(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3539984号
(P3539984)
(45)発行日　平成16年7月7日(2004.7.7)　(24)登録日　平成16年4月2日(2004.4.2)

(51)Int.Cl.[7]　F I
G06F 9/46　G06F 9/46　313D
G06F 12/06　G06F 12/06　570A

請求項の数 5　(全 20 頁)

(21)出願番号　特願平4-281954
(22)出願日　平成4年10月20日(1992.10.20)
(65)公開番号　特開平5-210512
(43)公開日　平成5年8月20日(1993.8.20)
審査請求日　平成11年5月18日(1999.5.18)
(31)優先権主張番号　特願平3-272750
(32)優先日　平成3年10月21日(1991.10.21)
(33)優先権主張国　日本国(JP)

(73)特許権者　000003078
株式会社東芝
東京都港区芝浦一丁目１番１号
(74)代理人　100083806
弁理士　三好　秀和
(74)代理人　100068342
弁理士　三好　保男
(72)発明者　滝口　修弘
神奈川県川崎市幸区堀川町５８０番１号
株式会社東芝　半導体システム技術センター内
(72)発明者　川崎　壮一
神奈川県川崎市幸区堀川町５８０番１号
株式会社東芝　半導体システム技術センター内

最終頁に続く

(54)【発明の名称】プロセッサ

(57)【特許請求の範囲】
【請求項１】
実行されるそれぞれの処理で使用する専用レジスタ及び汎用レジスタと、
現在使用中のバンクを示す情報を保持する第１のレジスタと、
現在実行中の処理が未終了で他の処理が開始される場合に、前記第１のレジスタに保持されている現在使用中のバンクを示す情報が転送され、この情報を戻りバンクを示す情報として保持し、前記他の処理が終了した時に、保持している情報が前記第１のレジスタに転送される第２のレジスタと、
前記現在実行中の処理が未終了で他の処理が開始される場合に、前記第２のレジスタと前記専用レジスタ及び前記汎用レジスタに保持されている情報が外部バスとは別の専用バスを介して転送されて保持され、前記他の処理が終了した時に、保持している情報が前記第２のレジスタと前記専用レジスタ及び前記汎用レジスタに前記専用バスを介して転送されるバンクを複数セット備えたバンクメモリとを有し、
前記現在実行中の処理が未終了で前記他の処理が開始される場合に、前記第２のレジスタと前記専用レジスタ及び前記汎用レジスタに保持されている情報を前記専用バスを介して前記バンクに転送した後、前記第１のレジスタに保持されている現在使用中のバンクを示す情報を前記第２のレジスタに転送し、
前記他の処理が終了した時に、前記第２のレジスタに保持している情報を前記第１のレジスタに転送した後、前記バンクに保持している情報を前記第２のレジスタと前記専用レジスタ及び前記汎用レジスタに前記専用バスを介して転送すること

を特徴とするプロセッサ。
【請求項２】
実行されるそれぞれの処理で使用する専用レジスタ及び汎用レジスタと、
現在使用中のバンクを示す情報を保持する第１のレジスタと、
現在実行中の処理が未終了で他の処理が開始される場合に、前記第１のレジスタに保持されている現在使用中のバンクを示す情報が転送され、この情報を戻りバンクを示す情報として保持し、前記他の処理が終了した時に、保持している情報が前記第１のレジスタに転送される第１のバンクと、
前記現在実行中の処理が未終了で他の処理が開始される場合に、前記専用レジスタ及び汎用レジスタに保持されている情報が外部バスとは別の専用バスを介して転送されて保持され、前記他の処理が終了した時に、保持している情報が前記専用レジスタ及び汎用レジスタに前記専用バスを介して転送される第２のバンクとをそれぞれ複数セット備えたバンクメモリと、
前記第１のレジスタに保持された情報に基づいて前記第１及び第２のバンクの位置を算出する算出手段とを有し、
現在実行中の処理が未終了で前記他の処理が開始される場合に、前記専用レジスタ及び前記汎用レジスタに保持されている情報を前記専用バスを介して前記第２のバンクに転送した後、前記第１のレジスタに保持されている前記現在使用中のバンクを示す情報を前記第１のバンクに転送し、
前記他の処理が終了した時に、前記第１のバンクに保持している情報を前記第１のレジスタに転送した後、前記第２のバンクに保持している情報を前記専用レジスタ及び前記汎用レジスタに前記専用バスを介して転送すること
を特徴とするプロセッサ。
【請求項３】
実行されるそれぞれの処理で使用する専用レジスタ及び汎用レジスタと、
　現在の処理が終了した後に、現在使用中のバンクから戻るべきバンクである戻りバンクを示す情報と、前記専用レジスタ及び汎用レジスタに保持されている情報とが、外部バスとは別の専用バスを介して転送されるバンクを複数セット備えたバンクメモリと
を有し、
前記戻りバンクを示す情報が特定の値である場合には、前記戻りバンクを示す情報及び前記専用レジスタの情報がスタックメモリに転送されることを特徴とするプロセッサ。
【請求項４】
実行されるそれぞれの処理で使用する専用レジスタ及び汎用レジスタと、
　現在使用中のバンクを示す情報を保持する第１のレジスタと、
　現在の処理が終了した後に、現在使用中のバンクから戻るべきバンクである戻りバンクを示す情報を保持する第２のレジスタと、
　前記第２のレジスタと前記専用レジスタ及び汎用レジスタに保持されたそれぞれの情報が、外部バスとは別の専用バスを介して転送されるバンクを複数セット備えたバンクメモリと
を有し、
前記第２のレジスタに保持された情報が特定の値である場合には、前記第２のレジスタに保持された情報及び前記専用レジスタの情報がスタックメモリに転送されることを特徴とするプロセッサ。
【請求項５】
実行されるそれぞれの処理で使用する専用レジスタ及び汎用レジスタと、
　現在使用中のバンクを示す情報を保持する第１のレジスタと、
　現在の処理が終了した後に、現在使用中のバンクから戻るべきバンクである戻りバンクを示す情報が外部バスとは別の専用バスを介して転送される第１のバンクと、前記専用レジスタ及び前記汎用レジスタに保持されたそれぞれの情報が前記専用バスを介して転送される第２のバンクとをそれぞれ複数セット備えたバンクメモリと、

前記第１のレジスタに保持された情報に基づいて前記第１及び第２のバンクの位置を算出する算出手段と
を有し、
前記第１のバンクに保持された情報が特定の値である場合には、前記第１のバンクに保持された情報及び前記専用レジスタの情報がスタックメモリに転送されることを特徴とするプロセッサ。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】
本発明はバンク構造を有するプロセッサに関し、特に多重割込みに対応可能なプロセッサに関する。
【０００２】
【従来の技術】
通常、マイクロプロセッサのＣＰＵ（中央演算装置）中の汎用レジスタは、半導体チップの大きさなどによって使用数の制限を受けている。そのため、バンク構造と呼ばれる構造を持ったマイクロプロセッサがある。
【０００３】
バンクとは、汎用レジスタ（プログラム内でおもに演算に用いる）の集まりである。一般に汎用レジスタを格納する場合は、ＲＡＭ（ランダムアクセスメモリ）の領域を、ある決められた大きさで区切られたスペース（バンクＲＡＭと呼ぶ）に格納する。バンクを切り換えることによって、いくつかのバンクＲＡＭの中から１つのバンクＲＡＭを選択して、汎用レジスタの格納や読み出しをすることができる。
【０００４】
従来のマイクロプロセッサのうちで、バンク構造を持つものにおいては、バンクを切り換えることにより、汎用レジスタを多数持っていることと同等になる。例えば図９に示すように、マイクロプロセッサが、１つのバンクにある汎用レジスタ１０１を４レジスタとし、２５６バンク持っていれば、４×２５６＝１０２４個のレジスタを持っているのと同等になる。
【０００５】
バンクを複数持っている場合には、そのバンクの番号を示すバンクポインタというレジスタが存在する。そのレジスタの値を変更することにより、他のバンクに格納している内容を現在使用できる汎用レジスタとして取り扱うことができる。
【０００６】
従来のバンク構造を持っているマイクロプロセッサシステムにおいては、割込みが発生したときに使用していたバンク番号の退避先として、１）スタックメモリ上、２）汎用レジスタ上が考えられている。両者とも、割込みルーチンでバンクの切り換えが行われても、割込みルーチンが終わった後にそれぞれの退避先から割込み前のバンク番号を読んでくるので、そのバンク番号を見ることによって元のバンクに戻ることができる。
【０００７】
【発明が解決しようとする課題】
バンク構造を持っているマイクロプロセッサシステムのうち、割込みが発生したときに使用していたバンク番号をスタックメモリに退避するものは、マイクロプロセッサの外部バスに接続された低速のスタックメモリへの書き込み（割込みが発生したとき）、及び読み出し（割込みの処理から戻るとき）を行う必要があり、時間がかかる。
【０００８】
一方、バンク構造を持っているマイクロプロセッサシステムのうち、割込みが発生してバンクを切り換えたときに、使用していたバンクのバンク番号をバンク切り換え後の汎用レジスタに退避するものは、上記のスタックメモリに退避する場合に比べて、外部バスへの低速なアクセスがないので、戻り情報となるバンク番号の読み書きが速い。
【０００９】

しかし、バンク切り換え後の汎用レジスタを使って割込み処理を行うときには、戻り情報の入っている汎用レジスタを使用することができず、割込み処理で使用できる汎用レジスタに制限ができてしまう。もし、その汎用レジスタを使いたい場合には、スタックメモリなどに戻りバンク番号を退避しなければならなくなる。このため、最初からスタックメモリに退避する方式よりも遅くなる。

【００１０】

そこで、この発明は、上記に鑑みてなされたものであり、その目的とするところは、割込み処理において使用できるレジスタに制限を受けることなく、多重割込みを可能にするとともに割込み処理に対する高速な応答性を達成し得るプロセッサを提供することにある。

【００１１】

【課題を解決するための手段】

上記目的を達成するために、この発明は、実行されるそれぞれの処理で使用する専用レジスタ及び汎用レジスタと、現在使用中のバンクを示す情報を保持する第１のレジスタと、現在実行中の処理が未終了で他の処理が開始される場合に、前記第１のレジスタに保持されている現在使用中のバンクを示す情報が転送され、この情報を戻りバンクを示す情報として保持し、前記他の処理が終了した時に、保持している情報が前記第１のレジスタに転送される第２のレジスタと、現在実行中の処理が未終了で他の処理が開始される場合に、前記第２のレジスタと前記専用レジスタ及び前記汎用レジスタに保持されている情報が外部バスとは別の専用バスを介して転送されて保持され、前記他の処理が終了した時に、保持している情報が前記第２のレジスタと前記専用レジスタ及び前記汎用レジスタに前記専用バスを介して転送されるバンクを複数セット備えたバンクメモリとを有する。そして、前記現在実行中の処理が未終了で前記他の処理が開始される場合に、前記第２のレジスタと前記専用レジスタ及び前記汎用レジスタに保持されている情報を前記専用バスを介して前記バンクに転送した後、前記第１のレジスタに保持されている前記現在使用中のバンクを示す情報を前記第２のレジスタに転送し、前記他の処理が終了した時に、前記第２のレジスタに保持している情報を前記第１のレジスタに転送した後、前記バンクに保持している情報を前記第２のレジスタと前記専用レジスタ及び前記汎用レジスタに前記専用バスを介して転送する。

【００１２】

【作用】

上記構成において、この発明は、実行されるそれぞれの処理毎に、その処理で使用する専用レジスタと汎用レジスタ及びその処理の前に実行されて未終了の処理に対応したバンクの位置を示す情報を、それぞれ対応したバンクに外部バスとは別の専用バスを介して転送して保持するようにしている。

【００１３】

【実施例】

以下、図面を用いてこの発明の実施例を説明する。

【００１４】

図１はこの発明の第１の特徴の一実施例に係るプロセッサの要部構成を示す図である。

【００１５】

図１は、半導体チップ全体のうち、本発明にかかわるＣＰＵコア１のバンクメモリ２の部分を抽出拡大した図である。本来、半導体チップは、ＣＰＵ１やバンクメモリ２の他にシステムバスを通じて内蔵ＲＯＭ（図示せず）、タイマー（図示せず）やバスコントローラ（図示せず）などを搭載している。これらの要素の間では、システムバスを介してデータの受け渡しが行われる。チップ外とのデータの受け渡しは、さらにバスコントローラを介して行われる。内蔵ＲＯＭには、ユーザが作成したプログラムの命令語が格納されている。内蔵ＲＯＭはまた別のチップで実現してもよく、その場合は、バスコントローラを介してシステムバスによってＣＰＵ１とデータの受け渡しを行う。本明細書では、このシステムバスを外部バス、即ちＣＰＵからみて外部という意味で使用する。一方、本発明の特徴は、バンク切替の際、汎用レジスタ３の内容をバンクメモリ２へ、ＣＰＵ１とバンクメモ

リ２を結ぶ専用バスを介して退避／復帰を行うので、通常のデータの受け渡しを高速に行うことができる。

【００１６】

図１において、プロセッサは、ＣＰＵ１とバンクメモリ２を備えている。ＣＰＵ１は、１６ビット長の汎用レジスタＲＷ０～ＲＷ１５からなる汎用レジスタ群３と、カレントバンクポインタ（ＣＢＰ）４、プレビアスバンクポインタ（ＰＢＰ）５、プログラムカウンタ（ＰＣ）６、ユーザスタックポインタ（ＵＳＰ）７、プロセッサステータスワード（ＰＳＷ）８、及び図示していないが割込みスタックポインタ（ＩＳＰ）、特殊スタックポインタ（ＥＳＰ）、コンディションコード（ＣＣ）等の専用レジスタ群を備えている。

【００１７】

カレントバンクポインタ（ＣＢＰ）４は、現在使用中のバンクの位置を示したレジスタである。プレビアスバンクポインタ（ＰＢＰ）５は、現在使用中のバンクから、現在使用中のバンクを用いて実行される処理が終了した後に戻るべきバンクとなる戻りバンクの位置を示したレジスタである。ただし、ＰＢＰ５の値が“０”ならば、スタックメモリに戻るべきバンクの位置の情報が存在していることを示す。プログラムカウンタ（ＰＣ）６は、プログラムの命令語が入っているメモリ領域のアドレスを示すレジスタであり、プログラムが動作していると、値が変化する。

【００１８】

割込みスタックポインタ（ＩＳＰ）は割込みから戻ってくる時のＰＣ６、ＰＳＷ８、ＰＢＰ５、ＵＳＰ７の値が格納されているメモリ領域のアドレスを示すレジスタである。ユーザスタックポインタ（ＵＳＰ）とはユーザプログラムのサブルーチンから戻ってくる時のＰＣ６、ＰＳＷ８、ＰＢＰ５の値が格納されているメモリ領域のアドレスを示すレジスタである。特殊スタックポインタ（ＥＳＰ）は、システムコールから戻ってくる時のＰＣ６、ＰＳＷ８、ＰＢＰ５、ＵＳＰの値が格納されているメモリ領域のアドレスを示すレジスタである。コンディションコード（ＣＣ）は、演算結果の状態を示すフラグを格納しているレジスタである。プロセッサステータスワード（ＰＳＷ）８は、マイクロプロセッサの状態を示すレジスタである。

【００１９】

コンディションコードＣＣは図２に示すように、キャリーフラグＣ２１（演算の結果、最上位ビットから桁上げが発生したり、ボローが発生した場合に１がセットされる）、オーバフローフラグＶ２２（演算の結果、オーバフローが発生したときに１がセットされる）、ゼロフラグＺ２３（演算の結果がゼロの時に１がセットされる）、サインフラグＳ２４（演算結果の最上位ビットの値をコピーする）から構成される。

【００２０】

また、ＰＳＷ８は図３に示すように、タスクモードＴＭ３１（スタックポインタの指定）、バンクサイズ（ＢＳ）３２（現在使用中のバンクのサイズ）、遅延割込みの指定（ＤＩ）３３、特権モードの指定（ＲＧ）３４、シングルステップ動作の指定（ＳＳ）３５、割込み許可（ＥＩ）３６、割込みマスク（ＩＭ）３７、コンディションコード（ＣＣ）２１～２４の内容等の情報が格納されている。図１に戻って、バンクメモリ２は、プロセッサに内蔵されたＲＡＭ内に設けられている。バンクメモリ２は、例えばバンク１～バンク２５６を備え、バンクｉ（ｉ＝１～２５６）は４つの単位バンクからなり４ワードの大きさを有している。このようなバンクメモリ２において、例えば汎用レジスタ群３は連続したバンクに割り付けられ、ＰＣ６、ＰＢＰ５、ＰＳＷ８、ＵＳＰ７からなる専用レジスタは１つのバンクに割り付けられる。また、以下に説明する実施例にあっては、図４に示すように、１バンクが４個のレジスタからなる連続した４つのバンクのレジスタ群Ｒ０～Ｒ１５に汎用レジスタ３を割り付け、ＰＣ６、ＰＢＰ５、ＰＳＷ８、ＵＳＰ７からなる専用レジスタを１つのバンクの連続した４つの単位バンクに割り付ける。このようにして割り付けられるバンクと専用レジスタ群及び汎用レジスタ群３とは、外部バスを介することなく高バンド幅でデータ転送が行なわれるように接続構成されている。

【００２１】

このような構成において、任意の時間においてあるバンクを使用していたが、割込みが発生したことによって割り込みルーチン（割込みの処理をするためのプログラム）へ行き、バンクを切り換えた。その時に、そのまま割込みルーチンへ行ったのでは、どこのバンクへ戻るかわからなくなるので、ＰＢＰ５を用意して割込みルーチンでの処理が終了したときに使用するバンクの位置（以下、戻りバンク番号とする）がわかるようにする。

【００２２】

また、同時にＣＢＰ４を用意しておいて、現在使用中のバンクの位置がわかるようにし、さらにバンク切り換えが発生したときにＣＢＰ４の内容をＰＢＰ５にロードすることにより戻りバンク番号を保存する。バンクを戻すときは、ＰＢＰ５の値をＣＢＰ４にロードする。

【００２３】

プログラム実行時に使用しているレジスタは、汎用レジスタ、ＰＣ６、ＩＳＰ、ＵＳＰ７、ＥＳＰ、ＣＢＰ４、ＰＢＰ５、ＣＣ、ＰＳＷ８がある。

【００２４】

バンクＲＡＭには図４に示した汎用レジスタ群３、ＰＣ６、ＵＳＰ７、ＰＳＷ８、ＰＢＰ５を退避するための領域（バンクエリア）が確保され、各バンクに割り当てられる大きさが、それぞれ等しくなるように区切られている。ＰＢＰ５、ＰＣ６、ＵＳＰ７、ＰＳＷ８を退避するために１バンク、これらの専用レジスタを用いた処理で使用している汎用レジスタ（４，８，１２，１６のいずれかのレジスタ数でそれぞれの処理内容によって異なる）を退避するために、それぞれのレジスタ数に対応した１から４バンクが必要になる。

【００２５】

一方、サブルーチンに切り換えたときは、切り換えられたルーチンはＰＢＰ５、ＰＣ６、ＵＳＰ７、ＰＳＷ８をスタックメモリ（後入れ先出し記憶装置）（図示せず）に退避させる。

【００２６】

バンクを切り換える場合は、主に割込み処理と、サブルーチンコールと、システムコールによるものにわけられる。

【００２７】

割込み処理によってバンクを切り換える場合には、まず割込み条件が成立して割込みが発生すると，割込み要求信号を割込み制御回路に出力する。制御回路は同時に受けた割込みの中で、優先度のいちばん高いものをプロセッサに出力する。プロセッサは、発生した割込みが現在行なっている処理に比べて優先度が高く、かつ割込みが許可されていれば、割込みを受け付け、割込み制御回路に許可信号を出力する。

【００２８】

ただし、チップ外部からのノンマスカブル割込みは割込みの許可、不許可にかかわらず、現在行なっている処理に比べて優先度が高ければ割込みを受け付け、許可信号を出力する。

【００２９】

割込み制御回路は許可信号を受けて、その割込み要求元に対応するベクタを発生し、そのベクタからマイクロプロセッサが割込み処理のスタートアドレスを計算してその値をＰＣ６に書き込む。

【００３０】

割込み処理の開始時にスタートアドレスに書いてある１ワードを読み込んで参照する。該１ワードにはバンクを切り換えるか否かという情報と切り替えるバンク番号とがプログラムによって設定されている。該１ワードを割込みモードワードと呼ぶ。

【００３１】

バンクを切り換えるときは、現バンクの内容（汎用レジスタ群３、ＰＣ６、ＰＳＷ８、ＵＳＰ７、ＰＢＰ５）をバンクＲＡＭに退避し、ＣＢＰ４の内容をＰＢＰ５にロードし、ＣＢＰ４には割込み処理で使用するバンク番号を格納する。このバンク番号で指定されたバンクＲＡＭから汎用レジスタをロードして、バンクを切り換え、割込み処理を開始する。

【００３２】
次に、サブルーチンコールによってバンクを切り換える場合には、ユーザのプログラム中でＣＡＬＬ命令によってサブルーチンが呼び出される。ＵＳＰ７が示すスタックメモリに現在使用しているバンクのＰＣ６、ＰＢＰ５を退避し、スタックメモリに退避させたのでＰＢＰを“０”にし、ＣＢＰ４はそのままにする。次に、ＣＡＬＬ命令で指定されたアドレスをＰＣ６に書き込み、サブルーチンの処理を開始する。
【００３３】
サブルーチンではバンクを切り換えないので、バンクを切り換えるときには、プログラムのＣＨＧＢＫ命令（バンク切り換え命令）で行なう。そして、汎用レジスタのみをバンクＲＡＭに退避し、ＣＢＰ４に新しいバンク番号を書き込む。バンクを変えたがＰＢＰ５は操作しない。バンクを切り換える際に、バンクを１から３ずらせば、パラメータ受け渡し、結果受け取りが可能になる。一方、４以上ずらすと、全く重なりの無いバンク切り換えになる。
【００３４】
次に、システムコールによってバンクを切り換える場合には、システムコール（ソフトウェア割込み）はノンマスカブル割込みなので、割込みの許可、不許可にかかわらず、割込みを受け付ける。割込み制御回路によってその割込み処理に対応するベクタを発生し、そのベクタから割込み処理のスタートアドレスを計算する。それと同時に、割り込まれた処理で使用していたＰＣ６、ＰＢＰ５、ＰＳＷ８をスタックメモリに書き込み、スタックメモリに書き込んだためにＰＢＰに“０”をセットする。
【００３５】
次に、計算されたスタートアドレスをＰＣ６に書き込む。システムコールでは、通常の割込みの処理のように自動的にバンクを切り換えないので、プログラムでバンクの切り換えを行なう。バンクの切り換えは、ＣＨＧＢＫ命令（バンク切り換え命令）で行なう。
【００３６】
まず、汎用レジスタのみをバンクＲＡＭに退避し、ＣＢＰ４に新しいバンク番号を書き込む。ＰＢＰ５は操作しない。そして、システムコールの処理を行なう。また、サブルーチンコールと同様、バンクを１から３ずらせば、パラメータ受け渡し、結果受け取りが可能になる。
【００３７】
次に、割込みによるバンク切り換えから回復する場合には、割込みの処理が終了し、ＲＥＴＩ命令（割込みからのリターン命令）によって元のバンクへ回復する。ＣＢＰ４にＰＢＰ５の内容をロードして、ＣＢＰ４が示すバンクＲＡＭから汎用レジスタ３、ＰＣ６、ＰＢＰ５、ＰＳＷ８、ＵＳＰ７をロードする。これにより、元のバンクへ回復できる。
【００３８】
ただし、次回に同じ割込みが入ったときに続きを実行した時など、割込み処理で使用した汎用レジスタなどの内容を保存する必要がある場合には、戻る前にその内容をバンクＲＡＭに書き戻す命令をプログラムで行なう。
【００３９】
次に、サブルーチンコールによるバンク切り換えから回復する場合には、サブルーチンコールの処理が終了し、プログラムのＣＨＧＢＫ命令（バンク切り換え命令）によってサブルーチン中の元のバンクへ戻す。ＣＢＰ４にプログラムで切り換える前のＣＢＰ４の値を書き込み、ＣＢＰ４が示すバンクＲＡＭから汎用レジスタ３のみをロードする。これにより、元のバンクへ回復できる。
【００４０】
次に、ＲＥＴ命令（サブルーチンからの戻り命令）によって元のルーチンへ回復する。ＰＢＰ５が“０”なので、ＵＳＰ７が示すスタックメモリからＰＣ６、ＰＢＰ５をロードする。その際に、バンクの重なりを１から３に設定すれば、結果の受け渡しが可能である。
【００４１】
次に、システムコールによるバンク切り換えから回復する場合には、システムコールの処

理が終了し、ＣＨＧＢＫ命令（バンク戻し命令）によって元のバンクへ回復する。ＣＢＰ４にプログラムで切り換える前のＣＢＰ４の値を書き込み、ＣＢＰ４が示すバンクＲＡＭから汎用レジスタ３のみをロードする。これにより、元のバンクへ回復できる。ＲＥＴＩ（割込みルーチンからの戻り命令）命令によって元のルーチンへ回復する。ＰＢＰ５が“０”なので、ＵＳＰ７が示すスタックメモリからＰＣ６、ＰＢＰ５、ＰＳＷ８をロードする。そして、元のルーチンへ戻ることができる。

【００４２】

また、サブルーチンコールや、システムコールによってバンクを切り換える場合には、切り換えたレジスタバンクにパラメータを受け渡すことができることは上述した通りである。バンクを切り換える際にレジスタバンクを重ねてバンク番号を設定すると、汎用レジスタに入っているパラメータを受け渡すことが可能になる。

【００４３】

すなわち、汎用レジスタ同志が重なった部分の間でパラメータの受け渡しを行うことができる。このため、スタックメモリなどを介さずに直接行えるので、時間が短縮される。ただし、受け渡しは、元のレジスタバンクのレジスタ番号と新しいバンクのレジスタ番号が異なることになる。

【００４４】

次に、図５及び図６を用いてバンク切り換えの一実施例を説明する。

【００４５】

図５はバンクの一使用例を示す図であり、図６はバンク切り換え時の処理の流れを示す図である。

【００４６】

ここで、システムコール１はバンクを切り換えず、サブルーチン２はシステムコール２とする。また、使用中のＰＢＰ５が“０”の時は、ＰＣ６、ＰＢＰ５、ＰＳＷ８がスタックに退避していることを示す。

【００４７】

まずはじめに、メインルーチンが実行されている。この時には、ＣＢＰ４には使用しているバンクの番号の１５、ＰＢＰ５には初期化されて“０”が入っている。ＰＣ６、ＰＢＰ５、ＵＳＰ７、ＰＳＷ８を格納するバンクはバンク１５、汎用レジスタ３を格納するバンクＲＡＭはバンク１６から１９を使用している。

【００４８】

次に、サブルーチン１がコールされると、今まで使用していたバンクのＰＣ６、ＰＢＰ５をスタックメモリに退避し、スタックメモリに退避したのでＰＢＰに“０”をセットする。サブルーチンコール命令によって指定されたプログラム番地をＰＣ６にセットする。プログラムによって使用するバンクを１５から１２へ切り換える。使用していた汎用レジスタ３の内容をバンク１６から１９に退避する。ＣＢＰ４に１２をセットし、汎用レジスタにバンク１３から１６までの値をロードする。先のメインルーチンで使用していたバンク１６のＲ０からＲ３までのレジスタを、サブルーチン１のＲ１２からＲ１５へ直接パラメータを受け渡すことが出来る。

【００４９】

次に、割込みルーチン１が発生すると、ルーチン１の開始時に予めプログラムで設定済みのスタートアドレスにある１ワードを読んで、割込みルーチン１で使用するバンク番号と大きさを読み込む。これと同時に、サブルーチン１の汎用レジスタはバンク１３から１６、ＰＣ６、ＰＳＷ８、ＵＳＰ７、ＰＢＰ５はバンク１２へ退避し、ＣＢＰ４の内容の１２をＰＢＰ５にロードする。

【００５０】

ＰＣ６に割込みルーチン１に対応するスタートアドレスを書き込み、ＣＢＰ４には割込みルーチン１で使用するバンク番号の１を入れて、バンクを切り換える。バンク２とバンク３から汎用レジスタをロードして使用する。

【００５１】

次に、割込みルーチン２が発生すると、割込みルーチン１の場合と同様に、ルーチンの始めに予めプログラムで設定済みのスタートアドレスにある１ワードを読んで、割込みルーチン２で使用するバンク番号と大きさを読み込む。それと同時に、割込みルーチン１の汎用レジスタはバンク２から３、ＰＣ６、ＰＳＷ８、ＵＳＰ７、ＰＢＰ５はバンク１へ退避し、ＣＢＰ４の内容の１をＰＢＰ５にロードする。

【００５２】

ＰＣ６に割込みルーチン２に対応するスタートアドレスを書き込み、ＣＢＰ４には割込みルーチン２で使用するバンク番号の８を入れて、バンクを切り換える。汎用レジスタをバンク９からロードして使用する。

【００５３】

次に、割込みルーチン２の処理が終了すると、割込みルーチン２からのリターン命令が発生し、ＣＢＰ４にＰＢＰ５の値の１をロードする。バンク１からＰＣ６、ＰＢＰ＝１２、ＰＳＷ８、ＵＳＰ７をロードし、バンク２とバンク３から汎用レジスタをロードし、元のバンクへ回復する。そして、割込みルーチン１の処理の続きを行なう。

【００５４】

次に、割込みルーチン１の処理が終了すると、割込みルーチン１からのリターン命令が発生し、割込みルーチン２の処理が終了したのと同様に、ＣＢＰ４にＰＢＰ５の値の１２をロードする。バンク１２からＰＣ６、ＰＢＰ＝０、ＰＳＷ８、ＵＳＰ７をロードし、バンク１３からバンク１６までを汎用レジスタにロードし、元のバンク１２へ回復する。そして、サブルーチン１の処理の続きを行なう。

【００５５】

次に、サブルーチン１の処理の途中で、システムコール２が発生して、これを受け付けると、サブルーチン１で使用していたＰＣ６、ＰＳＷ８、ＰＢＰ＝０をスタックメモリに退避する。次に、そのソフトウェア割込みルーチンに対応するスタートアドレスをＰＣ６に書き込む。

【００５６】

ソフトウェア割込みは自動的にバンクを切り換えないので、ＣＢＰ４の値は１２で変わらない。スタックメモリに退避したので、ＰＢＰには“０”をセットする。プログラムでバンクを切り換えて、ＣＢＰ＝１０にする。汎用レジスタをバンク１１とバンク１２からロードして使用する。

【００５７】

次に、システムコール２の処理の途中で、割込みルーチン３が発生すると、割込みルーチン２の場合と同様に、ルーチンの始めに予めプログラムで設定済みのスタートアドレスにある１ワードを読んで、割込みルーチン３で使用するバンク番号と大きさを読み込む。それと同時に、システムコール２の汎用レジスタはバンク１１から１２、ＰＣ６、ＰＳＷ８、ＵＳＰ７、ＰＢＰ＝０はバンク１０へ退避し、ＣＢＰ４の内容の１０をＰＢＰ５にロードする。

【００５８】

ＰＣ６には割込みルーチンに対応するスタートアドレスを書き込み、ＣＢＰには割込みルーチン３で使用するバンク番号の４を入れて、バンクを切り換える。汎用レジスタはバンク５から７をロードして使用する。

【００５９】

次に、割込みルーチン３の処理中にシステムコール１が発生して、これを受け付けると、割込みルーチン３で使用していたＰＣ６、ＰＳＷ８、ＰＢＰ＝１０をスタックメモリに退避する。次に、そのソフトウェア割込みルーチンに対応するスタートアドレスをＰＣ６に書き込む。

【００６０】

ソフトウェア割込みは自動的にバンクを切り換えないので、ＣＢＰ４の値は変わらない。スタックメモリに退避したので、ＰＢＰは“０”をセットする。システムコール１はバンク切り換えを行わないので、ＣＢＰ＝４のままとなり、汎用レジスタはそのままバンク５

から７を使用する。

【００６１】

次に、システムコール１の処理が終了すると、割込みからの復帰命令によって割込みルーチン３に戻る。ＰＢＰは“０”なので、スタックメモリから割込みルーチン３で使用していたＰＣ６、ＰＳＷ８、ＰＢＰ＝１０をロードする。バンクは切り換えていないので、ＣＢＰ４の値は４で変わらず、汎用レジスタはそのままバンク５から７を使用し、割込みルーチン３の処理を再開する。

【００６２】

次に、割込みルーチン３の処理が終了すると、割込みルーチン３からのリターン命令が発生し、ＣＢＰ４にＰＢＰ５の値の１０をロードする。バンク１０からＰＣ６、ＰＢＰ＝０、ＰＳＷ８、ＵＳＰ７をロードし、バンク１１とバンク１２から汎用レジスタをロードし、元のバンク１０へ回復する。そして、システムコール２の続きを行なう。

【００６３】

次に、システムコール２の処理が終了すると、プログラムのバンク切り換え命令によってＣＢＰへ１２をロードして、バンク１２へ回復させる。ＰＢＰ＝０なので、スタックメモリからＰＣ６、ＰＢＰ＝０、ＰＳＷ８をロードする。汎用レジスタにバンク１３からバンク１６までをロードし、再びサブルーチン１の処理を行なう。

【００６４】

次に、サブルーチン１の処理が終了すると、プログラムによってＣＢＰ４に１５をロードし、バンクを戻す。ＰＢＰ＝０なので、スタックメモリからＰＣ６、ＰＢＰ＝０をロードする。汎用レジスタにバンク１６からバンク１９までをロードし、メインルーチンの続きを再開する。

【００６５】

なお、ＣＢＰの示す値と、ＰＢＰの示す値の両方を保持するレジスタを設けるようにしてもよい。このように、本発明では、戻りバンク番号のアクセスの際に、低速な外部バスを介してスタックメモリにアクセスしないですむので、割込みに対して高速な応答性を持つことができる。

【００６６】

また、ＰＣやＰＳＷ等の専用レジスタを複数セット設けることなく、多重割込みに対応することが可能となり、かつ、高速な割込み応答／復帰が可能となる。さらに、戻りバンク番号を保持しているレジスタの値によって、戻りバンク番号を格納する場所の選択ができるので、高速な応答を必要としない割込みの際の戻りバンク番号を外部のスタックメモリに置くこともでき、内部にあるバンクのＲＡＭを効率的に使うことができる。例えば割込み１が応答しているときに、同一の汎用レジスタを利用する割込み２が発生した場合、割込み２で使用するＰＣ、ＰＳＷをバンクに書き込んでしまうので、割込み１で使用していたＰＣ、ＰＳＷをスタックメモリに退避することができ、また、スタックメモリに置くことによってバンクメモリを節約することもできる。

【００６７】

次に、本発明の第２の特徴に係る実施例を説明する。この実施例の特徴は、前述した実施例のＰＢＰを持つことのかわりに、戻りバンク番号を格納するために１バンク使用することにある。

【００６８】

ここで、１バンクは、図４に示すようにレジスタが４個から構成され、連続する１から４バンクを汎用レジスタとして使用し、ＰＣ６、ＰＳＷ８、ＵＳＰ７を退避するために１バンク使用する。

【００６９】

任意の時間においてあるバンクを使用していたが、割込みが発生したことによって割り込みルーチン（割込みの処理をするためのプログラム）へ行き、バンクを切り換えた。その時に、そのまま割込みルーチンへ行ったのでは、どこのバンクへ戻るかわからなくなるので、バンクＲＡＭに戻りバンク番号を格納してわかるようにする。

【００７０】
また、同時にＣＢＰ４を用意しておいて、現在使用中のバンクの位置がわかるようにする。さらにバンク切り換えが発生したとき、切り換えた先の戻りバンク番号を格納するバンクＲＡＭにＣＢＰ４をロードすることにより終了後に戻るバンクの位置を保存する。逆に、ＣＢＰ４にロードすれば、バンクを戻すことができる。
【００７１】
一方、サブルーチンに切り換えたときは、切り換えられたルーチンはＰＣ６、ＵＳＰ７、ＰＳＷ８をスタックメモリ（後入れ先出し記憶装置）に退避させる。バンクを切り換える場合は、主に割込み処理と、サブルーチンコールと、システムコールによるものにわけられる。
【００７２】
割込み処理によってバンクを切り換える場合には、まず割込み条件が成立して割込みが発生すると割込み要求信号を割込み制御回路に出力する。制御回路は同時に受けた割込みの中で、優先度のいちばん高いものをプロセッサに出力する。プロセッサは、発生した割込みが現在行なっている処理に比べて優先度が高く、かつ割込みが許可されていれば、割込みを受け付け、割込み制御回路に許可信号を出力する。
【００７３】
ただし、チップ外部からのノンマスカブル割込みは割込みの許可、不許可にかかわらず、現在おこなっているルーチンに比べて優先度が高ければ割込みを受け付け、許可信号を出力する。
【００７４】
割込み制御回路は許可信号を受けて、その割込み要求元に対応するベクタを発生し、そのベクタからマイクロプロセッサが割込みルーチンのスタートアドレスを計算してその値をＰＣ６に書き込む。
【００７５】
処理の始めにスタートアドレスに書いてある１ワードを読んで、バンクを切り換えるか否かを選択する。この１ワードはプログラムによって設定しておく。
【００７６】
バンクを切り換えるときは、現バンクの内容（汎用レジスタ３、ＰＣ６、ＰＳＷ８、ＵＳＰ７）をバンクＲＡＭに退避し、ＣＢＰ４の内容を割込みルーチンで使用するバンクに、戻りバンク番号としてロードし、ＣＢＰ４には割込みルーチンで使用するバンク番号を入れる。このバンク番号で指定されたバンクＲＡＭから汎用レジスタ３をロードして、バンクを切り換え、割込み処理を開始する。
【００７７】
次に、サブルーチンコールによってバンクを切り換える場合には、ユーザのプログラム中でＣＡＬＬ命令によってサブルーチンが呼び出される。ＵＳＰ７が示すスタックメモリに現在使用しているバンクのＰＣ６を退避し、スタックメモリに退避させたのでバンクにある戻りバンク番号を“０”にし、ＣＢＰ４はそのままにする。次に、ＣＡＬＬ命令で指定されたアドレスをＰＣ６に書き込み、サブルーチンの処理を開始する。
【００７８】
プログラムでバンクを切り換えるときは、ＣＨＧＢＫ命令（バンク切り換え命令）で行なう。汎用レジスタのみをバンクＲＡＭに退避し、ＣＢＰ４に新しいバンク番号を書き込む。バンクを変えたが、ＰＢＰ５は操作しない。そして、ルーチンの処理を行なう。その際に、バンクを１から３ずらせば、パラメータ受け渡し、結果受け取りが可能になる。一方、４以上ずらすと、全く重なりの無いバンク切り換えになる。
【００７９】
次に、システムコールによってバンクを切り換える場合には、システムコール（ソフトウェア割込み）はノンマスカブル割込みなので、割込みの許可、不許可にかかわらず、割込みを受け付ける。割込み制御回路によってその割込みルーチンに対応するベクタを発生し、そのベクタから割込みルーチンのスタートアドレスを計算する。それと同時に、割り込

まれたルーチンで使用していたＰＣ６、ＰＳＷ８をスタックメモリに書き込み、スタックメモリに書き込んだために戻りバンク番号に０をセットする。

【００８０】

次に、計算されたスタートアドレスをＰＣ６に書き込む。システムコールでは、通常の割込みのルーチンのように自動的にバンクを切り換えないので、プログラムでバンクの切り換えを行なう。バンクの切り換えは、ＣＨＧＢＫ命令（バンク切り換え命令）で行なう。まず、汎用レジスタのみをバンクＲＡＭに退避し、ＣＢＰ４に新しいバンク番号を書き込む。ＰＢＰ５は操作しない。そして、ルーチンの処理を行なう。また、サブルーチンコールと同様、バンクを１から３ずらせば、パラメータ受け渡し、結果受け取りが可能になる。

【００８１】

次に、割込みによるバンク切り換えから回復する場合には、割込みの処理が終了すると、ＲＥＴＩ命令（割込みからのリターン命令）によって元のバンクへ回復する。ＣＢＰ４へバンクにある戻りバンク番号の内容をロードして、ＣＢＰ４が示すバンクＲＡＭから汎用レジスタ３、ＰＣ６、ＰＳＷ８、ＵＳＰ７をロードする。これにより、元のバンクへ回復できる。

【００８２】

ただし、次回に同じ割込みが入ったときに続きを実行したい時など、割込みルーチンで使用した汎用レジスタなどの内容を保持する必要がある場合には、戻る前にその内容をバンクＲＡＭに書き戻す命令をプログラムで行なう。

【００８３】

次に、サブルーチンコールによるバンク切り換えから回復する場合には、サブルーチンコールの処理が終了すると、プログラムのＣＨＧＢＫ命令（バンク切り換え命令）によってサブルーチン中の元のバンクへ戻す。ＣＢＰ４にプログラムで切り換える前のＣＢＰ４の値を書き込み、ＣＢＰ４の示すバンクＲＡＭから汎用レジスタのみをロードする。これにより、元のバンクへ回復できる。

【００８４】

次に、ＲＥＴ命令（サブルーチンからの戻り命令）によって元のルーチンへ回復する。戻りバンク番号が“０”なので、ＵＳＰ７が示すスタックメモリからＰＣ６をロードする。その際に、バンクの重なりを１から３に設定すれば、結果の受け渡しが可能である。

【００８５】

次に、システムコールによるバンク切り換えから回復する場合には、システムコールの処理が終了すると、ＣＨＧＢＫ命令（バンク戻し命令）によって元のバンクへ回復する。ＣＢＰ４にプログラムで切り換える前のＣＢＰ４の値を書き込み、ＣＢＰ４の示すバンクＲＡＭから汎用レジスタのみをロードする。これにより、元のバンクへ回復できる。ＲＥＴＩ（割込みルーチンからの戻り命令）命令によって元のルーチンへ回復する。戻りバンク番号が“０”なので、ＵＳＰ７が示すスタックメモリからＰＣ６、ＰＳＷ８をロードする。そして、元のルーチンへ戻ることができる。

【００８６】

次に、ＣＢＰ４が処理が終了後に使用すべきバンクの位置を示している値を格納しているバンクの位置を指定した実施例を、図７及び図８を用いて説明する。図７はバンクの一使用例であり、図８はバンク切り換え時の処理の流れを示す図である。

【００８７】

ここで、システムコール１はバンクを切り換えず、サブルーチン２はシステムコール２とする。また、ＣＢＰ４から１を引いた値が指示するバンクに戻りバンク番号を格納する。戻りバンク番号が“０”の時は、ＰＣ６、ＰＳＷ８がスタックメモリに退避していることを示す。

【００８８】

まずはじめに、メインルーチンが実行されている。この時には、ＣＢＰ４には使用しているバンクの番号の１５、バンク１４にある戻りバンク番号には初期化されて“０”が入っ

ている。ＰＣ６、ＵＳＰ７、ＰＳＷ８を格納するバンクはバンク１５、汎用レジスタ３を格納するバンクＲＡＭはバンク１６から１９を使用している。

【００８９】

次に、サブルーチン１がコールされると、今まで使用していたバンクのＰＣ６、バンク１４にある戻りバンク番号をスタックメモリに退避し、スタックメモリに退避したのでバンク１４にある戻りバンク番号に“０”をセットする。サブルーチンコール命令によって指定されたプログラム番地をＰＣ６にセットする。プログラムによって使用するバンクを１５から１３へ切り換える。使用していた汎用レジスタの内容をバンク１６から１９に退避する。バンク１４にあった戻りバンク番号の“０”をバンク１２にロードする。ＣＢＰに１３をセットし、汎用レジスタにバンク１４から１６までの値をロードする。

【００９０】

先のメインルーチンで使用していたバンク１６のＲ０からＲ３までのレジスタを、サブルーチン１のＲ８からＲ１１へ直接、パラメータを受け渡すことが出来る。

【００９１】

次に、割込みルーチン１が発生すると、ルーチン１の始めに予めプログラムで設定済みのスタートアドレスにある１ワードを読んで、割込みルーチン１で使用するバンク番号と大きさを読み込む。これと同時に、サブルーチン１の汎用レジスタはバンク１４から１６、ＰＣ６、ＰＳＷ８、ＵＳＰ７はバンク１３へ退避し、ＣＢＰ４の内容の１３をバンク１にある戻りバンク番号にロードする。

【００９２】

ＰＣ６に割込みルーチンに対応するスタートアドレスを書き込み、ＣＢＰ４には割込みルーチン１で使用するバンク番号の２を入れて、バンクを切り換える。バンク３から汎用レジスタをロードして使用する。

【００９３】

次に、割込みルーチン２が発生すると、割込みルーチン１の場合と同様に、ルーチンの始めに、予めプログラムで設定済みのスタートアドレスにある１ワードを読んで、割込みルーチン２で使用するバンク番号と大きさを読み込む。これと同時に、割込みルーチン１の汎用レジスタはバンク３、ＰＣ６、ＰＳＷ８、ＵＳＰ７はバンク２へ退避し、ＣＢＰ４の内容の２をバンク８にある戻りバンク番号にロードする。

【００９４】

ＰＣ６に割込みルーチンに対応するスタートアドレスを書き込み、ＣＢＰ４には割込みルーチン２で使用するバンク番号の９を入れて、バンクを切り換える。汎用レジスタをバンク１０からロードして使用する。

【００９５】

次に、割込みルーチン２の処理が終了すると、割込みルーチン２からのリターン命令が発生し、ＣＢＰ４にバンク８にある戻りバンク番号の値の２をロードする。バンク２からＰＣ６、ＰＳＷ８、ＵＳＰ７をロードし、バンク３から汎用レジスタをロードし、元のバンクへ回復する。バンク１にある戻りバンク番号は１３が入っている。そして、割込みルーチン１の処理の続きを行なう。

【００９６】

次に、割込みルーチン１の処理が終了すると、割込みルーチン１からのリターン命令が発生し、割込みルーチン１の処理が終了したのと同様に、ＣＢＰ４にバンク１にある戻りバンク番号の値の１３をロードする。バンク１３からＰＣ６、ＰＳＷ８、ＵＳＰ７をロードし、バンク１４からバンク１６までを汎用レジスタにロードし、元のバンク１３へ回復する。バンク１２にある戻りバンク番号は“０”が入っている。そして、サブルーチン１の処理の続きを行なう。

【００９７】

次に、サブルーチン１の処理の途中で、システムコール２が発生して、これを受け付けると、サブルーチン１で使用していたＰＣ６、ＰＳＷ８、バンク１２にある戻りバンク番号の値の“０”をスタックメモリに退避する。次に、そのソフトウェア割込みルーチンに対

応するスタートアドレスをＰＣ６に書き込む。

【００９８】

ソフトウェア割込みは自動的にバンクを切り換えないので、ＣＢＰ４の値は１３で変わらない。スタックメモリに退避したので、バンク１２にある戻りバンク番号は“０”をセットする。プログラムでバンクを切り換えて、ＣＢＰ４を１３から１２にする。バンク１２にあった戻りバンク番号の“０”をバンク１１にロードする。汎用レジスタをバンク１３からロードして使用する。

【００９９】

次に、システムコール２の処理の途中で、割込みルーチン３が発生すると、割込みルーチン２の場合と同様に、ルーチンの始めに予めプログラムで設定済みのスタートアドレスにある１ワードを読んで、割込みルーチン３で使用するバンク番号と大きさを読み込む。これと同時に、システムコール２の汎用レジスタはバンク１３、ＰＣ６、ＰＳＷ８、ＵＳＰ７はバンク１２へ退避し、ＣＢＰ４の内容の１２をバンク４にある戻りバンク番号にロードする。

【０１００】

ＰＣ６には割込みルーチンに対応するスタートアドレスを書き込み、ＣＢＰ４には割込みルーチン３で使用するバンク番号の５を入れて、バンクを切り換える。汎用レジスタはバンク６から７をロードして使用する。

【０１０１】

次に、割込みルーチン３の処理中にシステムコール１が発生して、これを受け付けると、割込みルーチン３で使用していたＰＣ６、ＰＳＷ８、バンク４にある戻りバンク番号の１２をスタックメモリに退避する。次に、そのソフトウェア割込みルーチンに対応するスタートアドレスをＰＣ６に書き込む。

【０１０２】

ソフトウェア割込みは自動的にバンクを切り換えないので、ＣＢＰ４の値は５で変わらない。スタックメモリに退避したので、バンク４にある戻りバンク番号は“０”をセットする。汎用レジスタはそのままバンク６から７を使用する。

【０１０３】

次に、システムコール１の処理が終了すると、割込みからの復帰命令によって割込みルーチン３に戻る。バンク４にある戻りバンク番号は“０”なので、スタックメモリから割込みルーチン３で使用していたＰＣ６、ＰＳＷ８をロードし、バンク４に戻りバンク番号の１２をロードする。バンクは切り換えていないので、ＣＢＰ４の値は５で変わらず、汎用レジスタはそのままバンク６から７を使用し、割込みルーチン３の処理を再開する。

【０１０４】

次に、割込みルーチン３の処理が終了すると、割込みルーチン３からのリターン命令が発生し、ＣＢＰ４にバンク４にある戻りバンク番号の値の１２をロードする。バンク１２からＰＣ６、ＰＳＷ８、ＵＳＰ７をロードし、バンク１３から汎用レジスタをロードし、元のバンク１２へ回復する。バンク１１にある戻りバンク番号は“０”が入っている。そして、システムコール２の続きを行なう。

【０１０５】

次に、システムコール２の処理が終了すると、プログラムのバンク切り換え命令によってＣＢＰを１２から１３に変えて、バンク１３に戻す。バンク１１にある戻り番号の“０”をバンク１２にロードする。バンク１２にある戻りバンク番号は“０”なので、スタックメモリからＰＣ６、ＰＳＷ８、バンク１２にある戻り番号の“０”をバンク１２にロードする。汎用レジスタにバンク１４からバンク１６までをロードし、再びサブルーチン１の処理を行なう。

【０１０６】

次に、サブルーチン１の処理が終了すると、プログラムによってＣＢＰ４を１３から１５に変え、バンク１５に戻す。バンク１２にあった戻り番号の“０”をバンク１４にロードする。バンク１４にある戻りバンク番号は“０”なので、スタックメモリからＰＣ６、バ

ンク１４にある戻りバンク番号の“０”をバンク１４にロードする。汎用レジスタにバンク１６からバンク１９までをロードし、メインルーチンの続きを再開する。

【０１０７】

このように、ＰＢＰを設けずＰＢＰに格納する内容を算出してバンクメモリに直接転送するようにしても、前述した実施例と同様の効果が得られる。また、ＰＢＰ５のレジスタが不要となり、構成の簡略化を図ることができる。

【０１０８】

さらに、通常は、データの受け渡しはシステムバスを介して行われるが、本発明によれば、ＣＰＵコアとバンクメモリ２とがシステムバスとは別の専用バスで接続されている。そのため、バンク切り替えの際、汎用レジスタ３及び専用レジスタの内容を該専用バスを介してバンクメモリ２へ退避／復帰するので、高速なデータの転送が実現できる。また、該専用バスの幅を広くすることによって（６４ビット、１２８ビツト）、さらに高速なバンク切り替えが可能になる。

【０１０９】

前記第２実施例をさらに効率良く動作させるため、以下第３実施例～第５実施例を図１０～図１２に示す。

【０１１０】

第１及び第２実施例に比して、第３～第５実施例では、ＰＣ６やＰＳＷ８などの専用レジスタを指すポインタと汎用レジスタを指すポインタを別々にもつことに特徴がある。

【０１１１】

図１０Ａは本発明による第３実施例に基づくプロセッサの構成を示している。第３実施例では、新たに、ＰＣ６及びＰＳＷ８を退避するバンクを指すＳＢＰ（スペシャルバンクポインタ）１０が備えられている。図１０及び図１０Ｃを使って動作を説明する。ＣＢＰ４及びＳＢＰ１０は命令及び割込によって書き換えられる。命令で書き換える時は、ＣＢＰ４及びＳＢＰ１０を同時に書き換えられる専用の命令が必要である。これをＣＣＢＰ（チェンジカレントバンクポインタ）命令と名付ける。ＣＣＢＰ命令のオペランドとしてＣＢＰ４及びＳＢＰ１０の２つが必要である。ＣＢＰ４とＳＢＰ１０をＬＯＡＤ命令等で別々に書き換えこともできるが、その場合には、ＣＢＰ４とＳＢＰ１０の両方を書き換えるまで割込みは禁止にしなければならないので効率が悪くなる。一方、割り込みで書き換える時は、割り込みテーブル中の割り込みモードワードに新しいＣＢＰ４及びＳＢＰ１０の値を記述しておく。

【０１１２】

次に、図１０Ｃを使って、本実施例に於て、汎用レジスタ３の本数を越えるレジスタを使用中に割り込みが発生した時の動作を説明する。

【０１１３】

ＣＢＰ４及びＳＢＰ１０の初期設定は、割り込み時には割込みモードワードにより、そして命令によりタスクチェンジ時には、ＣＣＢＰ命令により行う。割込み処理やタクス内に於て、汎用レジスタ３の本数を越えるワークレジスタが必要になるとＬＯＡＤ命令でＣＢＰ４を書き換えて汎用レジスタ３の内容を更新する。さらに、ここで新たな割込みが発生した場合には、初ず、汎用レジスタ３の内容をＣＢＰ４で指示されるバンクに退避し、そしてＰＣ６Ｃ及びＰＳＷ８をＳＢＰ１０で指示されるバンクに退避する。

【０１１４】

図１１は本発明による第４実施例に基づくプロセッサの構成を示している。

【０１１５】

第４実施例では、レジスタのワーク領域のベース位置のバンクを指すＢＢＰ（ベースバンクポインタ）１２及び、割込みが発生するときにＰＣ６とＰＳＷ８を退避するバンクメモリーと該ＢＢＰ１２との距離であるところの相対値を保持するＤＳＲ（ディスタントトゥペシャルレジスタ）１４を備えていることに特徴がある。図１１Ｂ及び図１１Ｃを使って第４実施例の動作を説明する。ＣＢＰ４，ＢＢＰ１２及びＤＳＲ１４は命令及び割込みで書き換えられる。命令で書き換える時は、ＣＢＰ４，ＢＢＰ１２及びＤＳＲ１４は同時に

書き換えられる専用の命令が必要である。これをＣＢＢＰ（チェンジ　ベース　バンクポインタ）命令と名付ける。ＣＢＢＰ命令のオペランドとしてＢＢＰ１２及びＤＳＲ１４を指定しておき、ＣＢＰ４にはＢＢＰ１２と同じ値をロードする。一方、割り込みによって書き換える場合は、割り込みテーブルの割込みモードワードに新しいＢＢＰ１２およびＤＳＲの値を記述しておく。

【０１１６】

次に、図１１Ｃを使って、本実施例に於て、汎用レジスタ３の本数を越えるレジスタを使用中に割込みが発生した時の動作を説明する。

【０１１７】

ＢＢＰ１２，ＣＢＰ４及びＤＳＲ１４の初期設定は、割込み時には割込みモードによって、また、命令によるタスクチェンジの時にはＣＢＢＰ命令によって行う。割込み時やタスク内に於て、汎用レジスタ３の本数を越えるレジスタが必要になるとＬＯＡＤ命令でＣＢＰ４を書き換えて汎用レジスタ３の内容を更新する。さらに、ここで新たな割込みが発生した場合には、初ず、汎用レジスタ３の内容をＣＢＰ４で指されるバンクに退避し、そしてＰＣ６及びＰＳＷ８をＢＢＰ１２及びＤＳＲ１４で指されるバンクに退避する。その結果、第４実施例では第３実施例に比してオペランドの長さを短くできる利点がある。一方、レジスタの数は第４実施例の方が多くなってしまい、またＰＣ６及びＰＳＷ８の退避／復帰の際、ＢＢＰ１２及びＤＳＲ１４からの情報を合わせて計算を行う分だけ余分に時間がかかる。

【０１１８】

図１２に示す第５実施例では、第４実施例に於けるＣＢＰ４の代わりにＢＢＰからＣＢＰ４までの距離の情報を保持するＤＣＲ（ディスタント・トウ・ＣＢＰレジスタ）が設けられている。このような構成によって、第５実施例では、第３及び第４実施例よりもさらにオペランドの長さを短くすることができる。また、第５実施例ではレジスタの数は第４実施例に比して少なくすることができる。

【０１１９】

【発明の効果】

以上説明したように、この発明は、バンクを切り換えた際に戻りバンクの位置を示す情報及び専用レジスタと汎用レジスタの情報を外部バスとは別の専用バスを介して転送して退避するバンクをバンクメモリに複数セット設けるようにしたので、割込み処理等で使用できる汎用レジスタに制限を受けることなく、割込み処理の応答／復帰を高速に行なうことが可能となる。

【０１２０】

また、退避領域を複数セット設けることによって多重割込みに対応することができる。

【０１２１】

さらに、専用レジスタの内容を退避する領域を選択するようにしたので、バンクメモリを効率的に利用することができる。

【０１２２】

さらに、ＰＣやＰＳＷなどの専用レジスタの内容を退避するバンクをタスクごとに指定でき、また専用レジスタと汎用レジスタを指すポインタが別々に用意されているので、バンクチェンジによる汎用レジスタの更新が高速に実行でき、また、タスク内やタスク間で該バンクを共有できるため、バンクメモリの使用効率が向上する。

【図面の簡単な説明】

【図１】この発明の第１の特徴の一実施例に係るプロセッサの要部構成を示す図である。

【図２】コンディションコードの一構成例を示す図である

【図３】プロセッサステータスワードの一構成例を示す図である。

【図４】バンクＲＡＭに格納するレジスタ群の構成を示す図である。

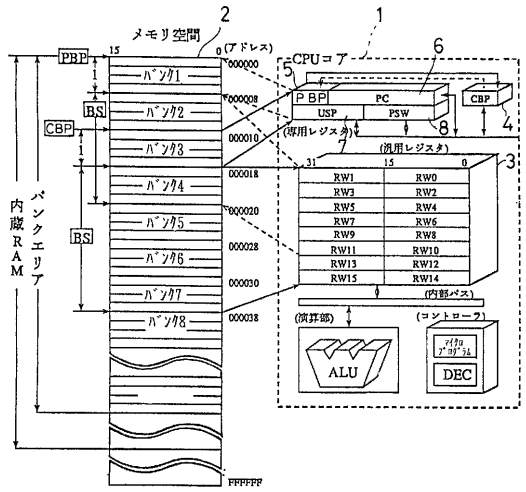
【図５】この発明の第１の特徴に係る実施例のバンクの使用例を示す図である。

【図６】図５に示すバンクの使用例におけるバンク切り換え時の処理の流れを示す図である。

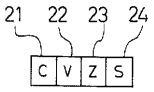
【図７】この発明の第２の特徴に係る実施例のバンクの使用例を示す図である。
【図８】図７に示すバンクの使用例におけるバンク切り換え時の処理の流れを示す図である。
【図９】従来のプロセッサにおけるバンク群の構成を示す図である。
【図１０】この発明の第３の特徴に係る実施例のバンクの使用例及び動作を示す図である。
【図１１】この発明の第４の特徴に係る実施例のバンクの使用例及び動作を示す図である。
【図１２】この発明の第５の特徴に係る実施例のバンクの使用例及び動作を示す図である。

【符号の説明】
１　ＣＰＵ
２　バンクメモリ
３　汎用レジスタ群
４　カレントバンクポインタ（ＣＢＰ）
５　プレビアスバンクポインタ（ＰＢＰ）
６　プログラムカウンタ（ＰＣ）
７　ユーザスタックポインタ（ＵＳＰ）
８　プロセッサステータスワード（ＰＳＷ）
１０　スペシャルバンクポインタ（ＳＢＰ）
１２　ベースバンクポインタ（ＢＢＰ）
１４　ディスタントトゥスペシャルレジスタ（ＤＳＲ）
１６　ディスタントトゥＣＢＰレジスタ（ＤＣＲ）

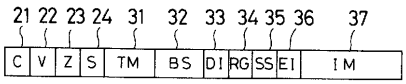
【図１】



【図２】



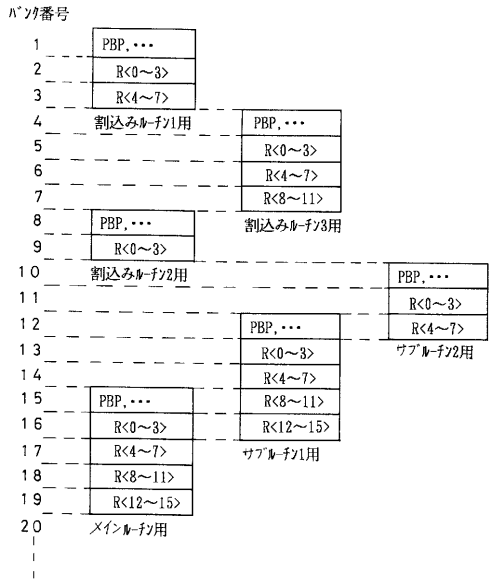
【図３】



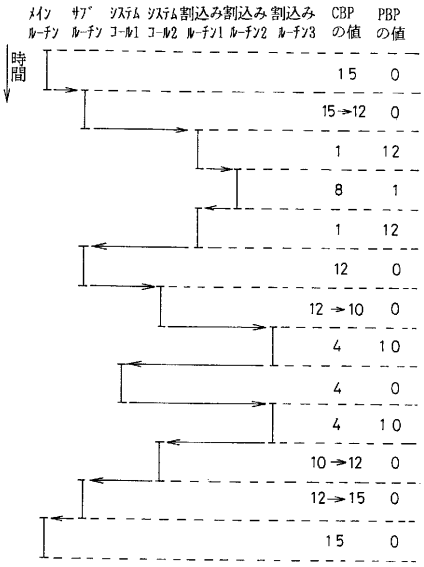
【図４】

PC	PBP	PSW	USP
R0	R1	R2	R3
R4	R5	R6	R7
R8	R9	R10	R11
R12	R13	R14	R15

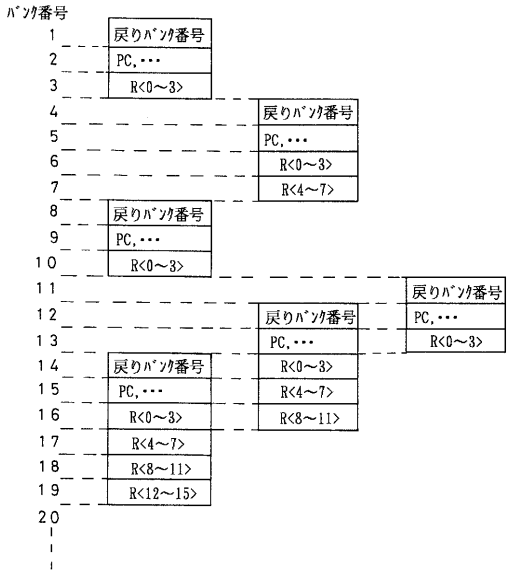
【図５】


ﾊﾞﾝｸ番号
PBP,･･･
R<0～3>
R<4～7>
割込みﾙｰﾁﾝ1用
PBP,･･･
R<0～3>
R<4～7>
R<8～11>
割込みﾙｰﾁﾝ3用
PBP,･･･
R<0～3>
割込みﾙｰﾁﾝ2用
PBP,･･･
R<0～3>
R<4～7>
ｻﾌﾞﾙｰﾁﾝ2用
PBP,･･･
R<0～3>
R<4～7>
R<8～11>
R<12～15>
ｻﾌﾞﾙｰﾁﾝ1用
PBP,･･･
R<0～3>
R<4～7>
R<8～11>
R<12～15>
ﾒｲﾝﾙｰﾁﾝ用

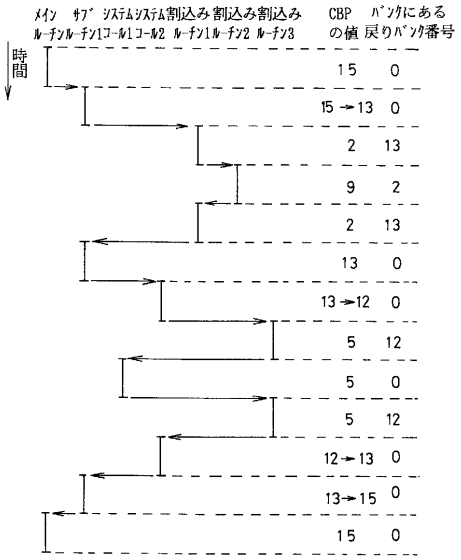
【図６】


ﾒｲﾝ ｻﾌﾞ ｼｽﾃﾑ ｼｽﾃﾑ 割込み 割込み 割込み CBP PBP
ﾙｰﾁﾝ ﾙｰﾁﾝ ｺｰﾙ1 ｺｰﾙ2 ﾙｰﾁﾝ1 ﾙｰﾁﾝ2 ﾙｰﾁﾝ3 の値 の値
時間
15 0
15→12 0
1 12
8 1
1 12
12 0
12→10 0
4 10
4 0
4 10
10→12 0
12→15 0
15 0

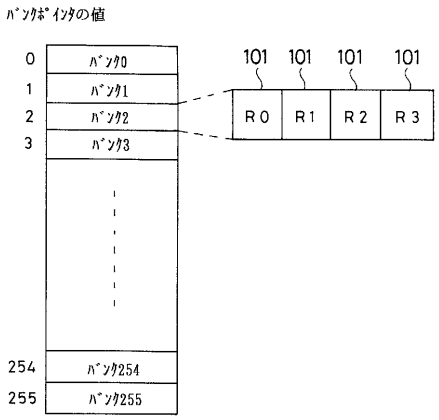
【図７】


ﾊﾞﾝｸ番号
戻りﾊﾞﾝｸ番号
PC,･･･
R<0～3>
戻りﾊﾞﾝｸ番号
PC,･･･
R<0～3>
R<4～7>
戻りﾊﾞﾝｸ番号
PC,･･･
R<0～3>
戻りﾊﾞﾝｸ番号
PC,･･･
R<0～3>
戻りﾊﾞﾝｸ番号
PC,･･･
R<0～3>
R<4～7>
戻りﾊﾞﾝｸ番号
PC,･･･
R<0～3>
R<4～7>
R<8～11>
R<12～15>

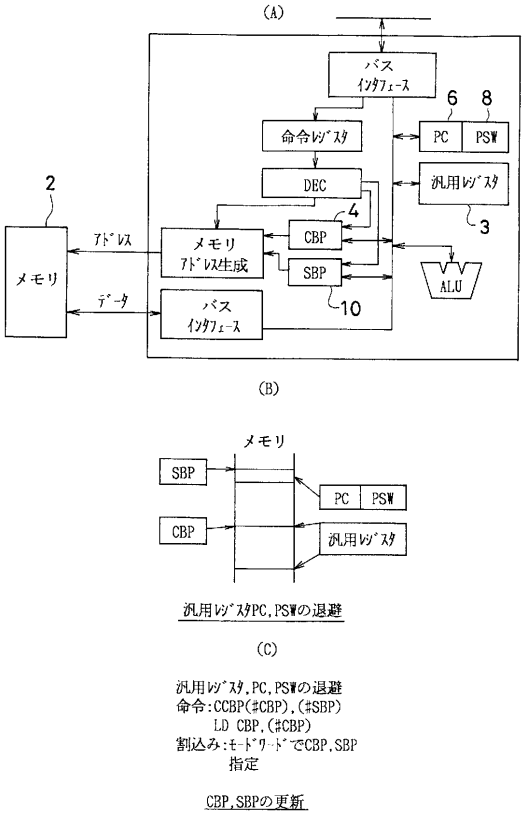
【図８】


ﾒｲﾝ ｻﾌﾞ ｼｽﾃﾑ ｼｽﾃﾑ 割込み 割込み 割込み CBP ﾊﾞﾝｸにある
ﾙｰﾁﾝ ﾙｰﾁﾝ1 ｺｰﾙ1 ｺｰﾙ2 ﾙｰﾁﾝ1 ﾙｰﾁﾝ2 ﾙｰﾁﾝ3 の値 戻りﾊﾞﾝｸ番号
時間
15 0
15→13 0
2 13
9 2
2 13
13 0
13→12 0
5 12
5 0
5 12
12→13 0
13→15 0
15 0

【図９】



【図１０】



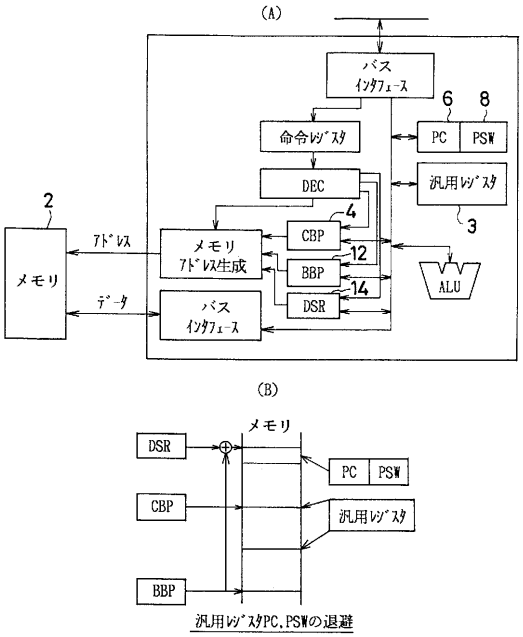
汎用ﾚｼﾞｽﾀPC,PSWの退避

(C)

汎用ﾚｼﾞｽﾀ,PC,PSWの退避
命令:CCBP(#CBP),(#SBP)
LD CBP,(#CBP)
割込み:ﾓｰﾄﾞﾜｰﾄﾞでCBP,SBP
指定

CBP,SBPの更新

【図１１】



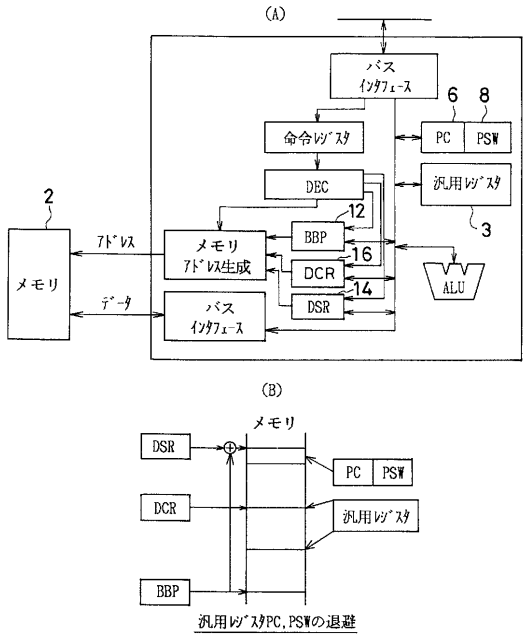
汎用ﾚｼﾞｽﾀPC,PSWの退避

(C)

命令:CBBP(#BBP),(#DSR)
LD CBP,(#CBP)
割込み:ﾓｰﾄﾞﾜｰﾄﾞでBBP(=BP)
DSRを指定

CBP,BBP,DSRの更新

【図１２】



汎用ﾚｼﾞｽﾀPC,PSWの退避

(C)

命令:CBBP(#BBP),(#DSR)
LD DCR,(#DCR)
割込み:ﾓｰﾄﾞﾜｰﾄﾞでBBP,DSR
を指定

BBP,DCR,DSRの更新

フロントページの続き

(72)発明者　山田　泰生
神奈川県川崎市幸区堀川町５８０番１号　株式会社東芝　半導体システム技術センター内
(72)発明者　加沼　安喜良
神奈川県川崎市幸区堀川町５８０番１号　株式会社東芝　半導体システム技術センター内
(72)発明者　矢口　俊行
神奈川県川崎市幸区堀川町５８０番１号　株式会社東芝　半導体システム技術センター内

審査官　▲羽▼立　章二

(56)参考文献　特開平３－７５８３１（ＪＰ，Ａ）
特開平２－１４３３０（ＪＰ，Ａ）
特開昭５３－９８７５３（ＪＰ，Ａ）
可児賢二編著，ＶシリーズマイクロコンピュータＩ，丸善株式会社，１９８６年　1月２０日，第1版，ｐ．１４８－１４９

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 9/46- 9/54
G06F12/00,550-12/06
G06F13/16-13/18