ入力端子とデータ出力端子とを有し、前記データメモリに接続され、前記第１のプロセッサから受け渡される入力データについて所定の処理を行うようにプログラミングされる第２～第ｎ（但し、ｎは３以上の整数）のプロセッサと、

データ入力端子とデータ出力端子とを有し、データ入力端子を前記第２～第ｎのプロセッサに接続される第ｎ＋１のプロセッサと、

前記第２～第ｎ＋１のプロセッサの出力データのいずれかを出力ポートに転送する出力データ転送手段とを備え、

前記第ｎ＋１のプロセッサは、前記第２～第ｎのプロセッサの出力データについて所定の処理を行い、その処理結果に基づいて、前記第２～第ｎ＋１のプロセッサの出力データのどれを前記出力ポートに転送するかを前記出力データ転送手段に指示するようにプログラミングされるものである

ことを特徴とするマルチプロセッサシステム。

【請求項２】

前記第１のプロセッサは、所定の入力データについては前記データメモリに受け渡すようにプログラミングされ、

前記第２～第ｎのプロセッサは、前記第１のプロセッサから受け渡された入力データについて所定の処理を行う場合、必要に応じて、前記データメモリに記憶された入力データを参照するようにプログラミングされるものである
ことを特徴とする請求項１記載のマルチプロセッサシステム。
【請求項３】
前記第１のプロセッサの前段に設けられ、前記入力ポートを介して逐次入力される入力データを格納する第１のＦＩＦＯ回路と、
前記第２～第ｎのプロセッサの前段に設けられ、前記第１のプロセッサから前記第２～第ｎのプロセッサに受け渡される入力データを格納する第２～第ｎのＦＩＦＯ回路と、
前記第２～第ｎのプロセッサの後段に設けられ、前記第２～第ｎのプロセッサの出力データを格納する第ｎ＋１～第２ｎ－１のＦＩＦＯ回路と、
前記第ｎ＋１のプロセッサの後段に設けられ、前記第ｎ＋１のプロセッサの出力データを格納する第２ｎのＦＩＦＯ回路とを備えている
ことを特徴とする請求項１又は２記載のマルチプロセッサシステム。
【請求項４】
前記第ｎ＋１～第２ｎ－１のＦＩＦＯ回路は、ライトポインタと、第１、第２のリードポインタとを有し、
前記ライトポインタは、前段のプロセッサに割り当てられ、前記第１のリードポインタは、前記第ｎ＋１のプロセッサに割り当てられ、前記第２のリードポインタは、前記出力データ転送手段に割り当てられている
ことを特徴とする請求項３記載のマルチプロセッサシステム。
【請求項５】
前記第１～第ｎ＋１のプロセッサは、前段のＦＩＦＯ回路のカウンタを更新する命令と、前段のＦＩＦＯ回路がエンプティのときは、後続する命令を停止する命令と、後段のＦＩＦＯ回路のカウンタを更新する命令と、後段のＦＩＦＯ回路がフルのときは、後続する命令の実行を停止する命令とを有している
ことを特徴とする請求項４記載のマルチプロセッサシステム。
【請求項６】
前記出力データ転送手段は、ＤＭＡコントローラである
ことを特徴とする請求項１～５のいずれか一項に記載のマルチプロセッサシステム。
【請求項７】
前記出力データ転送手段は、プロセッサである
ことを特徴とする請求項１～５のいずれか一項に記載のマルチプロセッサシステム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、複数のプロセッサを備えたコンピュータシステムであるマルチプロセッサシステムに関する。
【０００２】
コンピュータには常に高速・高性能が要求されており、それを実現するための手法として、命令の並列実行などの手法がとられているが、命令の並列実行の代表的な手法には、シングルプロセッサにおいて並列実行できる命令を増やす手法と、複数のプロセッサを使用して並列実行できる命令を増やす手法がある。
【０００３】
複数のプロセッサを備えるマルチプロセッサシステムは、スーパコンピュータや大型コンピュータなどで実現されている技術であるが、プロセッサ数を増やすことで性能を向上させることに主眼が置かれており、容易にプロセッサ数を増やせるシステム形態が主流となっている。
【０００４】
【従来の技術】

図６は従来のマルチプロセッサシステムの一例の要部を示すブロック回路図である。図６中、１－１、１－２、１－ｎはプロセッサ、２はバス、３は共有メモリであり、このマルチプロセッサシステムは、バス２に、複数のプロセッサ１－１～１－ｎと、これら複数のプロセッサ１－１～１－ｎに共有される共有メモリ３とを接続してなるマルチプロセッサシステムである。

【０００５】

図７は従来のマルチプロセッサシステムの他の例の要部を示すブロック回路図である。図７中、４－１、４－２、４－ｎはプロセッサ、５はクロスバネットワークであり、このマルチプロセッサシステムは、クロスバネットワーク５内のクロスバスイッチを介してプロセッサ間を接続するというマルチプロセッサシステムである。

【０００６】

【発明が解決しようとする課題】

図６に示す従来のマルチプロセッサシステムにおいては、２組のプロセッサ間でデータの受け渡しが行われる時に、他の２組のプロセッサ間でも同時にデータの受け渡し要求が発生した場合、どちらかのデータの受け渡しは待たされるように調停され、このため、データ処理の高速化を図ることができないという問題点があった。

【０００７】

また、データ伝送路に着目した場合、必ず、バス２を経由する構成とされているので、バス権を獲得してからデータの転送を行うというバストランザクションの時間が必要となり、この点からも、データ処理の高速化を図ることができないという問題点があった。

【０００８】

図７に示す従来のマルチプロセッサシステムにおいては、プロセッサ間を接続するクロスバスイッチが競合しない限り、図６に示す従来のマルチプロセッサシステムで発生するようなデータ転送のウエイトは生じないが、接続先のクロスバスイッチが競合した場合は、やはり、どちらかが待たされることになり、これがデータ処理の高速化を妨げるという問題点があった。

【０００９】

また、クロスバネットワーク５内に多くのクロスバスイッチを設ける必要があるため、回路構成が複雑になると共に、データ伝送路に着目した場合、必ずクロスバスイッチを経由する構成とされているため、この点からも、データ処理の高速化を図ることができないという問題点があった。

【００１０】

本発明は、かかる点に鑑み、データ処理の高速化を図ることができ、しかも、回路の簡略化を図ることができるようにしたマルチプロセッサシステムを提供することを目的とする。

【００１１】

【課題を解決するための手段】

本発明中、第１の発明のマルチプロセッサシステムは、データ入力端子とデータ出力端子とを有し、入力ポートを介して逐次入力される入力データの受け渡し先を判断し、各入力データを所定の受け渡し先に受け渡すようにプログラミングされる第１のプロセッサと、データ入力端子とデータ出力端子とを有し、第１のプロセッサから受け渡される入力データについて所定の処理を行うようにプログラミングされる第２～第ｎ（但し、ｎは３以上の整数）のプロセッサとを備えているというものである。

【００１２】

本発明中、第１の発明のマルチプロセッサシステムにおいては、第２～第ｎのプロセッサで命令の並列実行を行うことができるが、入力データを第１のプロセッサから第２～第ｎのプロセッサに受け渡すように構成されているので、図６に示す従来のマルチプロセッサシステムが必要としている複数のプロセッサに共有されるバスを設ける必要がなく、また、図７に示す従来のマルチプロセッサシステムが必要としているクロスバネットワークを設ける必要もない。

【００１３】
この結果、図６に示す従来のマルチプロセッサシステムの場合のようなデータ転送のウエイトという事態が発生することがなくなり、また、バス権を獲得するという手順も不要となり、更に、図７に示す従来のマルチプロセッサシステムの場合のようなクロスバスイッチが競合することによるデータ転送のウエイトという事態が発生することもなくなる。
【００１４】
本発明中、第２の発明のマルチプロセッサシステムは、第１の発明のマルチプロセッサシステムにおいて、第１のプロセッサは、所定の入力データについてはデータメモリに受け渡すようにプログラミングされ、第２～第ｎのプロセッサは、第１のプロセッサから受け渡された入力データについて所定の処理を行う場合、必要に応じて、データメモリに記憶された入力データを参照するようにプログラミングされるものであるというものである。
【００１５】
本発明中、第３の発明のマルチプロセッサシステムは、第１又は第２の発明のマルチプロセッサシステムにおいて、データ入力端子とデータ出力端子とを有する第ｎ＋１のプロセッサと、第２～第ｎのプロセッサの出力データ又は第ｎ＋１のプロセッサの出力データを出力ポートに転送する出力データ転送手段とを備え、第ｎ＋１のプロセッサは、第２～第ｎのプロセッサの出力データについて所定の処理を行い、その処理結果に基づいて、出力ポートに転送すべき出力データを出力データ転送手段に指示するようにプログラミングされるものであるというものである。
【００１６】
本発明中、第４の発明のマルチプロセッサシステムは、第３の発明のマルチプロセッサシステムにおいて、第１のプロセッサの前段に設けられ、入力ポートを介して逐次入力される入力データを格納する第１のＦＩＦＯ回路と、第２～第ｎのプロセッサの前段に設けられ、第１のプロセッサから第２～第ｎのプロセッサに受け渡される入力データを格納する第２～第ｎのＦＩＦＯ回路と、第２～第ｎのプロセッサの後段に設けられ、第２～第ｎのプロセッサの出力データを格納する第ｎ＋１～第２ｎ－１のＦＩＦＯ回路と、第ｎ＋１のプロセッサの後段に設けられ、第ｎ＋１のプロセッサの出力データを格納する第２ｎのＦＩＦＯ回路とを備えているというものである。
【００１７】
本発明中、第５の発明のマルチプロセッサシステムは、第４の発明のマルチプロセッサシステムにおいて、第ｎ＋１～第２ｎ－１のＦＩＦＯ回路は、ライトポインタと、第１、第２のリードポインタとを有し、ライトポインタは、前段のプロセッサに割り当てられ、第１のリードポインタは、第ｎ＋１のプロセッサに割り当てられ、第２のリードポインタは、出力データ転送手段に割り当てられているというものである。
【００１８】
本発明中、第６の発明のマルチプロセッサシステムは、第５の発明のマルチプロセッサシステムにおいて、第１～第ｎ＋１のプロセッサは、前段のＦＩＦＯ回路のカウンタを更新する命令と、前段のＦＩＦＯ回路がエンプティのときは、後続する命令を停止する命令と、後段のＦＩＦＯ回路のカウンタを更新する命令と、後段のＦＩＦＯ回路がフルのときは、後続する命令の実行を停止する命令とを有しているというものである。
【００１９】
本発明中、第７の発明のマルチプロセッサシステムは、第３、第４、第５又は第６の発明のマルチプロセッサシステムにおいて、出力データ転送手段は、ＤＭＡ（Direct Memory Access）コントローラであるというものである。
【００２０】
【発明の実施の形態】
以下、図１～図５を参照して、本発明の第１実施形態及び第２実施形態について説明する。
【００２１】
第１実施形態・・図１

図１は本発明の第１実施形態の要部を示すブロック回路図である。図１中、７はデータ入力端子とデータ出力端子とを有し、入力ポートを介して逐次入力されるパケット形式の入力データに含まれるコマンドをデコードして受け渡し先を判断し、各入力データを所定の受け渡し先（この例では、後述するデータメモリ９及びプロセッサ１０～１３のいずれか）に受け渡すようにプログラミングされたプロセッサである。

【００２２】

また、８は入力データの受け渡しのためにプロセッサ７によって専用される専用バス、９はデータメモリであり、プロセッサ７は、入力データが所定のデータ（この例では、後述するプロセッサ１０～１３がプロセッサ７から受け渡される入力データの処理のために参照すべきデータ）の場合には、入力データをデータメモリ９に受け渡すようにプログラミングされている。

【００２３】

また、１０～１３はデータ入力端子とデータ出力端子とを有し、データ入力端子を専用バス８に接続され、必要に応じて、データメモリ９に記憶されたデータを参照し、プロセッサ７から受け渡される入力データについて所定の処理を行うようにプログラミングされたプロセッサであり、これらプロセッサ１０～１３は、処理結果である出力データを保持するバッファメモリを備えている。

【００２４】

また、１４は後述するプロセッサ１５に制御され、プロセッサ１０～１３のいずれかの出力データを選択するセレクタ、１５はデータ入力端子とデータ出力端子とを有し、セレクタ１４を制御し、プロセッサ１０～１３の出力データについて所定の処理を行い、その処理結果に基づいて、プロセッサ１０～１３、１５の出力データのどれを出力ポートに転送するかを後述するＤＭＡコントローラ１７に指示するようにプログラミングされたプロセッサである。

【００２５】

また、１６は後述するＤＭＡコントローラ１７に制御され、プロセッサ１０～１３、１５の出力データを選択するセレクタ、１７はセレクタ１６を制御し、プロセッサ１５から指示されたプロセッサの出力データを出力ポートに転送するＤＭＡコントローラである。

【００２６】

このように構成された本発明の第１実施形態においては、プロセッサ７は、逐次入力される入力データに含まれるコマンドをデコードして受け渡し先を判断し、各入力データをデータメモリ９又はプロセッサ１０～１３のいずれかに受け渡し、プロセッサ１０～１３は、プロセッサ７から入力データを受け渡されると、必要に応じて、データバッファ９に記憶されたデータを参照し、入力データについて所定の処理を行うことになる。

【００２７】

そして、プロセッサ１５は、プロセッサ１０～１３の出力データについて所定の処理を行い、その処理結果に基づいて、プロセッサ１０～１３、１５の出力データのどれを出力ポートに転送するかをＤＭＡコントローラ１７に指示し、ＤＭＡコントローラ１７は、プロセッサ１５から指示されたプロセッサの出力データを出力ポートに転送することになる。

【００２８】

ここに、本発明の第１実施形態においては、プロセッサ１０～１３で命令の並列実行を行うことができるが、入力データをプロセッサ７から専用バス８を介してプロセッサ１０～１３に受け渡すように構成されているので、図６に示す従来のマルチプロセッサシステムが必要としている複数のプロセッサに共有されるバスを設ける必要がなく、また、図７に示す従来のマルチプロセッサシステムが必要としているクロスバネットワークを設ける必要もない。

【００２９】

この結果、図６に示す従来のマルチプロセッサシステムの場合のようなデータ転送のウエイトという事態が発生することがなくなり、また、バス権を獲得するという手順も不要となり、更に、図７に示す従来のマルチプロセッサシステムの場合のようなクロスバスイッ

チが競合することによるデータ転送のウエイトという事態が発生することもなくなる。
【００３０】
したがって、本発明の第１実施形態によれば、データ転送に要する時間を短縮し、データ処理の高速化を図ることができ、しかも、回路の簡略化を図ることができる。
【００３１】
なお、本発明の第１実施形態においては、プロセッサ１０～１３、１５の出力データを出力ポートに転送する出力データ転送手段として、ＤＭＡコントローラ１７を設けた場合について説明したが、この代わりに、プロセッサを設けるように構成しても良い。
【００３２】
第２実施形態・・図２～図５
図２は本発明の第２実施形態の要部を示すブロック回路図であり、図２中、１９は入力ポートを介して逐次入力されるパケット形式の入力データを格納するＦＩＦＯ（First In First Out）回路である。
【００３３】
また、２０はデータ入力端子とデータ出力端子とを有し、ＦＩＦＯ回路１９に格納された入力データに含まれるコマンドをデコードして受け渡し先を判断し、各入力データを所定の受け渡し先（この例では、後述するデータメモリ２２及びプロセッサ２７～３０のいずれか）に受け渡すようにプログラミングされたプロセッサである。
【００３４】
また、２１は入力データの受け渡しのためにプロセッサ２０により専用される専用バス、２２はデータメモリであり、プロセッサ２０は、入力データが所定のデータ（この例では、後述するプロセッサ２７～３０がプロセッサ２０から受け渡される入力データの処理のために参照すべきデータ）の場合には、入力データをデータメモリ２２に受け渡すようにプログラミングされている。
【００３５】
また、２３～２６はプロセッサ２０から後述するプロセッサ２７～３０に受け渡される入力データを格納するＦＩＦＯ回路、２７～３０はデータ入力端子とデータ出力端子とを有し、データ入力端子を専用バス２１に接続され、必要に応じて、データメモリ２２に記憶されたデータを参照し、ＦＩＦＯ回路２３～２６に格納された入力データについて所定の処理を行うようにプログラミングされたプロセッサである。
【００３６】
また、３１～３４はプロセッサ２７～３０の出力データを格納するＦＩＦＯ回路、３５は後述するプロセッサ３６に制御され、ＦＩＦＯ回路３１～３４に格納されているプロセッサ２７～３０の出力データを選択するセレクタである。
【００３７】
また、３６はデータ入力端子とデータ出力端子とを有し、セレクタ３５を制御し、ＦＩＦＯ回路３１～３４に格納されているプロセッサ２７～３０の出力データについて所定の処理を行い、その処理結果に基づいて、プロセッサ２７～３０、３６の出力データのどれを出力ポートに転送するかを後述するＤＭＡコントローラ３９に指示するようにプログラミングされたプロセッサである。
【００３８】
また、３７はプロセッサ３６の出力データを格納するＦＩＦＯ回路、３８は後述するＤＭＡコントローラ３９に制御され、ＦＩＦＯ回路３１～３４、３７の出力データを選択するセレクタ、３９はセレクタ３８を制御し、プロセッサ３６から指示されたプロセッサの出力データを出力ポートに転送するＤＭＡコントローラである。
【００３９】
図３はＦＩＦＯ回路３１の概念図であり、ＦＩＦＯ回路３２～３４、３７も同様に構成されている。図３中、４１はメモリ部、４２－０、４２－１、・・・４２－ｎはアドレス（ＡＤＤ）として、０番地、１番地、・・・ｎ番地を割り当てられたデータ格納領域である。

【００４０】
また、４３はライトアクセスするアドレスを指定するライトポインタ、４４、４５はリードアクセスするアドレスを指定するリードポインタであり、ライトポインタ４３はプロセッサ２７に割り当てられ、リードポインタ４４はプロセッサ３６に割り当てられ、リードポインタ４５はＤＭＡコントローラ３９に割り当てられている。
【００４１】
即ち、この例では、ＦＩＦＯ回路３１～３４は、１個のライトポインタと、２個のリードポインタとを有し、ＦＩＦＯ回路３１～３４のライトポインタは、それぞれ、ＦＩＦＯ回路３１～３４に出力データのライトを行うプロセッサ２７～３０に割り当てられ、ＦＩＦＯ回路３１～３４の一方のリードポインタは、ＦＩＦＯ回路３１～３４から出力データのリードを行うプロセッサ３６に割り当てられ、ＦＩＦＯ回路３１～３４の他方のリードポインタは、ＦＩＦＯ回路３１～３４から出力データのリードを行うＤＭＡコントローラ３９に割り当てられている。
【００４２】
このように構成された本発明の第２実施形態においては、入力ポートを介して逐次入力される入力データはＦＩＦＯ回路１９に格納され、プロセッサ２０は、ＦＩＦＯ回路１９から入力データをリードし、入力データに含まれるコマンドをデコードして受け渡し先を判断し、各入力データをデータメモリ２２及びＦＩＦＯ回路２３～２６のいずれかに受け渡すことになる。
【００４３】
そして、プロセッサ２７～３０は、ＦＩＦＯ回路２３～２６から入力データをリードし、必要に応じて、データメモリ２２に記憶されたデータを参照し、入力データについて所定の処理を行い、出力データをＦＩＦＯ回路３１～３４に格納することになる。
【００４４】
更に、プロセッサ３６は、ＦＩＦＯ回路３１～３４に格納されたプロセッサ２７～３０の出力データについて所定の処理を行い、ＤＭＡコントローラ３９に対して、プロセッサ２７～３０、３６の出力データのどれを出力ポートに出力するかを指示し、ＤＭＡコントローラ３９は、プロセッサ３６の指示に従い、ＦＩＦＯ回路３１～３４、３７に格納されているプロセッサ２７～３０、３６の出力データのいずれかを出力ポートに転送することになる。
【００４５】
ここに、たとえば、ＦＩＦＯ回路３１にプロセッサ２７の出力データがライトされる場合には、プロセッサ２７においては、ライトポインタ４３にアドレス０、１、・・・ｎを順に指定させ、プロセッサ２７の出力データをアドレス０、１、・・・ｎのデータ格納領域４２－０、４２－１、・・・４２－ｎに順に格納させる。
【００４６】
そして、プロセッサ３６においては、ＦＩＦＯ回路３１にプロセッサ２７の出力データが格納されると、リードポインタ４４にアドレス０、１、・・・ｎを順に指定させ、アドレス０、１、・・・ｎのデータ格納領域４２－０、４２－１、・・・４２－ｎから順にプロセッサ２７の出力データをリードすることになる。
【００４７】
そして、たとえば、プロセッサ３６からＤＭＡコントローラ３９に対してプロセッサ２７の出力データを出力ポートに転送する旨の指示があると、ＤＭＡコントローラ３９は、リードポインタ４４にアドレス０、１、・・・ｎを順に指定させ、ＦＩＦＯ回路３１のアドレス０、１、・・・ｎのデータ格納領域４２－０、４２－１、・・・４２－ｎから順にプロセッサ２７の出力データをリードし、これを出力ポートに転送することになる。
【００４８】
これに対して、プロセッサ３６からＤＭＡコントローラ３９に対してプロセッサ３６の出力データを出力ポートに転送する旨の指示があると、ＤＭＡコントローラ３９は、ＦＩＦＯ回路３７のリードポインタにアドレス０、１、・・・ｎを順に指定させ、ＦＩＦＯ回路

３７のアドレス０、１、・・・ｎのデータ格納領域から順にプロセッサ３６の出力データをリードし、これを出力ポートに転送することになる。

【００４９】

このように、ＦＩＦＯ回路３１は、入力されたデータから順に出力するというＦＩＦＯ回路の機能を損なうことなく、ＦＩＦＯ回路に格納されたデータの再利用を行うことができるようにしたものである。ＦＩＦＯ回路３２～３４も同様である。

【００５０】

また、図４は本発明の第２実施形態におけるプログラム例を示す図であり、図４中、“ SYNCff ”はデータ受け渡し先のＦＩＦＯ回路がフルの時には後続の命令の実行を停止する命令、“ SYNCfe ”はデータを受け取り元のＦＩＦＯ回路がエンプティの時には後続の命令の実行を停止する命令、“ FIFOup ”はＦＩＦＯ回路のＦＩＦＯカウンタをカウントアップする命令、“ FIFOdn ”はＦＩＦＯ回路のＦＩＦＯカウンタをカウントダウンする命令である。

【００５１】

また、“ mov r0,ofifo ・・・ ”は、プロセッサ２０のレジスタｒ０・・・のデータを受け渡し先のＦＩＦＯ回路に受け渡す命令、“ add ififo,r15 ・・・ ”は、ＦＩＦＯ回路２３のデータをプロセッサ２７のレジスタｒ１５・・・に加える命令を示している。

【００５２】

したがって、この例では、プロセッサ２０は、データ受け渡し先のＦＩＦＯ回路がフルの時は、後続の命令の実行を停止し、データ受け渡し先のＦＩＦＯ回路がフルでない時は、レジスタｒ０、ｒ１、ｒ２・・・の内容を受け渡し先のＦＩＦＯ回路に転送し、転送が終了した時は、転送先のＦＩＦＯ回路が備えるＦＩＦＯカウンタをカウントアップするというようにプログラミングされていることになる。

【００５３】

また、プロセッサ２７は、データ受け取り元のＦＩＦＯ回路がエンプティの時は、後続の命令の実行を停止し、データ受け取り元のＦＩＦＯ回路がエンプティでない時は、ＦＩＦＯ回路２３の内容をレジスタｒ１５、ｒ１４、ｒ１３・・・に加え、これを終了したときは、ＦＩＦＯ回路２３が備えるＦＩＦＯカウンタをカウントダウンするようにプログラミングされていることになる。

【００５４】

また、図５は本発明の第２実施形態における動作例を示す図であり、図４に示すプログラム例に基づく動作を示している。即ち、この例では、プロセッサ２０はＦＩＦＯ回路２３のフル解除待ちをしており、プロセッサ２７でＦＩＦＯ回路２３のＦＩＦＯカウンタのカウントダウン命令が実行されると、プロセッサ２０においてはデータ転送のウエイトが解除され、レジスタｒ０、ｒ１、ｒ２・・・の内容のＦＩＦＯ回路２３への転送が行われると共に、プロセッサ２７においては、ＦＩＦＯ回路２３のエンプティ解除待ちの状態となる。

【００５５】

そして、プロセッサ２０において、ＦＩＦＯ回路２３のカウントアップ命令が実行されると、プロセッサ２７においては、ウエイトが解除され、ＦＩＦＯ回路２３の内容をレジスタｒ１５、ｒ１４、ｒ１３・・・に加える命令が実行されている。

【００５６】

ここに、本発明の第２実施形態においては、プロセッサ２７～３０で命令の並列実行を行うことができるが、入力データをプロセッサ２０から専用バス２１を介してプロセッサ２７～３０に受け渡すように構成されているので、図６に示す従来のマルチプロセッサシステムが必要としている複数のプロセッサに共有されるバスを設ける必要がなく、また、図７に示す従来のマルチプロセッサシステムが必要としているクロスバネットワークを設ける必要もない。

【００５７】

この結果、図６に示す従来のマルチプロセッサシステムの場合のようなデータ転送のウエ

イトという事態が発生することがなくなり、また、バス権を獲得するという手順も不要となり、更に、図７に示す従来のマルチプロセッサシステムの場合のようなクロスバスイッチが競合することによるデータ転送のウエイトという事態が発生することもなくなる。

【００５８】

したがって、本発明の第２実施形態によれば、データ転送に要する時間を短縮し、データ処理の高速化を図ることができ、しかも、回路の簡略化を図ることができる。

【００５９】

また、データ受け渡し先のＦＩＦＯ回路がフルの時には後続の命令の実行を停止する命令“SYNCff”と、データ受け取り元のＦＩＦＯ回路がエンプティの時には後続の命令の実行を停止する命令“SYNCfe”と、ＦＩＦＯ回路のＦＩＦＯカウンタをカウントアップする命令“FIFOup”と、ＦＩＦＯ回路のＦＩＦＯカウンタをカウントダウンする命令“FIFOdn”とを備えるとしたことにより、データ転送における同期取りを容易に行うことができるという格別の効果を得ることができる。なお、ＦＩＦＯフル／エンプティを条件に分岐命令を実行するようにしても良い。

【００６０】

また、本発明の第２実施形態においては、プロセッサ２７～３０、３６の出力データを出力ポートに転送する出力データ転送手段として、ＤＭＡコントローラ３９を設けた場合について説明したが、この代わりに、プロセッサを設けるように構成しても良い。

【００６１】

【発明の効果】

以上のように、本発明によれば、第２～第ｎのプロセッサで命令の並列実行を行うことができるが、入力データを第１のプロセッサから第２～第ｎのプロセッサに受け渡すことができるように構成したことにより、図６に示す従来のマルチプロセッサシステムの場合のようなデータ転送のウエイトという事態が発生することがなくなり、また、バス権を獲得するという手順も不要となり、更に、図７に示す従来のマルチプロセッサシステムの場合のようなクロスバスイッチが競合することによるデータ転送のウエイトという事態が発生することもなくなるので、データ転送に要する時間を短縮し、データ処理の高速化を図ることができ、しかも、回路の簡略化を図ることができる。

【図面の簡単な説明】

【図１】本発明の第１実施形態の要部を示すブロック回路図である。

【図２】本発明の第２実施形態の要部を示すブロック回路図である。

【図３】本発明の第２実施形態が備えるＦＩＦＯ回路の概念図である。

【図４】本発明の第２実施形態におけるプログラム例を示す図である。

【図５】本発明の第２実施形態における動作例を示す図である。

【図６】従来のマルチプロセッサシステムの一例の要部を示すブロック回路図である。

【図７】従来のマルチプロセッサシステムの他の例の要部を示すブロック回路図である。

【符号の説明】

（図１）

７、１０～１３、１５　プロセッサ

８　専用バス

９　データメモリ

１４、１６　セレクタ

１７　ＤＭＡコントローラ

（図２）

１９、２３～２６、３１～３４、３７　ＦＩＦＯ回路

２０、２７～３０、３６　プロセッサ

２１　専用バス

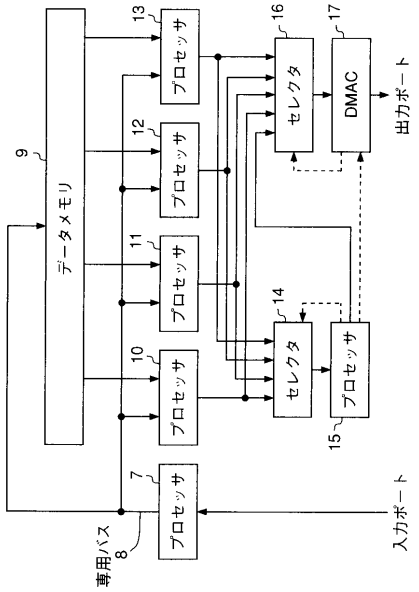
２２　データメモリ

３５、３８　セレクタ

３９　ＤＭＡコントローラ

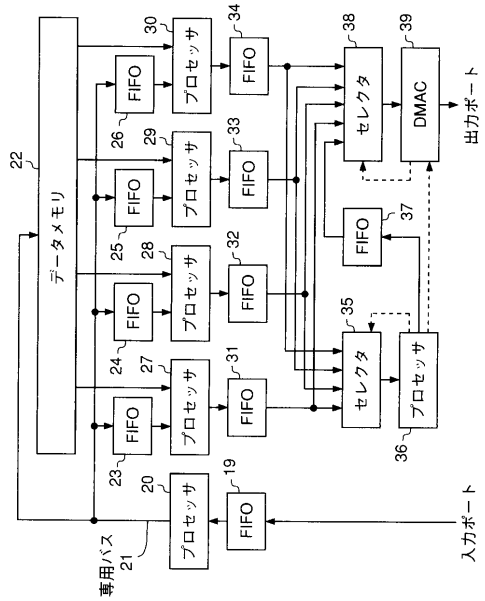
【図１】

本発明の第１実施形態の要部を示すブロック回路図



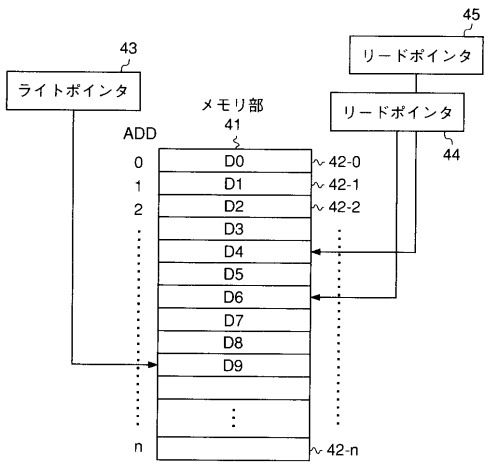
【図２】

本発明の第２実施形態の要部を示すブロック回路図



【図３】

ＦＩＦＯ回路３１の概念図



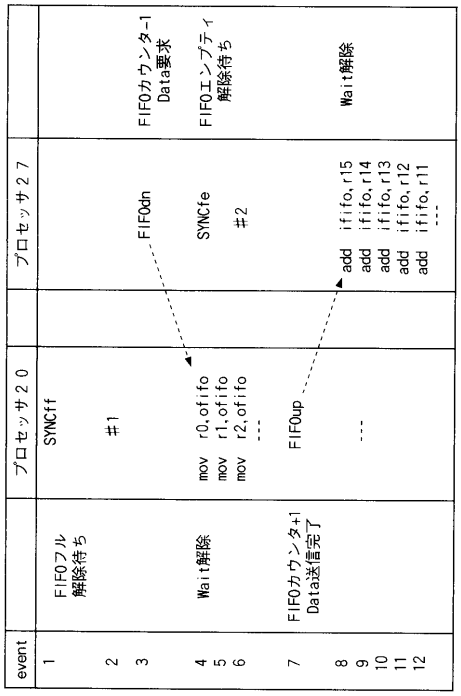
【図４】

本発明の第２実施形態におけるプログラム例を示す図

プロセッサ２０	プロセッサ２７
⇩	⇩
start: SYNCff	start: SYNCfe
#1	#2
mov r0,ofifo	add ififo,r15
mov r1,ofifo	add ififo,r14
mov r2,ofifo	add ififo,r13
---	add ififo,r12
FIFOup	add ififo,r11
---	---
b,start	b,start
nop	FIFOdn

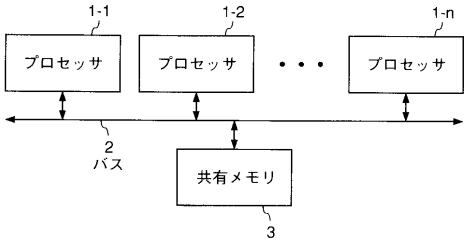
【図５】

本発明の第２実施形態における動作例を示す図



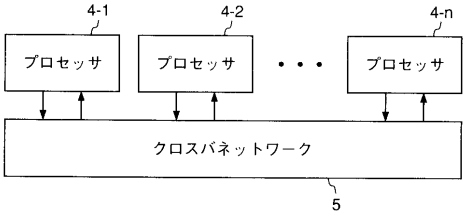
【図６】

従来のマルチプロセッサシステムの一例の要部を示すブロック回路図



【図７】

従来のマルチプロセッサシステムの他の例の要部を示すブロック回路図



フロントページの続き

(56)参考文献 特開平０７－３３４４６９（ＪＰ，Ａ）
特開平０６－２５９５８２（ＪＰ，Ａ）
特開平１０－１６２１３０（ＪＰ，Ａ）
特公平０７－１０４８４５（ＪＰ，Ｂ２）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 15/80
G06F 15/167-173