(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5148029号
(P5148029)

(45)発行日　平成25年2月20日(2013.2.20)　(24)登録日　平成24年12月7日(2012.12.7)

(51)Int.Cl.　　F I
G06F 9/318 (2006.01)　　G06F 9/30 320A
G06F 17/50 (2006.01)　　G06F 17/50 654D
G06F 9/45 (2006.01)　　G06F 9/44 322Z

請求項の数 9　(全 14 頁)

(21)出願番号　特願2000-600174 (P2000-600174)
(86)(22)出願日　平成12年1月26日 (2000.1.26)
(65)公表番号　特表2002-537599 (P2002-537599A)
(43)公表日　平成14年11月5日 (2002.11.5)
(86)国際出願番号　PCT/EP2000/000590
(87)国際公開番号　WO2000/049496
(87)国際公開日　平成12年8月24日 (2000.8.24)
審査請求日　平成19年1月25日 (2007.1.25)
(31)優先権主張番号　99200431.7
(32)優先日　平成11年2月15日 (1999.2.15)
(33)優先権主張国　欧州特許庁 (EP)

前置審査

(73)特許権者　590000248
コーニンクレッカ　フィリップス　エレクトロニクス　エヌ　ヴィ
オランダ国　５６２１　ベーアー　アインドーフェン　フルーネヴァウツウェッハ　１
(74)代理人　100087789
弁理士　津軽　進
(74)代理人　100122769
弁理士　笛田　秀仙
(72)発明者　デ　オリベイラ　カストルプ　ペレイラ　ベルナルド
オランダ国　５６５６　アーアー　アインドーフェン　プロフ　ホルストラーン　６

最終頁に続く

(54)【発明の名称】構成可能な機能ユニットを備えるデータプロセッサ及びそのようなデータプロセッサを使用する方法

(57)【特許請求の範囲】
【請求項１】
構成プログラムをロードすることにより実行時に効果を再定義することが可能な、再構成可能な命令を実行することができる構成可能な機能ユニットを含むプロセッサを用いてコンピュータプログラムを実行する方法において、
コンパイラが、再構成可能な命令の組み合わせを選択するステップであって、各再構成可能な命令について入力オペランドのどのビットが当該命令の結果に影響を与えるかを決定し、該影響を与えるビットのうち各再構成可能な命令間で共用されない入力オペランドのビットの数に応じて前記再構成可能な命令間の非類似性を決定し、前記決定された非類似性に基づいて、相対的に少ない非類似性を持つ前記再構成可能な命令をグループ化することを含み、前記共用されない入力オペランドのビットは、当該再構成可能な命令の結果に影響を与えると決定された前記ビットである、ステップと、
前記コンパイラが、前記再構成可能な命令の各組み合わせに対して構成プログラムを各々発生するステップと、
前記プロセッサが、前記コンピュータプログラムの実行の間に、前記組み合わせのうちの１つの組み合わせからの命令が必要とされ、且つ、前記構成可能な機能ユニットが前記１つの組み合わせ用の構成プログラムを用いて構成されていない都度に、前記１つの組み合わせの全命令に対する構成プログラムを前記構成可能な機能ユニットにロードするステップと、
を有していることを特徴とする方法。

【請求項２】
請求項１に記載の方法において、前記構成プログラムを発生するステップが、当該構成プログラムに関連する前記再構成可能な命令の前記１つの組み合わせに対してハードウェア資源使用を最小化するステップを有していることを特徴とする方法。
【請求項３】
請求項２に記載の方法において、前記コンピュータプログラム実行の間に前記組み合わせから複数の命令を選択し、且つ、これら複数の命令によりオペランドデータを処理するためのハードウェア資源使用が最小化されることを特徴とする方法。
【請求項４】
請求項２に記載の方法において、前記構成可能な機能ユニットは、交点スイッチを、オペランドデータ用の入力端子と、該交点スイッチの出力端子を異なる論理合成回路に接続する接続ラインとの間に含み、前記交点スイッチは前記構成プログラムにより制御され、前記交点スイッチ内の接続のプログラミングが前記構成可能な機能ユニットにおけるハードウェア資源使用を最小化するために使用されることを特徴とする方法。
【請求項５】
請求項４に記載の方法において、前記組み合わせから前記再構成可能な命令を選択するためのビットの少なくとも一部が、前記オペランドデータと交換可能に前記交点スイッチに供給されることを特徴とする方法。
【請求項６】
構成プログラムをロードすることにより効果を実行時に再定義することが可能な、再構成可能な命令を実行することができる構成可能な機能ユニットを有するようなプロセッサであって、前記構成可能な機能ユニットが、再構成可能な命令を実行するために必要とされる際に前記構成プログラムが未だロードされていない場合に該構成プログラムのロードを起動するロード起動回路を有するようなプロセッサにおいて、前記構成プログラムは少なくとも２つの組み合わされた再構成可能な命令の効果を定義し、前記ロード起動回路は前記組み合わされた再構成可能な命令のうちの少なくとも１つが必要とされる場合に該組み合わされた命令の全てに対する前記構成プログラムのロードを起動することを特徴とするプロセッサであって、
前記組み合わされた再構成可能な命令は、コンパイラが、各再構成可能な命令について入力オペランドのどのビットが当該命令の結果に影響を与えるかを決定し、該影響を与えるビットのうち各再構成可能な命令間で共用されない入力オペランドのビットの数に応じて前記再構成可能な命令間の非類似性を決定し、前記決定された非類似性に基づいて、相対的に少ない非類似性を持つ前記再構成可能な命令をグループ化して組み合わされるものであり、前記共用されない入力オペランドのビットは、当該再構成可能な命令の結果に影響を与えると決定された前記ビットである、プロセッサ。
【請求項７】
請求項６に記載のプロセッサにおいて、前記構成可能な機能ユニットは前記組み合わせから命令を選択するための命令選択入力端子と、オペランドデータ入力端子と、合成ロジックと、結果出力端子とを有し、前記オペランドデータ入力端子及び前記命令選択入力端子は、共に、命令選択ビット及びオペランドデータビットが交換可能に使用することができるように、前記合成ロジックを介して前記結果出力端子に結合されていることを特徴とするプロセッサ。
【請求項８】
請求項７に記載のプロセッサにおいて、前記構成可能な機能ユニットは、前記合成ロジックと、前記オペランドデータ入力端子及び前記命令選択入力端子との間に交点スイッチを有し、該交点スイッチは前記命令選択ビット及び前記オペランドデータビットを前記合成ロジックに機能的に交換可能に接続することができることを特徴とするプロセッサ。
【請求項９】
請求項６に記載のプロセッサにおいて、前記構成可能な機能ユニットが、前記構成可能な機能ユニット内に交点スイッチ及び論理部を有し、前記組み合わせの構成可能な命令を

構成する可能な接続の部類の何れが前記構成プログラムの制御の下で接続されるかに無関係に、前記交点スイッチ及び前記論理部を介しての、オペランドの演算結果の略一定の遅延を与えることを特徴とするプロセッサ。

【発明の詳細な説明】

【０００１】

【技術分野】

本発明は、その効果を実行時に再定義することが可能な、再構成可能な命令（reconfigurable instructions）を実行することができる構成可能な（configurable）機能ユニットを含むプロセッサを用いてコンピュータプログラムを実行する方法に関する。また、本発明は斯様な方法を使用するデータプロセッサにも関する。

【０００２】

【背景技術】

構成可能な機能ユニットを含むプロセッサを用いてコンピュータプログラムを実行する方法は、John Schewelにより編集された１９９５年の“会議録、高速回路基板開発及び再構成可能なコンピューティング用のＦＰＧＡ”（ＳＰＩＥ2607会議録）の第９２～１０３頁に公開されたMichael J. Wirthlin及びBrad L. Hutchingsによる“ＤＩＳＣ：ダイナミック命令セットコンピュータ”なる題名の文献から既知である。

【０００３】

この文献は、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）を含む機能ユニットを備えるようなデータプロセッサを記載している。ＦＰＧＡは、出力信号を入力信号の関数として生成するような回路である。ＦＰＧＡは、構成可能な回路素子の行及び列のマトリクスからなる。入力と出力との関係は、該ＦＰＧＡの異なる回路素子間の接続及びこれら回路素子の機能を制御するメモリセルに情報をロードすることにより構成（configure）することができる。

【０００４】

構成プログラム（configuration program）の使用は、マイクロプログラムからは区別されるべきである。周知のように、マイクロプログラムは、機能回路を制御するために使用される個々の制御信号を規定する。異なる制御信号が、マイクロコードの実行の異なる段階に対して、且つ、異なる命令に対して規定される。対照的に、構成プログラムのビットを記憶する該当メモリセルは、入出力関係に対して不変的制御を有する。即ち、これらメモリセルは、実行されている命令及び如何なる実行段階とも無関係に回路素子を不変的に制御する。通常、制御される入出力関係は、時間的に連続した回路特性である。

【０００５】

構成プログラムは、異なる構成可能な命令を実行するために導出される。Whirthlin他による上記文献によれば、ＦＰＧＡマトリクスは回路素子の行の多数のバンド（bands）に分割される。各構成プログラムは、１以下のバンドしか占めることはなく、如何なるバンドにも配置することができる。実行時において、或る構成可能な命令に遭遇した場合は、この命令用の構成プログラムが上記バンドのうちの何れかに既にロードされているかが判断される。もしそうなら、この命令は該構成プログラムを使用して実行される。もしそうでない場合は、上記命令用の構成プログラムがロードされ、次いで、この構成プログラムを使用して該命令が実行される。

【０００６】

同時には、限られた数の構成プログラムしかロードすることができない。新たな構成プログラムをロードする余裕がない場合は、他の構成可能な命令用の構成プログラムが当該バンドから削除され、該新たな構成プログラム用の余裕を作る。

【０００７】

構成プログラムがロードされる毎に、かなりのオーバーヘッドが存在する。前記文献によれば、このオーバーヘッドは、ロードされている構成プログラムを、他の構成プログラムをロードするために削除するまで、可能な限り長く維持することにより最小化される。このようにして、構成プログラムの一種のキャッシュが実現され、これが、構成プログラム

が繰り返し使用される場合はオーバーヘッドを最小化する。しかしながら、依然として、構成プログラムをロードするためのかなりのオーバーヘッドが存在する。

【０００８】

【発明の開示】

本発明の一つの目的は、なかでも、構成プログラムをロードするために必要とされるオーバーヘッドを低減することにある。本発明の他の目的は、一緒にロードされた状態に維持することができる構成プログラムの数を増加させ、これにより構成プログラムが少ない回数しかロードする必要がないようにすることにある。また、本発明の他の目的は、コンピュータプログラム用に必要とされる全ての構成プログラムを記憶するのに要するメモリの量を低減することにある。

【０００９】

本発明によるコンピュータプログラムを実行する方法の一実施例が請求項１に記載されている。この実施例によれば、構成可能な命令の組み合わせが定義され、個別にではなく、組み合わせとしてロードされる。当該プログラムを実行する前に、各々が少なくとも２つの構成可能な命令からなる１以上の組み合わせが選択される。典型的には、各組み合わせは、当該コンピュータプログラムの１以上の連続した命令の領域に関連付けられる。当該プログラムの該特定の領域が実行される場合、該領域に関連する組み合わせの全ての構成可能な命令用の構成プログラムがロードされる。

【００１０】

上記命令の組み合わせ及びそれらに関連する領域は、当該プログラムを実行する前に、構成プログラムをロードするためのオーバーヘッドが最小となるように、即ち、当該組み合わせのために選択される構成可能な命令が該組み合わせに属さない他の構成可能な命令により中断されることなく連続的に発生するように（もし、これらの他の構成可能な命令が他の組み合わせをロードする必要性を生じるであろうようなら）、選択することができる。このように、オーバーヘッドを最小化するためになされる作業は、実行時というよりはコンパイル時になされる。

【００１１】

更に、多くのコンピュータプログラムに関して、命令サイクルカウントは、オペランドにおける同一位置からのビットの使用又は類似しているが僅かに異なる論理関数の計算のような強い類似性を持つ命令の組み合わせを用いて最小化することができる。これらの命令は、当該組み合わせにおける全ての命令により共通に使用されるハードウェア資源と、個々の命令（又は、これら命令の部分集合）に特有の幾つかのハードウェア資源とをプラスしたものを用いて実現することができる。このように、上記組み合わせにおいてロードすることができる命令の数が増加される。

【００１２】

本発明による方法の他の実施例によれば、上記命令の組み合わせ用の構成プログラムは、該プログラムが再構成可能な機能ユニットにおける上記組み合わせ内の異なる命令の再構成可能なハードウェア資源の使用を相互最小化（cross-minimize）するように選択される。幾つかの機能に対する資源使用の相互最小化とは、資源使用が各機能に関して独立に最小化されるのではなく、全ての機能を実行する全ての構成プログラムの設計空間において極小が追求されることを意味する。上記組み合わせにおける異なる命令の間の相互最小化の結果として、資源使用が各命令個別に対して最小化された場合に上記組み合わせに対して必要とされたであろうよりも、少ないハードウェア資源しか必要とされない。

【００１３】

構成可能な機能ユニットにおけるハードウェア資源の例は、回路素子及びプログラム可能な接続部である。典型的な構成可能な機能ユニットは、オン又はオフとなるように構成することができると共に回路素子を互いに、他の回路型式の回路素子に又は当該機能ユニットの入力端子若しくは出力端子に接続するような接続部を備える多数の同一の回路素子を含んでいる。典型的には、限られた数の斯様な接続部のみを構成することが可能である。例えば、幾つかの回路素子のみが、直接に入力端子若しくは出力端子、又は所与の他の回

路素子に接続することができる。
【００１４】
構成プログラムが異なる命令に対して独立に選択されるとすると、或る組み合わせにおける或る命令用の各構成プログラムは、それらハードウェア資源がたとえ実際に使用されないであろう場合でも、該組み合わせ内の他の命令用の他の構成プログラムにより使用するために、これらハードウェア資源を空き状態にしておかなければならない。相互最小化により、１つの命令用の構成プログラムは、他の命令のために使用されない如何なるハードウェアも使用することができる。
【００１５】
更に悪いことに、当該組み合わせにおける他の命令を考慮しない、或る命令用の構成プログラムに使用するための回路素子の選択は、この選択が当該接続部を入力端子又は出力端子に固定するような場合、追加の資源使用の原因となる。これは、他の命令用の入力端子／出力端子接続部を最適に選択することによるハードウェア資源使用を最小化する可能性を取り除くことになる。
【００１６】
同一の組み合わせにおける異なる命令のハードウェア資源の使用を相互最小化することにより、斯様なハードウェア資源の浪費を防止することができる。更に、異なる命令の間で、共通のハードウェア資源を共用することが可能になる。ハードウェア使用を相互最小化することにより、共通のハードウェア資源が当該命令の組み合わせに対して２回以上割り付けられねばならないということが防止される。
【００１７】
本発明による方法の他の実施例によれば、当該組み合わせ内の異なる命令を選択するための、及び上記異なる命令によるオペランドデータを処理するためのハードウェア資源使用が、相互最小化される。通常、命令選択は、演算コードのオペランドデータ処理回路を可能化する信号への解読を含んでいる。該実施例においては、命令及びオペランドデータ処理用の構成プログラムのハードウェア資源使用が互いに独立に最小化された場合よりも、少ない数のハードウェア資源しか命令選択及びオペランドデータ処理用として必要とされない。
【００１８】
好ましくは、当該プロセッサはパイルライン化される。このことは、命令処理が、命令解読とオペランドフェッチとをプラスした段階、命令実行段階及び結果の書き戻し段階等の順次の段階に分割されることを意味する。パイプライン化されたプロセッサにおいては、順次の命令の命令処理の異なる段階が互いに並列に実行される。命令処理の構成可能な部分は、上記実行段階で生じる。本発明の一実施例によれば、オペランドデータ処理と、異なる命令の間を区別する命令選択ビットの使用との両方が、該構成可能な命令の処理の実行段階で生じる。
【００１９】
本発明による方法の他の実施例によれば、上記の再構成可能な機能ユニットは再構成可能な交点スイッチ（cross-point switch）を、オペランドデータ用の入力端子と、該交点スイッチの各出力端子を異なる論理合成回路へ接続する接続ラインとの間に含む。
【００２０】
本発明の、これら及び他の特徴は添付図面を参照して説明されるであろう。
【００２１】
【発明を実施するための最良の形態】
図１は、構成可能な命令をサポートするプロセッサのアーキテクチャを示している。本発明に影響を与えない該プロセッサアーキテクチャの種々の特徴は、明瞭化のために図１から削除されていることに注意されたい。例示として、パイプライン化されたＲＩＳＣアーキテクチャが考察されるが、本発明は斯様なアーキテクチャに限定されるものではない。例えば、代わりにＣＩＳＣアーキテクチャ又はＤＳＰアーキテクチャを使用することもできる。該実施例は、異なるパイプライン段を分離する３つのレジスタ１０、１４及び１９

を備えるパイプライン化されたプロセッサを示している。命令レジスタ１０は、当該パイプラインの源に位置する。この命令レジスタ１０のオペランド参照フィールド出力端子は、レジスタファイル１２の入力端子に接続されている（例として、これらのフィールドは５ビット幅とする）。このレジスタファイル１２の出力端子（例えば、ビット幅ｗ＝３２）は、上記命令レジスタ１０の他の出力端子と共に、実行段レジスタ１４に結合されている。上記他の出力端子は、結果アドレス出力端子（例として、５ビット幅）及び構成可能な命令の選択コード用出力端子（例として、１１ビット幅）を含んでいる。

【００２２】

レジスタファイル１２からのデータを通過させる実行段レジスタ１４の出力端子は、ＡＬＵ機能ユニット１６と構成可能な機能ユニット１８とに並列に結合されている。ＡＬＵ機能ユニット１６はＡＬＵ160を含み、該ＡＬＵの入力端子はマルチプレクサ162及び164に各々結合されている。各マルチプレクサ162、164は実行段レジスタ１４の出力端子に接続されている。マルチプレクサ162及び164は、プログラムカウンタ値及び上記命令レジスタからの中間値を各々入力する入力端子を更に有している（これらの入力端子用の接続は図示されていない）。

【００２３】

上記実行段レジスタ１４の他の出力端子は前記の構成可能な命令の選択コードを構成可能な機能ユニット１８に受け渡し、結果アドレスは書き戻し段レジスタ１９に渡される。ＡＬＵ機能ユニット１６及び構成可能な機能ユニット１８の出力端子は、書き戻し段レジスタ１９に接続されている。上記実行段レジスタはレジスタファイル１２に対し、ＡＬＵ機能ユニット１６又は構成可能な機能ユニット１８の結果をレジスタファイル１２における結果オペランドアドレスにより示されるロケーションに書き込むための接続部（図示略）を有している。

【００２４】

他の機能ユニット（例えば、メモリアクセスユニット）、制御ライン、分岐回路、命令解読回路及びレジスタ１４、１９等に対する入力を選択するためのマルチプレクサのような種々の回路は、明瞭化のために図１から削除されている。

【００２５】

動作時に、図１のアーキテクチャは、パイプライン化された実行機構を達成する。順次の命令は、順次のクロックサイクルにおいて命令レジスタ１０にロードされる。或る命令がロードされた後のクロックサイクルにおいて、そのオペランド参照が、レジスタファイル１２からオペランドをロードするために使用される。このクロックサイクルは、例えば機能ユニット１６、１８（又は図示せぬ他のもの）の何れが当該命令を実行するかの選択のような、命令解読を含むことができる。このサイクルの終了時に、オペランド、結果オペランドアドレス及び構成可能な命令の選択コードが、実行に要する全ての他のデータ（図示略）と共に、実行段レジスタ１４にロードされる。次のクロックサイクルにおいては、この情報は機能ユニット１６、１８（及び／又は図示せぬ他の機能ユニット）に渡され、結果を得るために処理される。このクロックサイクルの終了時に、選択された機能ユニット１６、１８からの結果及び結果オペランドアドレスが、書き戻し段レジスタ１９にロードされる。この次のサイクルの後のクロックサイクルにおいて、上記結果がレジスタファイル１２に書き戻される。

【００２６】

当該命令の演算コードフィールドが構成可能な命令を選択した場合は、構成可能な機能実行ユニット１８が該命令を実行して結果を生成するために選択される。この場合、当該命令における構成可能な命令の選択コードは、どの特定の構成可能な命令が実行されるかを決定するために使用される。

【００２７】

勿論、本発明から逸脱することなく、各々が異なる命令の組み合わせを用いて構成された２以上の構成可能な機能ユニットを並列に設け、所要の構成プログラムの交換のためのオーバーヘッドなしで、２以上の組み合わせを同時に利用可能にすることもできる。

【００２８】
図２は、構成可能な機能ユニットの一実施例を示している。これは、基本的にそれ自体既知のＣＰＬＤ（複合プログラマブル論理装置）コアである。該構成可能な機能ユニットは、前記オペランドの各々のｗビット及び上記構成可能な命令の選択コードのＮビット（例として、Ｎ＝４）を入力するための入力ポート２０a、２０b及び２２を有している。これら入力ポートは交点スイッチ２４の入力端子に接続されている。この交点スイッチ２４は多数の出力端子を有している。該交点スイッチ２４は、その２＊ｗ＋Ｎ個の入力端子の各々が、当該構成可能な機能ユニットにロードされる構成プログラムの制御の下で上記出力端子のうちの何れかに接続することができるように設計されている。
【００２９】
交点スイッチ２４の出力端子は論理ブロック２６a及び２６bの各々に接続されている。これら論理ブロック２６a及び２６bの出力端子は、当該構成可能な機能ユニットの出力ポート２８に結合されている。例示として、各々が３６個の入力端子とｗ／２（例えば、１６）個の出力端子を備える２つの論理ブロック２６a及び２６bが示されている。２つの論理ブロック２６a、２６bのｗ／２ビットの出力は一緒にｗビットの結果出力を形成する。
【００３０】
該構成可能な機能ユニットは、再構成制御回路２３を含んでいる。この再構成制御回路２３は、前記構成可能な命令の選択コードのうちの交点スイッチ２４に供給されなかったビットを入力する入力端子を含んでいる。該再構成制御回路２３は、交点スイッチ２４及び論理ブロック２６a、２６bに接続された出力端子を有している。
【００３１】
動作時において、制御回路２３は、上記命令選択コードの入力されたビットを、当該構成可能な機能ユニットが現在実行するようにプログラムされている再構成可能な命令の組み合わせの対応するビットと比較する。好ましくは、上記命令選択コードのビットの部分集合が当該組み合わせを示すために使用され、残りのビットが該組み合わせ内の構成可能な命令を示すようにする。上記選択コードが、異なる組み合わせからの命令が実行されるべきであることを示している場合は、上記再構成制御回路２３は、メモリ（図示略）から該新たな組み合わせの全命令用の構成プログラムをロードし、交点スイッチ２４及び論理ブロック２６a、２６bを該新たな組み合わせが要するように再プログラムする。その後、該新たな組み合わせからの命令は実行することができる。
【００３２】
新たな組み合わせが一旦ロードされるか、又は既にロードされていた組み合わせからの命令が選択されると、該構成可能な機能ユニットは該命令を処理する。この場合、前記交点スイッチに供給されるＮ個の命令ビットが、どの命令（当該構成可能な機能ユニットにロードされた命令の組み合わせからの）に従いオペランドが処理されるかを決定する。
【００３３】
構成プログラムのロードの最も容易な実施は、該構成プログラムがロードされるまで当該プロセッサによる更なる命令の実行を停止（stall）させることである。しかしながら、もっと少ない命令サイクルオーバーヘッドしか要さない他の構成を使用することもできる。例えば、当該構成プログラムのロードを起動するために、先駆（precursor）命令を使用することもできる。該先駆命令は、上記組み合わせからの構成可能な命令が使用されるであろうことを示す。該先駆命令自体は構成プログラムを必要としないが、指示された構成プログラムのロードを起動する。
【００３４】
他の例では、当該プロセッサは通常の（例えば、ＡＬＵ）命令のサブルーチンに跳び、構成プログラムがロードされなければならない又はロードされつつある場合は、これら命令が上記構成可能な命令を実行する。これは、アドレス指定されたオペランドレジスタの内容をサブルーチン呼び出しスタックに配置し、該サブルーチンを呼び出し、該サブルーチンからの帰還後に上記の構成可能な命令のアドレス指定された結果レジスタに上記呼び出しスタックからの結果を戻すことにより達成することができる。

【００３５】
図３は、図２の構成可能な機能ユニットに使用する論理ブロックの一例の実施例を示している（それ自体は、ＣＰＬＤから既知である）。この論理ブロックはＰＡＬアレイ３０とＰＬＡアレイ３２とを含み、両者は前記交点スイッチ２４の全出力端子に結合されている。アレイ３０及び３２の出力端子には、アンドゲート３４a～３４b及び３５a～３５cが記号的に図示されている。例示として、ＰＡＬアレイ３０に対しては６４個のアンドゲート３４a～３４bが存在し、ＰＬＡアレイ３２に対しては３２個のアンドゲート３５a～３５cが存在する。
【００３６】
アレイ３０、３２は列導体及び行導体（図示略）から構成され、各列は交点スイッチ２４の各出力端子に対応し（好ましくは、各列は各信号と、それらの否定の両者に対して存在するようにする）、各行は当該論理ブロックのアンドゲート３４a～３４b及び３５a～３５cに対応している。行と列との交点には、トランジスタとメモリセルとが存在する（図示略）。メモリセルはトランジスタが駆動されるか否かを制御する。トランジスタは、駆動されると、アンドゲート３４a～３４b及び３５a～３５bの入力を形成し、これらアンドゲート３４a～３４b及び３５a～２５cは、それらのトランジスタが駆動された列導体の論理レベルのアンドを出力する。
【００３７】
上記ＰＬＡアレイのアンドゲート３５a～３５cの出力端子は、マトリクス３３の行導体に接続されている。このマトリクスの列導体はオアゲート３６a～３６cに接続されるように図示されている。行及びの交点には、トランジスタとメモリセル（図示略）が存在する。メモリセルはトランジスタが駆動されるか否かを制御する。トランジスタは、駆動されると、オアゲート３６a～３６cの入力を形成し、これらオアゲート３６a～３６cは、それらのトランジスタが駆動された行導体の論理レベルのオアを出力する。
【００３８】
ＰＡＬアレイ３０の出力端子は、４つの群毎に、更なるオアゲート３８a～３８bに接続されている。各オアゲート３６a～３６cは、上記更なるオアゲート３８a～３８bの各々の入力端子に結合された出力端子を有している。例示として、１６個のオアゲート３６a～３６cと、１６個の更なるオアゲート３８a～３８bが存在する。更なるオアゲート３８a～３８bの各々は、プログラム可能なインバータ／非インバータ３９a～３９bを介して当該論理ブロックの出力ビットラインに接続されている。各インバータ／非インバータ３９a～３９bに対してメモリセル（図示略）が設けられ、これらメモリセルの内容が、これらインバータ／非インバータ３９a～３９bが反転するか否かを制御する。
【００３９】
上記論理ブロックの論理機能は、ＰＡＬマトリクス３０、ＰＬＡマトリクス３２、マトリクス３３及びインバータ／非インバータ３９a～３９b内の各交点におけるトランジスタの駆動を制御するメモリセルに、構成プログラムのビットをロードすることによりプログラムすることができる。
【００４０】
命令選択コードのＮビットは、前記交点スイッチ２４に供給される。これらＮビットは、前記オペランドのビットと同様の方法で供給される。当該構成可能な機能ユニットの構成プログラムは、これらＮビットをオペランドビットの何れかのように扱う自由度を有している。上記Ｎビットが命令の特定の１つを検出するために最初に合成され、データがどの様に処理されるかを制御するために該検出結果が使用される必要はない。反対に、上記Ｎビットの個々のビットは、当該Ｎビットの他のビットとは独立に、オペランドビットと共に論理機能に引数として参加することができる。
【００４１】
図５は、命令の或る組み合わせからの命令を実行するようにプログラムされた場合の、上記機能ユニットのハードウェア機能記述のモデルを示している。この記述に示される構造は機能的のみのものであって、物理的なものではないことに注意されたい。異なる機能ブ

ロックへの分割は、当該構成可能な機能ユニットにおいて実施化される回路の構造の如何なる分割にも対応する必要はなく、異なるブロックが当該構成可能な機能ユニットにおける同一の物理的回路素子を共用することもできる。

【００４２】

該モデルは、２つのソースオペランド用の入力部５０ａ及び５０ｂと、構成可能な命令の各々を実行するための多数のブロック５２ａ～５２ｃと、これら命令の結果用のポート５４ａ～５４ｃと、上記結果の１つを出力部５８に通過させるマルチプレクサ５６とを示している。マルチプレクサ５６は、前記命令選択コードのＮビットにより制御される。

【００４３】

このモデルは、その機能を果たすためになされなければならない接続のリストに変換される。この変換の間に、図５の種々のブロックの間で資源使用が相互最小化される。即ち、マルチプレクサ５６の機能はブロック５２ａ～５２ｃのものと（部分的に）併合することができ、これらブロック５２ａ～５２ｃの機能は互いに併合することができる。

【００４４】

構成可能な命令及び一緒にロードされる命令の組み合わせは、好ましくは、当該プロセッサ上で走る特定の各プログラムに対して独立に選択されるものとする。以下においては、これらの命令及び組み合わせは、各々、“カスタム命令”及び“クラスタ”とも呼ぶ。カスタム命令及びクラスタの選択は、好ましくは、コンピュータプログラムのコンパイルを用いて、即ち上記プロセッサが該コンピュータプログラムを実行する前に、なされる。

【００４５】

図４はコンパイルされたプログラムを発生するためのフローチャートを示している。図４のフローチャートは以下のステップを実行する。

４１：ソースコード（典型的には、Ｃで書かれる）がコンパイラ前置部により処理され、データフローグラフとして表された中間コードを発生する。

４２：上記中間コードはクラスタ検出／選択モジュールにより読み取られ、該モジュールはハードウェア合成に潜在的に適したデータフローセグメント（候補）を探す。各“候補”はカスタム命令を定義する。好ましくは、当該アプリケーションの臨界的経路内で純粋に算術的又は論理的演算からなるセグメントのみが考慮されるものとする。上記検索を案内するためにプロファイルデータが使用される。候補は、或る評価規準に従いカスタム命令のクラスタにグループ化される（以下の説明を参照）。

４３：上記クラスタはトランスレータにより処理され、該トランスレータはデータフローセグメントの算術演算をＨＤＬ（標準ハードウェア定義言語）でのハードウェア記述に変換する。異なるカスタム命令が独立に実行することができるように、このハードウェア記述には解読ロジックが付加される。図５は、この段階で生成された回路記述のモデルの一例を示している。

４４：結果としての回路記述はハードウェア合成ツールにより処理され、そこでは、タイミング及び適合性レポート（所要の処理時間量及びハードウェアを記述している）が、回路ネットリストと共に発生される。この段階で、当該回路記述内のマルチプレクサの機能を、図５の機能ブロックの資源使用とで相互最小化することができる。この資源の相互最小化それ自体は、通常、機能を備えるプログラマブルロジックをプログラムするためとして既知である。

４５：タイミング及び適合性情報は前記クラスタ検出／選択モジュールに送り返され、そこでは、クラスタが再配置若しくは廃棄されるか、又は新たなクラスタが形成される。該サイクルは最終的なクラスタの集合が選択されるまで繰り返される。

４６：最終的なクラスタの集合が選択されたら、該最終的に選択されたクラスタで構成された中間コードのデータフローセグメントは、それらの等価なカスタム命令ラベルにより置換される。

４７：次いで、結果としてのコードは後置部により後処理（レジスタの割り付け、アセンブリコードの出力、命令のスケジューリング及び／又は当てはまる何らかの他の後処理のために）される。

４８：結果としてのアセンブリは、新たに合成されたカスタム命令ラベルを認識する修正されたアセンブラに送られる。前記ハードウェア合成ツールにより発生されたネットリストは、該アセンブリと組み合わされて、最終的な実行可能なものを生成する。

【００４６】

上記のハードウェア合成ステップは、人のプログラマからは完全に隠すことができる。他の例として、好ましくは上述した指針を用いて前記候補及び／又はクラスタを選択する人のプログラマを必要とするかもしれない。

【００４７】

プログラムのデータフローグラフから候補の構成可能な命令を選択すること自体は既知である。基本的に、これは、上記プログラムのデータフローグラフからサブグラフを選択することを含み、その場合、これらサブグラフは２以下の変数オペランド入力しか有さない。（これは、上記の構成可能な機能ユニットが２つのオペランド入力を有する場合である。該構成可能な機能ユニットが小さな数の又は大きな数のオペランド入力を有している場合は、対応して大きな又は小さな入力を有するサブグラフを選択することができる。）

【００４８】

好ましくは、候補は、当該プログラムにおける上記候補が発生する連続した命令の領域に基づいてクラスタにグループ化されるようにする。或る領域に対しては、同時にロードすることが可能な構成プログラムの数よりも多くのクラスタが定義されるべきではない（例えば、１つのみの構成可能な機能ユニットしか存在せず、該ユニットが一度に１つのみの構成プログラムを用いてしか構成することができない場合は、１つのみのクラスタ）。上記プログラム領域のサイズ及び該領域用に選択される候補の数は、全ての候補を同時にロードされる最大数のクラスタにおいてプログラムすることができるように、選択されねばならない。

【００４９】

原理的には、候補命令の選択及びそれらのクラスタへの合成の両者は、厄介な最適化問題である。多数の構成可能な命令の可能な組み合わせの集合が存在する。狙いは、プログラムの典型的な実行（実行プロファイルにより定義される）に必要とされる命令サイクルの数を最小化するような集合を見付けることである。命令サイクルのカウントの最小化は、或るクラスタに関する全ての選択された候補が構成プログラムに適合するという拘束を受ける。

【００５０】

この目的のため、通常の命令のみが使用されたとしたら一層多数の命令サイクルを必要としたであろうような、各々が１命令サイクルにおいて効果を生成する構成可能な命令を考慮する。当該プロファイルにおいてカスタム命令により置換されるべき全ての通常の命令を実行するのに要する追加の命令サイクルの数は、構成プログラムをロードするためのオーバーヘッドサイクルよりも大きくなくてはならない。さもなければ、何の組み合わせも選択されるべきでない。オーバーヘッドサイクルの数は、組み合わせをロードする場合の方が、同数の命令を個々にロードする場合よりも小さいことに注意すべきである。何故なら、ローディングは、上記組み合わせにおける全ての命令に対しては１回しか行う必要がなく、平均して、相互最小化により命令当たり少ない構成プログラム空間しか占めないからである。

【００５１】

クラスタに合成することができる候補の数は、構成可能な機能ユニットにおける資源使用の最小化に依存する。構成可能な命令が“小さい”程、又は一層高度に類似している程、より多くの構成可能な命令を１つのクラスタに組み合わせることができる。

【００５２】

クラスタを選択するアルゴリズムの一例は、

１）当該プログラムの範囲から領域を、コンパイラにより生成された中間コードにおける領域の開始命令及び終了命令に関して選択する。好ましくは、領域として、頻繁に実行されるループ又は頻繁に実行されるサブルーチンにおける命令を選択するものとするが、非

常に頻繁には実行されないが類似の命令の繰り返しを含む領域も良好な候補である。
２）上記の選択された領域における命令に対するデータフローセグメントに関して多数の候補のカスタム命令を選択する。
３）上記の選択されたカスタム命令が全て当該構成可能な機能ユニットに一緒に納まり、且つ、完了するのにパイプラインサイクル未満しか掛からないような形で、これら選択されたカスタム命令を組み合わせるクラスタに対する構成プログラムを発生することができるか判断する。もしそうなら、通常の命令の組み合わせを選択された候補のカスタム命令により置換することにより当該プロファイルにおいて得られる命令サイクルの数を決定する。
４）ステップ１ないし３を、一層大きな大きな及び一層小さな領域に対して、且つ、別の選択された命令に対して繰り返し、選択された領域及び選択された命令を保持し、及び最も多い命令サイクルを得るクラスタを保持する。このステップは、或る領域に対して見つかった最も有利なクラスタから始めて、同一の領域から又は該領域の拡張からの何れかから、上記クラスタを更なるカスタム命令を用いて拡張することにより発見的に高速化することができる。
５）上記ステップ１ないし４を、当該プログラムの異なる、重ならない領域に対して、これら全ての異なる領域に関する各クラスタを保持しながら繰り返す。

【００５３】

上記の最小化は、当該プログラムの上記領域に局部化された局部的処理であり、当該プログラム全体に対するものでないことに注意すべきである。重要な点は、クラスタが特定の領域に対する命令サイクルカウントを減少させることである。他の領域で何が起きるかは問題ではない。何故なら、当該クラスタは、それら領域に関してはロードする必要はないからである。事実、異なる領域用の異なるクラスタは同一の効果を伴う幾つかのカスタム命令を含むかもしれない。これらの同一の効果を持つカスタム命令の１つが実行されると、実行された領域が、上記クラスタの何れがロードされたかを決定する。従って、ロードされる構成プログラムは、実行されねばならない如何なる特定の命令によるというよりも、実行されている領域により決定される。

【００５４】

クラスタ及び領域の選択は、これらの点を選択するための多数の発見的評価規準を用いて単純化することができる。種々の評価規準を、候補をクラスタにグループ化するために使用することができる。例えば、
－　共通のループ内の候補は、同一のクラスタにグループ化される（これは、ループ内の再構成オーバーヘッドを防止する）。
－　サブルーチン内で発生する候補は、当該サブルーチン用のクラスタ又は複数のクラスタにグループ化される。
－　低い論理的複雑さの候補は、大きなクラスタ（一層多くの候補を持つ）にグループ化される。
－　高い論理的複雑さの候補は、小さなクラスタ（一層少ない候補を持つ）にグループ化される。
－　一層良好に論理的相互最小化の機会を利用するために、類似の候補（論理の点で）の同一のクラスタへの合成が好まれる。

【００５５】

実際に効果的に働くことが分かったクラスタ選択用の評価規準は、カスタム命令を、該カスタム命令の結果に影響を与えるような、それらのオペランドの入力ビットの類似性に依存してクラスタに配置することである。或るカスタム命令が与えられた場合、その入力オペランドのどのビットが、該命令の結果に影響を与えるか、及びどのビットが該結果に影響を与えないかを決定するのは回りくどくない。この場合、カスタム命令の非類似性は、共用されない入力ビットの数により測ることができる。従って、カスタム命令のクラスタは、好ましくは、各命令の入力のどのビットが該命令の結果に影響するかを計算し（これらは“関連ビット”と呼ばれる）、命令間の非類似性の目安を共用されない関連ビットの

数に関して計算し、且つ、設定された量より少ない非類似性を持つカスタム命令のクラスタを選択することにより選択される。

【００５６】

ＣＰＬＤコアの規則的な予測可能なタイミングモデルは、クラスタの形成に有利である。複数候補の単一構成へのグループ化は、典型的なＦＰＧＡ構造における導入遅延をかなり変化させ得、自動的なクラスタ形成用のアルゴリズムをかなり困難にする。何故なら、その場合には、相互最小化を、該最小化が過度に多い遅延を導入しないという制限の下で実施しなければならないからである。ＣＰＬＤの場合は、或る構成への一層多くのカスタム命令の追加は、前記ＰＡＬ又はＰＬＡからの一層多くの積項（ＰＴ）を単に必要とする。回路が当該コアに適合する限り、上記遅延は前記交点スイッチ及びＰＬＡを介しての時間遅延（Tpd_pla）に限定され、当該アルゴリズムはクラスタを形成する際に遅延変動の問題を考慮する必要がない。

【００５７】

図２の機能ユニットにおいては、交点スイッチ２４が特に有効である。何故なら、該交点スイッチは、オペランドの異なるビットに関する信号を、これら異なるビットが該オペランド内の非常に異なる位置に不規則に広がっている場合でさえも、１つの論理ブロック２６ａ、２６ｂに一緒にもたらすことを可能にするからである。これは、同一の効果がＡＬＵ１６を用いて実現されなければならなかった場合に、非常に多くの通常の命令を必要としたであろう構成可能な命令を実施化することを可能にするからである。

【００５８】

また、交点スイッチ２４は、命令選択ビットを異なるブロック２６ａ、２６ｂにおいてオペランドデータビットと自由に混合することを可能にする。このようにして、命令の区別をオペランドの処理と統合することにより、ハードウェア資源使用の一層良好な相互最小化が可能となる。

【図面の簡単な説明】

【図１】　図１は、構成可能な命令をサポートするプロセッサのアーキテクチャを示す。

【図２】　図２は、構成可能な機能ユニットを示す。

【図３】　図３は、構成可能な論理ブロックを示す。

【図４】　図４は、コンパイルされたプログラムを発生させるためのフローチャートを示す。

【図５】　図５は、命令の組み合わせを実行する機能ユニットのモデルを示す。

【符号の説明】

１４…実行段レジスタ
１６…ＡＬＵ機能ユニット
１８…構成可能な機能ユニット
２０ａ、２０ｂ、２２…入力ポート
２３…再構成制御回路
２４…交点スイッチ
２６ａ、２６ｂ…論理ブロック
２８…出力ポート

【図１】

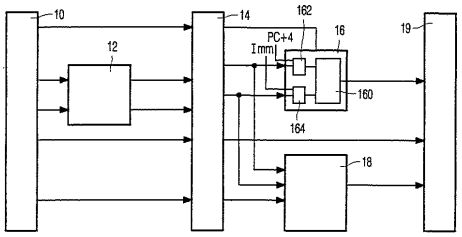

FIG. 1

【図２】

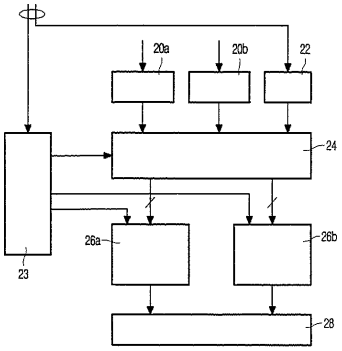

FIG. 2

【図３】

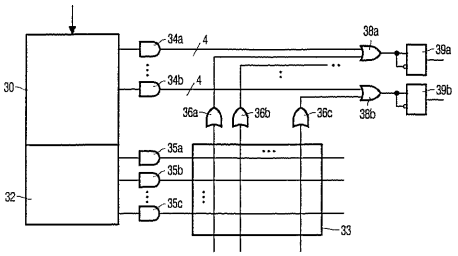

FIG. 3

【図４】

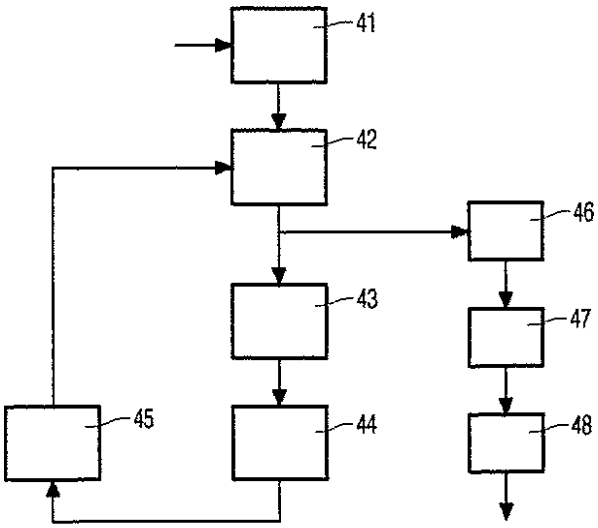

FIG. 4

【図５】

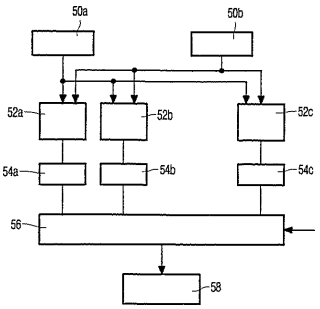

FIG. 5

フロントページの続き

(72)発明者 ビンク アドリアヌス ジェイ
オランダ国 ５６５６ アーアー アインドーフェン プロフ ホルストラーン ６
(72)発明者 フーゲルブルージュ ジャン
オランダ国 ５６５６ アーアー アインドーフェン プロフ ホルストラーン ６

審査官 三坂 敏夫

(56)参考文献 特開平１１－０２４８９１（ＪＰ，Ａ）
特開平１０－３２０２１４（ＪＰ，Ａ）
特開平０４－２８８６８０（ＪＰ，Ａ）
特開平０４－２１３１６７（ＪＰ，Ａ）
特開平１１－０８５５０７（ＪＰ，Ａ）
特開平０８－２８６９０８（ＪＰ，Ａ）
特開平０４－２１９８７５（ＪＰ，Ａ）
特開平０５－２３３７５５（ＪＰ，Ａ）
米国特許第０５１２８８７１（ＵＳ，Ａ）
Wittig et al., OneChip: an FPGA processor with reconfigurable logic, IEEE Symposium on FPGAs for Custom Computing Machines, 1996. Proceedings. , 米国, IEEE, １９９６年 ４月１９日, pp.126-135, ISBN: 0-8186-7548-9

(58)調査した分野(Int.Cl., ＤＢ名)
G06F 9/318
G06F 9/45
G06F 17/50