

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5175517号  
(P5175517)

(45) 発行日 平成25年4月3日(2013.4.3)

(24) 登録日 平成25年1月11日(2013.1.11)

(51) Int.Cl.		F I			
<b>G06F</b>	<b>9/38</b>	<b>(2006.01)</b>	<b>G06F</b>	<b>9/38</b>	<b>370X</b>
<b>G06F</b>	<b>9/46</b>	<b>(2006.01)</b>	<b>G06F</b>	<b>9/46</b>	<b>410</b>
<b>G06F</b>	<b>15/80</b>	<b>(2006.01)</b>	<b>G06F</b>	<b>15/80</b>	

請求項の数 5 (全 19 頁)

(21) 出願番号	特願2007-263941 (P2007-263941)	(73) 特許権者	000005821
(22) 出願日	平成19年10月10日(2007.10.10)		パナソニック株式会社
(62) 分割の表示	特願2007-513029 (P2007-513029) の分割		大阪府門真市大字門真1006番地
原出願日	平成18年4月12日(2006.4.12)	(74) 代理人	100090446
(65) 公開番号	特開2008-52750 (P2008-52750A)		弁理士 中島 司朗
(43) 公開日	平成20年3月6日(2008.3.6)	(74) 代理人	100072442
審査請求日	平成21年3月31日(2009.3.31)		弁理士 松村 修治
(31) 優先権主張番号	特願2005-114133 (P2005-114133)	(74) 代理人	100125597
(32) 優先日	平成17年4月12日(2005.4.12)		弁理士 小林 国人
(33) 優先権主張国	日本国(JP)	(72) 発明者	森下 広之
(31) 優先権主張番号	特願2005-309352 (P2005-309352)		大阪府門真市大字門真1006番地 松下
(32) 優先日	平成17年10月25日(2005.10.25)	(72) 発明者	橋本 隆
(33) 優先権主張国	日本国(JP)		大阪府門真市大字門真1006番地 松下
			電器産業株式会社内
			電器産業株式会社内

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】

複数の命令から成るプログラムを実行するプロセッサであって、  
再構成可能な集積回路と、  
回路構成情報に基づいて、前記集積回路の一部を再構成する再構成手段と、  
複数の命令各々に対応する回路構成情報を記憶する構成情報記憶手段と、  
前記回路構成情報に基づいて前記集積回路を同時に再構成可能な2以上の命令を選択する  
選択手段と、  
前記選択手段で選択された2以上の命令に対応する回路構成情報に基づき再構成された  
集積回路を用いて、当該2以上の命令を並列実行する実行手段と  
を備えることを特徴とするプロセッサ。

【請求項2】

命令記憶部からオペコードとオペランドを含む命令コードを読み込む命令フェッチ部と、  
読み込んだ命令コードをデコードする命令デコード部と、  
再構成可能演算器を含む命令実行部と、  
前記命令デコード部のデコード結果に従って、前記命令実行部の動作を制御する制御部  
と、  
オペコードの種別毎に、当該オペコードが示す処理の実行に必要な回路を構成するための  
回路構成情報を記憶した構成情報記憶部と、

を備え、

前記命令デコード部は、デコード結果に従って再構成を行うように前記再構成可能演算器に指示し、

前記再構成可能演算器は、前記命令デコード部からの指示に従って、デコードしたオペコードに対応する回路構成情報を用いて再構成を行い、

前記制御部は、デコードした命令コードを再構成済みの再構成可能演算器に発行することで、オペランドが示すデータに対して、オペコードが示す処理を実行するよう制御することを特徴とするプロセッサ。

【請求項 3】

前記命令デコード部は、前記回路構成情報に基づいて前記再構成可能演算器に同時に再構成可能な 2 以上の命令を選択し、

前記再構成可能演算器は、選択された 2 以上の命令に対応する回路構成情報を用いて再構成を行い、

前記制御部は、再構成済みの 2 以上の回路を用いて、当該 2 以上の命令を実行するよう制御する

請求項 2 記載のプロセッサ。

【請求項 4】

前記命令実行部は、固定機能演算器を更に含み、

前記制御部は、デコードしたオペコードに対応する回路構成情報が前記構成情報記憶部に存在しない場合には、前記固定機能演算器を用いて命令を実行するよう制御する

請求項 2 記載のプロセッサ。

【請求項 5】

前記オペランドは、オペコードと回路構成情報の対応付けを示すコードを含む請求項 2 記載のプロセッサ。

【発明の詳細な説明】

【技術分野】

【0001】

本発明はプロセッサに関するものであり、特に、再構成可能な集積回路を有するプロセッサに関する。

【背景技術】

【0002】

近年のプロセッサ、例えば、デジタル信号を用いた映像、音響機器に搭載されるプロセッサは、複数の処理に対応する必要がある。

映像を圧縮する場合を見てみると、その圧縮方法として M P E G (Moving Picture Experts Group) 2、M P E G 4、H . 2 6 3、H . 2 6 4 など、多数の規格が実用化されている。

【0003】

従って、ユーザの利便性などを考慮し、近年の映像、音響機器は、1 機器で、これら複数の規格への対応など、複数の機能の実現が求められる。

この要求に答える為には、一つの処理を行うハードウェアを複数搭載することで複数の処理を実現するか、又は、ハードウェアは一つ搭載して、ソフトウェアで複数の処理を実行するかの方法が考えられる。

【0004】

前者の方法は、高性能を実現できるという利点があるものの、実現すべき機能が多数になった場合、回路規模が大きくなるという欠点がある。更に、新しい機能を追加する場合などには、ハードウェアの追加が必要となる。

一方、後者の方法は、ソフトウェアの追加、変更等により、複数の機能の実現、追加などを柔軟に行うことができるという利点があるが、性能を上げることが困難であるという欠点がある。

【0005】

10

20

30

40

50

そこで、均質な回路構成の一部に特定の処理に適した回路を組み込み、動的にハードウェア構成を変更することにより、特定の処理に関して、柔軟且つ高性能な処理を実現する再構成可能なハードウェアが提案されている（特許文献1参照）。

【特許文献1】国際公開第2002/095946号パンフレット

【発明の開示】

【発明が解決しようとする課題】

【0006】

しかし、このような再構成可能なハードウェアは、回路機能を実装する部分のほかに、配線部分やスイッチなども必要であり、回路規模が大きくならざるを得ず、また、再構成のためには時間を必要とする。

10

そこで、本発明は、回路規模を抑えつつ、柔軟且つ高性能なプロセッサの提供を目的とする。

【課題を解決するための手段】

【0007】

上記課題を解決する為に、複数のスレッドを巡回的に、そのスレッドに割当てられた時間ずつ実行するプロセッサであって、再構成可能な集積回路と、回路構成情報に基づいて、前記集積回路の一部分を再構成する再構成手段と、複数のスレッド各々に対応する回路構成情報を記憶する構成情報記憶手段と、順次、スレッドに対応する回路構成情報に基づき再構成された集積回路を用いて、当該スレッドを実行させる制御手段と、前記制御手段があるスレッドを実行させている間に、次に実行するスレッドを選択する選択手段とを備えることを特徴とする。

20

【発明の効果】

【0008】

本発明に係るプロセッサは、上述の構成を備えることにより、スレッドごとに回路を再構成できるので、そのスレッドに適した回路を用いて実行することができるようになる。

また、前記制御手段は、更に、スレッドを実行させている間に、実行させているスレッドが使用している前記集積回路の部分以外の部分について、前記再構成手段に前記選択手段で選択されたスレッドに対応する回路構成情報に基づいた再構成をさせることとしてもよい。

【0009】

30

これにより、スレッドを実行中に、次のスレッドの為の再構成を行うことができるようになるので、再構成に要する時間が不要となり、柔軟且つ高性能なプロセッサを実現できるようになる。

また、前記プロセッサは、更に、演算器を含み、前記制御手段は、前記演算器と前記再構成された集積回路とを用いて、当該スレッドを実行させることとしてもよい。

【0010】

これにより、通常の演算器と、再構成した集積回路を使用してスレッドを実行することができ、処理に応じて通常の演算器を使用したり、再構成した演算器を使用したり、又は、双方使用したりできるので、回路規模を抑えつつ、柔軟且つ高性能な処理が出来るようになる。例えば、再構成した集積回路を、特定処理用の演算器として使用するなどである。

40

【0011】

すなわち、全ての処理のために集積回路を再構成するのではないので、再構成可能な回路の規模を抑え、プロセッサ全体の規模を小さくすることが可能となり、且つ、処理に必要な回路は再構成できるので柔軟且つ高性能な処理が可能となる。

また、複数の命令から成るプログラムを実行するプロセッサであって、再構成可能な集積回路と、回路構成情報に基づいて、前記集積回路の一部分を再構成する再構成手段と、複数の命令各々に対応する回路構成情報を記憶する構成情報記憶手段と、前記回路構成情報に基づいて前記集積回路を同時に再構成可能な2以上の命令を選択する選択手段と、前記選択手段で選択された2以上の命令に対応する回路構成情報に基づき再構成された集積

50

回路を用いて、当該 2 以上の命令を並列実行する実行手段とを備えることを特徴とする。

【 0 0 1 2 】

本発明に係るプロセッサは、上述の構成を備えることにより、命令毎に集積回路を再構成でき、再構成する回路の規模などに応じて、複数の命令用の再構成を同時に行えるので、回路規模を抑えつつ、柔軟且つ高性能な処理が出来るようになる。再構成を同時に行うためには、命令の順序を考慮するだけでなく、ある命令に必要な集積回路の回路規模と、別の命令に必要な集積回路の回路規模とを合わせても、再構成可能な論理回路の回路規模を越えない必要がある。

【発明を実施するための最良の形態】

【 0 0 1 3 】

<実施形態 1 >

<概要>

本発明に係るプロセッサは、通常のプロセッサが備えている演算器のほかに、再構成可能なハードウェアを備え、処理を分担することで、回路規模を抑えつつ、高性能な処理を実現するものである。

【 0 0 1 4 】

すなわち、複数の処理であっても、処理の全てが異なっているわけではなく、共通する処理や命令、頻繁に使用される処理や命令があることに着目し、共通するものは通常の演算装置で行い、各処理に特有の処理は再構成可能なハードウェアによって構成された演算装置で行わせることで、プロセッサ全体としての回路規模を押さえつつ、高性能を維持するものである。

【 0 0 1 5 】

本実施形態のプロセッサは、マルチスレッドプロセッサであり、マルチスレッドを実現するための方法として、各タスクを一定時間ずつ順番に実行していくラウンドロビン方式を取るものとする。

各スレッドは、それぞれに固有の回路の方が適していたり、固有の回路を必要とする処理は、再構成可能なハードウェアで再構成した回路を使用して実行する。

【 0 0 1 6 】

すなわち、本プロセッサは、スレッドそれぞれに固有の回路を備える必要が無い場合、全体の回路規模を抑えることが可能となる。

但し、再構成には時間を要するため、その時間を削減することがポイントとなる。

以下、本発明の実施形態のマルチスレッドプロセッサについて説明する。

<構成>

以下、図 1 を用いて、本発明にかかるプロセッサ 1 0 0 0 の構成を説明する。

【 0 0 1 7 】

図 1 は、プロセッサ 1 0 0 0 の構成例を表す図である。

プロセッサ 1 0 0 0 は、マルチスレッドプロセッサ 1 1 0 0、再構成可能演算器 1 2 0 0、構成情報記憶部 1 3 0 0 及び再構成制御部 1 4 0 0 で構成される。

マルチスレッドプロセッサ 1 1 0 0 は、通常のプロセッサであり、複数の異なる処理を時分割で実行することができる、いわゆるマルチスレッドプロセッサである。

【 0 0 1 8 】

マルチスレッドプロセッサ 1 1 0 0 は、通常演算器である固定機能演算器 1 1 2 0 とスレッドスケジューラ部 1 1 1 0 とを有し、スレッドスケジューラ部 1 1 1 0 は、次に実行するスレッドを決定するなど、スレッドのスケジューリング機能を有する。

このスレッドスケジューラ部 1 1 1 0 は、次に実行するスレッドを決定して、レジスタの退避・復帰などの準備を行う他、本発明に特有の処理も行う。

【 0 0 1 9 】

具体的には、スレッド実行中に、次に実行するスレッドを選択し、そのスレッドを再構成制御部 1 4 0 0 に通知するなどである。

このマルチスレッドプロセッサ 1 1 0 0 は、内部の固定機能演算器 1 1 2 0 と外部の再

10

20

30

40

50

構成可能演算器 1 2 0 0 の双方と、必要に応じて演算データのやり取りを行いながら処理を実行していく。

【 0 0 2 0 】

次に、再構成可能演算器 1 2 0 0 は、組合せ回路や順序回路を実現できる論理ブロックと、論理ブロック間の配線部分から成る。論理ブロックは、ルックアップテーブルとフリップフロップを含む回路ユニットであり、ルックアップテーブルの設定値を変えることで、所望の論理回路を実現する。また、配線部分には、トランジスタスイッチなどが配され、配線経路が自由に設定可能となっている。

【 0 0 2 1 】

本実施形態では、論理ブロックはすべて同一の構成であるとし、個々に機能を変更でき、それらを組み替え可能な配線群で接続することで、各種機能の回路を実現する。

また、再構成可能演算器 1 2 0 0 は、同一な構成の 1 0 個のエリアに分割されているものとする。これらエリアはそれぞれ独立して再構成が可能であり、エリア間は組み替え可能な配線が引かれ、複数エリアで一つの回路を実現できるものとする。

【 0 0 2 2 】

構成情報記憶部 1 3 0 0 は、再構成可能演算器 1 2 0 0 を所望の回路に再構成する為の構成情報を記憶する機能を有する。構成情報は、所望の回路の個数分あるものとする。

この構成情報は、論理ブロックのルックアップテーブルの設定値や、配線経路を設定する為の各トランジスタスイッチへの制御信号の情報を含むものである。

また、構成情報記憶部 1 3 0 0 は、構成情報以外に、後で説明するスレッド情報テーブル 1 4 1 0 を記憶しておく機能を有する。このテーブルは、スレッドとそのスレッドで使用する構成情報を対応付けているものである。

【 0 0 2 3 】

再構成制御部 1 4 0 0 は、スレッド実行中にスレッドスケジューラ部 1 1 1 0 から次のスレッドの通知を受けて、通常の固定機能演算器 1 1 2 0 のみで実行可能なのか、再構成可能演算器 1 2 0 0 を必要とするのかを判断し、再構成可能演算器 1 2 0 0 を再構成する場合には、再構成可能演算器 1 2 0 0 と構成情報記憶部 1 3 0 0 に指示を出す機能を有する。

【 0 0 2 4 】

再構成可能演算器 1 2 0 0 に対して再構成を行う旨と再構成を行うエリアとを通知し、構成情報記憶部 1 3 0 0 には、構成情報を指定して、その構成情報を再構成可能演算器 1 2 0 0 に供給するよう指示する。

また、再構成ができない場合には、その旨をスレッドスケジューラ部 1 1 1 0 に返す機能を有する。再構成ができない場合とは、再構成可能演算器 1 2 0 0 に、再構成できるエリアが無い場合である。

【 0 0 2 5 】

< 動作 >

次に、図 2 ~ 図 4 を用いて、本発明に係るプロセッサの動作を説明する。

図 2 を用いて、実行順通りにスレッドを実行する場合の例を説明し、図 3 を用いて、スレッドの実行順を変える場合の例を説明する。

最後に、図 4 を用いて、スレッドの制御処理をフローチャートを用いて説明する。

【 0 0 2 6 】

< 実行順通りにスレッドを実行する場合 >

図 2 ( a ) は、スレッド情報テーブル 1 4 1 0 の構成例及び内容例を示す図であり、図 2 ( b ) は、スレッドの実行例を示すタイムチャートである。

図 2 ( a ) で示すスレッド情報テーブル 1 4 1 0 のスレッドを実行した場合のタイムチャートが、図 2 ( b ) で示すタイムチャートである。

【 0 0 2 7 】

まず、図 2 ( a ) のスレッド情報テーブル 1 4 1 0 を説明する。

このスレッド情報テーブル 1 4 1 0 は、構成情報記憶部 1 3 0 0 に記憶されているもの

10

20

30

40

50

である。

スレッド情報テーブル 1 4 1 0 は、スレッド名 1 4 1 1、構成情報 1 4 1 2 及び使用エリア数 1 4 1 3 で構成される。

【 0 0 2 8 】

スレッド名 1 4 1 1 は、スレッドの識別子である。以下、「 T H 0 」 ~ 「 T H 3 」 で表される 4 つのスレッドが順に実行されるものとして説明する。

構成情報 1 4 1 2 は、スレッド名 1 4 1 1 で表されるスレッドが、再構成可能演算器 1 2 0 0 を使用する場合の再構成の為の構成情報を示している。

使用エリア数 1 4 1 3 は、再構成可能演算器 1 2 0 0 を使用する場合に必要なエリア数を表す。

【 0 0 2 9 】

例えば、スレッド名 1 4 1 1 「 T H 0 」 のスレッドは、構成情報 1 4 1 2 「 構成 A 」 の構成で再構成された再構成可能演算器 1 2 0 0 を使用し、構成情報 1 4 1 2 「 構成 A 」 で再構成可能演算器 1 2 0 0 を再構成するには、使用エリア数 1 4 1 3 「 6 」 個のエリアが必要となる。また、スレッド名 1 4 1 1 「 T H 1 」 に対応する構成情報 1 4 1 2 「     」 は、再構成可能演算器 1 2 0 0 を使用しない旨を示し、当然に使用エリア数 1 4 1 3 「 0 」 個である。

【 0 0 3 0 】

次に、図 2 ( b ) を用いて、スレッドの実行例を説明する。

ここでは、スレッドが固定機能演算器 1 1 2 0 を使用しているタイムチャート 1 0 と、そのスレッドが、再構成可能演算器 1 2 0 0 を使用している場合の構成情報を表すタイムチャート 2 0、及び、そのスレッドが実行中に、再構成可能演算器 1 2 0 0 を再構成している構成情報を表すタイムチャート 3 0 を示している。また、時間を示す矢印の上方には構成情報を、下方には構成情報が必要とするエリア数を示す。尚、本図 2 ( b ) では、全てのスレッドは固定機能演算器 1 1 2 0 を使用する場合を示しているが、使用しない期間があってももちろん良い。

【 0 0 3 1 】

まず、スレッド名 「 T H 0 」、 「 T H 1 」、 「 T H 2 」、 「 T H 3 」 の順で、スレッドが実行され、それぞれのスレッドが実行中に、次のスレッドが使用する再構成可能演算器 1 2 0 0 を再構成しておくものとする。

このように事前に準備することで、再構成に要する時間を取る必要がなくなり、実質的にスレッドの実行時間のみとなる。

【 0 0 3 2 】

例えば、スレッド名 「 T H 1 」 のスレッド 1 0 0 が、固定機能演算器 1 1 2 0 のみを使用して、スレッドを実行している。

この間に、次に実行予定のスレッド名 1 4 1 1 「 T H 2 」 のスレッド 1 1 0 が使用する構成情報 1 4 1 2 「 構成 C 」 で再構成可能演算器 1 2 0 0 を再構成する。

この場合、再構成可能演算器 1 2 0 0 は全部で 1 0 個のエリアを備えていることから、使用している再構成可能演算器 1 2 0 0 のエリア数は、「 構成 C 」 に必要な 「 3 / 1 0 」 エリア 1 0 1 となる。

【 0 0 3 3 】

同様に、スレッド名 「 T H 2 」 のスレッド 1 1 0 が、固定機能演算器 1 1 2 0 と、再構成可能演算器の 「 構成 C 」 を使用して、スレッドを実行している。

この間に、次に実行予定のスレッド名 1 4 1 1 「 T H 3 」 のスレッドが使用する構成情報 1 4 1 2 「 構成 D 」 で再構成可能演算器 1 2 0 0 を再構成する。

この場合、使用している再構成可能演算器 1 2 0 0 のエリア数は、「 構成 C 」 が使用している 「 3 / 1 0 」 エリアと 「 構成 D 」 が必要とする 「 4 / 1 0 」 エリアを加えた 「 7 / 1 0 」 エリア 1 1 1 となる。

【 0 0 3 4 】

このように、順繰りにスレッド実行前に、必要な再構成を事前に行う。

10

20

30

40

50

<スレッドの実行順を変える場合>

次に、図3を用いて、スレッドの実行順を変えることにより、スレッド実行中に、次のスレッドの為に再構成が可能となる例を説明する。

図3(a)は、スレッド情報テーブル1420の構成例及び内容例を示す図であり、図3(b)及び(c)は、スレッドの実行例を示すタイムチャートである。

【0035】

図3(a)で示すスレッド情報テーブル1420のスレッドを実行した場合のタイムチャートが、図3(b)及び(c)で示すタイムチャートである。

まず、図3(a)のスレッド情報テーブル1420を説明する。

この図3(a)のスレッド情報テーブル1420は、図2(a)のスレッド情報テーブル1410とほとんど同じである為、差異のみを説明する。

【0036】

異なる点は、スレッド名1411「TH1」のスレッド1421が、再構成可能演算器1200を使用する点である。構成情報1412「構成B」、使用エリア数1413「5」個である。

次に、図3(b)を用いて、スレッドの実行例を示すタイムチャートを説明する。タイムチャートの意味するところは、図2(b)と同様である。

【0037】

スレッド名「TH0」のスレッド200が、固定機能演算器と、再構成可能演算器の「構成A」を使用して、スレッドを実行している。

この間に、次に実行予定のスレッド名1411「TH1」のスレッドが使用する構成情報1412「構成B」で再構成可能演算器1200を再構成しようとする。

この場合、使用している再構成可能演算器1200のエリア数は、「構成A」が使用している「6/10」エリアと「構成B」が必要とする「5/10」エリアを加えた「11/10」エリア201となり、スレッド名「TH0」のスレッドの実行中に「構成B」の再構成は出来ない。

【0038】

そこで、図3(c)に示すように、スレッド名「TH1」のスレッド230の前に、スレッド名「TH2」のスレッド220を実行する。すなわち、実行順序を入れ替えて、スレッドを実行する。

とすると、スレッド名「TH0」のスレッド実行中に、再構成するのはスレッド名「TH2」のスレッド220が使用する「構成C」であり、「構成A」が使用している「6/10」エリアと「構成C」が必要とする「3/10」エリアを加えた「9/10」エリア211となり、事前に再構成を行うことが出来るようになる。

【0039】

同様に、スレッド名「TH2」のスレッド220が、固定機能演算器1120と、再構成可能演算器1200の「構成C」を使用して、スレッドを実行している間に、次に実行予定のスレッド名1411「TH1」のスレッドが使用する構成情報1412「構成B」で再構成可能演算器1200を再構成する。

この場合、使用している再構成可能演算器1200のエリア数は、「構成C」が使用している「3/10」エリアと「構成B」が必要とする「5/10」エリアを加えた「8/10」エリア222となる。

【0040】

通常、ラウンドロビン方式では、各スレッドで実行すべき処理に応じて、スレッドに割り当てられるタイムスライスが考慮される。すなわち、処理時間を保障しなければいけない場合などには、タイムスライスの長いスレッドを割り当てるなどである。

したがって、周期が崩れないことが、処理をスレッドに割り当てる際の前提条件となる。

【0041】

しかし、一つのタイムスライスの時間は、処理全体の実行時間に比べれば、非常に小さ

10

20

30

40

50

いものである。本プロセッサでは、所定時間内で、各スレッドの実行時間が予定通りになるようにスレッドスケジューラ部 1110 で調整を行うこととする。例えば、全スレッドが 10 回実行することを一区切りとし、各スレッドの実行回数をカウントしておいて、最初のスレッドが 11 回目の実行を開始する前に、他のスレッドで 10 回に足りないものは、優先的に実行する。全スレッドが 10 回実行後、最初のスレッドの 11 回目の実行を開始するなどである。

#### 【0042】

<スレッドの制御処理>

次に、図 4 を用いて、本プロセッサのスレッド制御の処理について説明する。

図 4 は、本プロセッサのスレッド制御の処理を示すフローチャートである。

スレッドスケジューラ部 1110 が、次に実行するスレッドを選択する（ステップ S100）。制御処理を開始直後の場合は、最初のスレッドである。

#### 【0043】

各スレッドに割り当てられている処理がすべて終了している場合は（ステップ S110：Y）、スレッドの制御処理を終了する。

選択されたスレッドである次スレッドを実行する場合は、そのスレッド名 1411 を再構成制御部 1400 に渡し、再構成を依頼する。

依頼を受けた再構成制御部 1400 は、受取ったスレッド名 1411 が、再構成可能演算器 1200 を使用するか否かを構成情報記憶部 1300 に記憶されているスレッド情報テーブル 1410 を参照し、判断する。具体的には、受取ったスレッド名 1411 に対応する構成情報 1412 に構成が指定されている場合は、再構成可能演算器 1200 を使用すると判断する。

#### 【0044】

使用しないと判断した場合（ステップ S120：N）、再構成制御部 1400 は、その旨スレッドスケジューラ部 1110 に返し、スレッドスケジューラ部 1110 は、現在実行中のスレッドが終了次第、次スレッドの実行を開始する（ステップ S150）。

一方、使用すると判断した場合は（ステップ S120：Y）、再構成するエリアが空いているか否かを判断する（ステップ S130）。具体的には、受取ったスレッド名 1411 に対応する使用エリア数 1413 に示されている数分のエリアが空いているかを判断する。

#### 【0045】

再構成制御部 1400 は、内部に、現在使用されているエリアの番号を記憶しているものとし、使用しているスレッドのタイムスライスが終了した場合は、使用していたエリアは空いたものとされ、記憶しているエリア番号から消去する。

エリアが空いていないと判断した場合は（ステップ S130：N）、再構成制御部 1400 は、スレッドスケジューラ部 1110 にその旨通知する。スレッドスケジューラ部 1110 は、異なるスレッドを選択する（ステップ S100）。スレッドスケジューラ部 1110 は、各スレッドの実行回数を記憶しておき、適切な時に優先的に選択して、全スレッドの実行回数をあわせるものとする。

#### 【0046】

一方、エリアが空いていると判断した場合には（ステップ S130：Y）、再構成制御部 1400 は、再構成可能演算器 1200 に再構成する旨通知し、構成情報記憶部 1300 には、受取ったスレッド名 1411 に対応する構成情報 1412 を、エリアを指定して送信させる。送信後、再構成制御部 1400 は、内部に記憶している使用中のエリア番号を更新する。

#### 【0047】

再構成可能演算器 1200 は、構成情報記憶部 1300 から送られた構成情報で、再構成を行い（ステップ S140）、完了したら再構成制御部 1400 に通知する。

再構成が完了した旨の通知を受けた再構成制御部 1400 は、その旨スレッドスケジューラ部 1110 に返し、スレッドスケジューラ部 1110 は、現在実行中のスレッド終了

10

20

30

40

50

後、すぐに次スレッド実行を開始する（ステップS150）。

【0048】

スレッドを開始させたスレッドスケジューラ部1110は、次のスレッドを選択する（ステップS100）。

<実施形態2>

<概要>

実施形態1が、スレッド毎に、再構成した再構成可能演算器を使用したのに対して、本実施形態では、命令毎に、再構成した再構成可能演算器を使用するものである。

【0049】

以下、本実施形態2の構成等を説明する。

<構成>

図5は、実施形態2のプロセッサ5000の構成例を表す図である。

プロセッサ5000は、命令フェッチ部5100、命令デコード部5200、演算制御部5300、アドレステーブル記憶部5400、再構成情報記憶部5500、再構成可能演算器5600及び固定機能演算器5700を備え、外部に命令記憶部5010を備える。

【0050】

命令記憶部5010は、プロセッサ5000で実行する命令コードを記憶しておく機能を有する。

命令フェッチ部5100は、命令記憶部5010から、命令コードを読み込み、命令デコード部5200に渡す機能を有する。

命令デコード部5200は、命令フェッチ部5100から命令コードを受け取り、解析する通常の機能のほか、本発明に独自の機能を有する。

【0051】

具体的には、デコードの結果、再構成可能演算器5600を用いる命令の場合は、命令種別から構成情報の記憶されているアドレスをアドレステーブル記憶部5400から取得し、再構成情報記憶部5500にアドレスを渡して再構成可能演算器5600に構成情報を送信させる機能である。

アドレステーブル記憶部5400は、命令種別とその構成情報のアドレスとを対応付けて記憶している機能を有している。

【0052】

演算制御部5300は、命令デコード部5200がデコードした結果に従って、演算動作を制御する機能を有する。固定機能演算器5700と再構成可能演算器5600とに対して、タイミングをとりながら命令を発行していく。

再構成情報記憶部5500は、複数の各命令に対応して、構成情報を記憶する機能を有する。ここで記憶されている各構成情報の先頭アドレスが、アドレステーブル記憶部5400で、命令種別と対応付けて記憶されている。この構成情報は、実施形態1の構成情報記憶部1300に記憶されている構成情報と同様である。

【0053】

また、再構成情報記憶部5500は、命令デコード部5200からの指示で、指定されたアドレスの構成情報を、再構成可能演算器5600に送信する機能も有している。

再構成可能演算器5600は、再構成が可能な演算器であって、実施形態1の再構成可能演算器1200と同様である。但し本実施形態では、4つのエリアに分割されているものとする。

【0054】

固定機能演算器5700は、複数の固定機能演算器で構成され、本実施形態では3つの固定機能演算器（5701、5702、5703）で構成されるものとする。

以下、命令と構成情報との対応関係を簡単に説明し、動作を説明する。

<命令コードと構成情報との対応関係>

本発明で使用する命令コードから、その命令の実行に必要な再構成を行う為の構成情報

10

20

30

40

50

の求め方を、図 6 及び図 7 を用いて説明する。

【 0 0 5 5 】

図 6 は、本発明で使用する命令コードの構成例を示す図であり、図 7 は、命令情報テーブル 5 4 1 0 の構成例及び内容例を示す図である。

まず、図 6 の命令コードの構成例から説明する。

本発明で使用する命令コード 5 1 1 0 は、命令の種別を示すオペコード 5 1 1 1 と、この命令で扱う値などを示すオペランド 5 1 1 2 で構成される。

【 0 0 5 6 】

本発明では、このオペコード 5 1 1 1 と構成情報が対応付けられており（矢印参照）、命令の実行には再構成可能演算器 5 6 0 0 が必要であると、プロセッサによって判断された場合には、対応付けられている構成情報によって再構成された再構成可能演算器 5 6 0 0 で命令が実行される。

プロセッサによって、命令の実行には再構成可能演算器 5 6 0 0 は不要であると、判断された場合には、固定機能演算器 5 7 0 0 を用いて、実行がなされる。

【 0 0 5 7 】

次に、図 7 の命令情報テーブル 5 4 1 0 について説明する。

この命令情報テーブル 5 4 1 0 は、アドレステーブル記憶部 5 4 0 0 に記憶されているものとする。

命令情報テーブル 5 4 1 0 は、オペコード種別 5 4 1 1、アドレス 5 4 1 2 及び使用エリア数 5 4 1 3 とで構成される。

【 0 0 5 8 】

オペコード種別 5 4 1 1 は、命令コードのオペコード、すなわち、命令を示すものである。ここでは、再構成可能演算器 5 6 0 0 を用いる命令のみが記載されているものとする。

従って、ここに記載されていない命令は、固定機能演算器 5 7 0 0 で実行することになる。

【 0 0 5 9 】

次に、アドレス 5 4 1 2 は、オペコード種別 5 4 1 1 で表されるオペコードに対応付けられている構成情報の、再構成情報記憶部 5 5 0 0 内でのアドレスを示す。尚、本実施形態ではアドレスとしているが、ID 等、再構成情報を特定できるものであればよい。

使用エリア数 5 4 1 3 は、再構成可能演算器 5 6 0 0 を使用する場合のエリア数を表す。例えば、オペコード種別 5 4 1 1 「Sub」の命令は、アドレス 5 4 1 2 「a d d r 1」で示されるアドレスに記憶されている構成情報で再構成された再構成可能演算器 5 6 0 0 を使用し、再構成可能演算器 5 6 0 0 を再構成するには、使用エリア数 5 4 1 3 「3」個のエリアが必要である。

【 0 0 6 0 】

本実施形態の場合、命令の使用する再構成可能演算器 5 6 0 0 のエリア数を考慮し、プログラムを機械語に変換するコンパイル時に、本実施形態では命令コード 5 1 1 0 に変換する時に、命令の順序及び命令で再構成するエリアの番号を決めておくものとする。すなわち、コンパイル時に、実行する命令の順序を考慮した上で、前の命令の実行中に再構成できるような命令順に構成し、それぞれの再構成を行うエリアを決めておくものとする。また、使用するエリア番号は、例えば、オペランドでエリア番号を指定したり、命令毎にエリア番号を決めておくなどして、命令デコード部が知ることができるようにしてあるものとする。

【 0 0 6 1 】

< 動作 >

次に、図 8 ~ 図 1 0 とを用いて、どのように命令が実行されるかを説明する。

図 8 は、本発明に係る命令セットを用いたプログラム例であり、図 9 は、プログラムを動作させたプロセッサのパイプライン動作の例である。

また、図 1 0 は、本実施形態のプロセッサの命令実行処理を示すフローチャートである

10

20

30

40

50

## 【 0 0 6 2 】

まず、図 8 のプログラムについて簡単に説明する。

オペコード 5 1 1 1 「Add」、オペランド 5 1 1 2 「r0,r1,r2」の命令コードは、レジスタ 1 の内容とレジスタ 2 の内容を加算し、結果をレジスタ 0 に代入することを意味し、オペコード 5 1 1 1 「Sub」、オペランド 5 1 1 2 「r3,r1,r3」の命令コードは、レジスタ 1 の内容からレジスタ 3 の内容を減算し、結果をレジスタ 3 に代入することを意味する。

## 【 0 0 6 3 】

また、オペコード 5 1 1 1 「Reconf0」、オペランド 5 1 1 2 「r2,r0,0xfe」の命令コードは、レジスタ 0 の内容と即値「0xfe」を用いて、演算「Reconf0」を行い、結果をレジスタ 2 に代入することを意味し、オペコード 5 1 1 1 「Reconf1」、オペランド 5 1 1 2 「r3,r1,r3」の命令コードは、レジスタ 1 の内容とレジスタ 3 の内容とを用いて、演算「Reconf1」を行い、結果をレジスタ 3 に代入することを意味する。

10

## 【 0 0 6 4 】

次に、図 9 及び図 10 を用いて、プログラムを実行するプロセッサの動作を説明する。図 10 のフローチャートに沿って、図 9 のタイムチャートを参照しながら説明する。

図 8 に示すプログラムが命令記憶部 5 0 1 0 に記憶されているものとする。

まず、命令フェッチ部 5 1 0 0 は、命令コード「Add r0,r1,r2」をフェッチし（図 10 : ステップ S 8 0 0、図 9 : ステップ S 5 0 0）、命令デコード部 5 2 0 0 に渡す。

20

## 【 0 0 6 5 】

命令コードを受取った命令デコード部 5 2 0 0 は、受取った命令コードを解析する。

受取った命令コードが終了を示す旨のコードである場合には（図 10 : ステップ S 8 1 0 : Y）、命令デコード部 5 2 0 0 は、処理を終了する。

また、命令コードが終了を示す旨のコードでない場合には（図 10 : ステップ S 8 1 0 : N）、命令デコード部 5 2 0 0 は、オペコード 5 1 1 1 「Add」をアドレステーブル記憶部 5 4 0 0 に渡して、構成情報のアドレスを要求する。

## 【 0 0 6 6 】

アドレステーブル記憶部 5 4 0 0 は、命令情報テーブル 5 4 1 0 を参照し、オペコード種別 5 4 1 1 に、渡されたオペコード 5 1 1 1 「Add」が在るかを検索し、結果、ないことから、再構成可能演算器 5 6 0 0 は使用しない旨を命令デコード部 5 2 0 0 に返す（図 10 : ステップ S 8 2 0 : N、図 9 : ステップ S 5 1 0）。

30

再構成可能演算器 5 6 0 0 を使用しない旨を受取った命令デコード部 5 2 0 0 は、「Add r0,r1,r2」のデコード済みの命令を演算制御部 5 3 0 0 に渡す。

## 【 0 0 6 7 】

デコード結果を渡された演算制御部 5 3 0 0 は、固定機能演算器 5 7 0 0 に命令を発行し、命令コード「Add r0,r1,r2」を実行する（図 10 : ステップ S 8 3 0、8 4 0）。

命令フェッチ部 5 1 0 0 は、命令コード「Add r0,r1,r2」をフェッチ後、次の命令コード「Sub r3,r1,r3」をフェッチし（図 10 : ステップ S 8 0 0、図 9 : ステップ S 5 2 0）、命令デコード部 5 2 0 0 に渡す。

40

## 【 0 0 6 8 】

命令コードを受取った命令デコード部 5 2 0 0 は、オペコード 5 1 1 1 「Sub」をアドレステーブル記憶部 5 4 0 0 に渡して、構成情報のアドレスを要求する。

アドレステーブル記憶部 5 4 0 0 は、命令情報テーブル 5 4 1 0 を参照し、オペコード種別 5 4 1 1 に、渡されたオペコード 5 1 1 1 「Sub」が在るかを検索し、結果、あることから、アドレス 5 4 1 2 「a d d r 1」を命令デコード部 5 2 0 0 に返す（図 10 : ステップ S 8 2 0 : Y、図 9 : ステップ S 6 0 0）。

## 【 0 0 6 9 】

アドレスを受取った命令デコード部 5 2 0 0 は、再構成情報記憶部 5 5 0 0 に受取ったアドレス 5 4 1 2 「a d d r 1」を渡し、そのアドレスの構成情報を再構成可能演算器 5

50

600に送信して再構成を行う旨を指示する。

指示を行った命令デコード部5200は、「Sub r3,r1,r3」のデコード済みの命令を演算制御部5300に渡す。

【0070】

一方指示を受取った再構成情報記憶部5500は、受取ったアドレスの構成情報を再構成可能演算器5600に送信して再構成を行う(図10:ステップS850、図9:ステップS610)。

この再構成は、再構成可能演算器5600の4エリアのうち、3エリアを再構成する(図7参照)。

【0071】

デコード結果を渡された演算制御部5300は、再構成可能演算器5600に命令を発行し、命令コード「Sub r3,r1,r3」を実行する(図10:ステップS860、図9:ステップS620)。

その後、実行結果をレジスタ3に書き出す(図9:ステップS630)。

このように、命令を順次同時に行っていく。

【0072】

ここで、命令コード「Sub r3,r1,r3」の次の命令コード「Reconf0 r2,r0,0xfe」を実行する場合を見ると、このオペコード「Reconf0」は再構成可能演算器5600を使用する。

従って、「Sub r3,r1,r3」を実行中(図9:ステップS620)に、再構成を行うことになる(図9:ステップS700)。

【0073】

このオペコード種別5411「Reconf0」は、使用エリア数5413「1」個であることから、「Sub r3,r1,r3」がエリアを3個使用して実行中であっても、再構成が可能となる。

次の命令コード「Reconf1 r3,r1,r3」を実行する場合も同様である。

このように、再構成可能な演算器を持つプロセッサにおいて、命令単位で再構成可能な演算器を制御することが可能となり、柔軟且つ高性能な演算処理を、高い面積効率で実現することができる。

【0074】

<変形例>

次に、実施形態2の変形例について説明する。

実施形態2では、1命令ごとに再構成可能演算器を再構成したが、本変形例では、複数の命令を一度に再構成する例を説明する。同時に複数の命令を実行できることから、処理速度の向上を図ることができる。

【0075】

図11~図13とを用いて、どのように命令が実行されるかを説明する。

図11は、本変形例の命令セットを用いたプログラム例であり、図12は、プログラムを動作させたプロセッサのパイプライン動作の例である。

また、図13は、本変形例のプロセッサの命令実行処理を示すフローチャートである。

まず、図11のプログラムの、各命令の内容については、図8と同様である。

【0076】

但し、「Reconf0 r2,r0,0xfe」と「Reconf1 r3,r1,r3」とを同じステージで行う点異なる。図11では、説明の便宜上、「Reconf0 r2,r0,0xfe」と「Reconf1 r3,r1,r3」とを横に並べて記載することで、同時に実行されることを示しているが、2命令を同時に実行することを命令デコード部5200が解釈できるように、コンパイラが出力しておくこととする。

【0077】

例えば、命令の使用する再構成可能演算器5600のエリア数を考慮し、コンパイル時に、命令の順序及び命令で再構成するエリアの番号を決める。すなわち、コンパイル時に

10

20

30

40

50

、命令実行の順序を考慮して、同時に実行可能で、且つ、同時に再構成できる命令を選択し、それぞれの再構成を行うエリアを決める。この並列に実行する命令と使用するエリア番号とは、例えば、並列実行を示す命令コードを設けて、そのオペランドにこれら命令とエリア番号などを記載する。

【 0 0 7 8 】

具体的には、コンパイルにおいて、使用エリア数 5 4 1 3 が考慮されて、「Sub」と「Reconf0」とは同時に再構成可能であるが、「Sub」と「Reconf1」とは同時に再構成は出来ないことになる。「Sub」と「Reconf0」とは合わせて4つのエリアを使用するが、「Sub」と「Reconf1」とは合わせて5つのエリアを必要とするからである(図7参照)。

図12及び図13を用いて、プログラムを実行するプロセッサの動作を簡単に説明する。図13のフローチャートに沿って、図12のタイムチャートを参照しながら説明する。

10

【 0 0 7 9 】

図13のフローチャートは、図10のフローチャートとの差異のみを説明する。具体的には、ステップS900～ステップS920までである。

「Reconf0 r2,r0,0xfe」と「Reconf1 r3,r1,r3」は、再構成可能演算器を使用し、且つ、同時に実行することを解析した命令デコード部5200は、オペコード5111「Reconf0」と「Reconf1」とをアドレステーブル記憶部5400に渡して、構成情報のアドレスを要求する。

【 0 0 8 0 】

アドレステーブル記憶部5400は、命令情報テーブル5410を参照し、アドレス5412「addr4」と「addr5」とを命令デコード部5200に返す(図13:ステップS820:Y、ステップS900:Y)。

20

アドレスを受取った命令デコード部5200は、再構成情報記憶部5500に受取ったアドレス5412「addr4」と「addr5」を渡し、そのアドレスの構成情報を再構成可能演算器5600に送出して再構成を行う旨を指示する。

【 0 0 8 1 】

指示を行った命令デコード部5200は、「Reconf0 r2,r0,0xfe」と「Reconf1 r3,r1,r3」のデコード済みの命令を演算制御部5300に渡す。

一方指示を受取った再構成情報記憶部5500は、受取ったアドレスの構成情報を再構成可能演算器5600に送信して再構成を行う(図13:ステップS910、図12:ステップS700)。

30

【 0 0 8 2 】

デコード結果を渡された演算制御部5300は、再構成可能演算器5600に命令を発行し、命令コード「Reconf0 r2,r0,0xfe」と「Reconf1 r3,r1,r3」を実行する(図13:ステップS920、図12:ステップS710)。

命令コードを受取った命令デコード部5200は、再構成可能演算器を使用しない場合(図13:ステップS820:N)と、使用したとしても1つの命令の場合(図13:ステップS900:N)とは、それぞれ図10と同様の処理を行う(図13:ステップS830～ステップS870)。

【 0 0 8 3 】

40

尚、ここでは、再構成可能な演算器を用いる二つの命令を同時に処理する場合を説明したが、同時に発行できる命令数は二つに限るものではない。また、固定機能演算器5700を使用する命令と同時に処理してもよい。

< 補足 >

以上、本発明に係るプロセッサについて実施形態に基づいて説明したが、このプロセッサを部分的に変形することもでき、本発明は上述の実施形態に限られないことは勿論である。即ち、

(1) 実施形態2では、命令コードのオペコードと構成情報とを対応付けていたが、これに限られない。

【 0 0 8 4 】

50

例えば、オペランドの一部に構成情報を示すコードを入れておいても良い。図14に示すように、命令コード5150のオペランドのフィールド5160に、構成情報のIDなどを入れておき、実行時にIDに応じて構成情報を特定する。

(2) 実施形態2では、説明の便宜上、複数の固定機能演算器の実行については詳細な説明はしていないが、これらの固定機能演算器と再構成可能演算器とが同時に実行できるものである場合には、同時に複数の命令を発行することとしても良い。

【0085】

同時に発行すべき命令の決定如何によっては、演算効率を大幅に向上させることが可能となる。

すなわち、再構成可能なハードウェアによって構成された演算器においては、複数種類の演算機能が選択的に実行可能である。従って、任意の命令を実行することでき、本発明による命令セットを用いることにより、命令の並列度を向上させた最適な機能を実現するプログラムを作成することが可能となるからである。

【0086】

同時に発行すべき命令を決定する動作は、例えば、プロセッサの内部で命令解釈時に行われるものでも、プロセッサに対して与えられるプログラムの時点で、予め行われているものでもよい。

(3) 実施形態では、再構成可能演算器は、均質な複数のエリアに分かれていることとしているが、エリア毎に異なる論理ブロックを有していてもよく、また、そのエリアの大きさが異なってもよい。

(4) 実施形態では、再構成可能演算器を構成する論理ブロックは、ルックアップテーブルとフリップフロップを含む回路ユニットであることとしているが、論理ブロックは、ALU (Arithmetic and Logical Unit)、シフトやデータ制御、論理演算を行なうユニット、フリップフロップなどから構成されるもの、すなわち、一般の論理回路の組合せで構成されるものであってもよい。

【産業上の利用可能性】

【0087】

本発明にかかるプロセッサは、回路規模を抑えつつ柔軟かつ高性能な処理を実現することが可能であるため、画像処理LSIの演算器等として特に有用である。

【図面の簡単な説明】

【0088】

【図1】プロセッサ1000の構成例を表す図である。

【図2】図2(a)は、スレッド情報テーブル1410の構成例及び内容例を示す図であり、図2(b)は、スレッドの実行例を示すタイムチャートである。

【図3】図3(a)は、スレッド情報テーブル1420の構成例及び内容例を示す図であり、図3(b)及び(c)は、スレッドの実行例を示すタイムチャートである。

【図4】本プロセッサのスレッド制御の処理を示すフローチャートである。

【図5】実施形態2のプロセッサ5000の構成例を表す図である。

【図6】実施形態2で使用する命令コードの構成例を示す図である。

【図7】命令情報テーブル5410の構成例及び内容例を示す図である。

【図8】実施形態2に係る命令セットを用いたプログラム例である。

【図9】プログラムを動作させたプロセッサのパイプライン動作の例である。

【図10】実施形態2のプロセッサの命令実行処理を示すフローチャートである。

【図11】変形例の命令セットを用いたプログラム例である。

【図12】変形例のプログラムを動作させたプロセッサのパイプライン動作の例である。

【図13】変形例のプロセッサの命令実行処理を示すフローチャートである。

【図14】実施形態2で使用する命令コードの構成の変形例を示す図である。

【符号の説明】

【0089】

1000 5000 プロセッサ

10

20

30

40

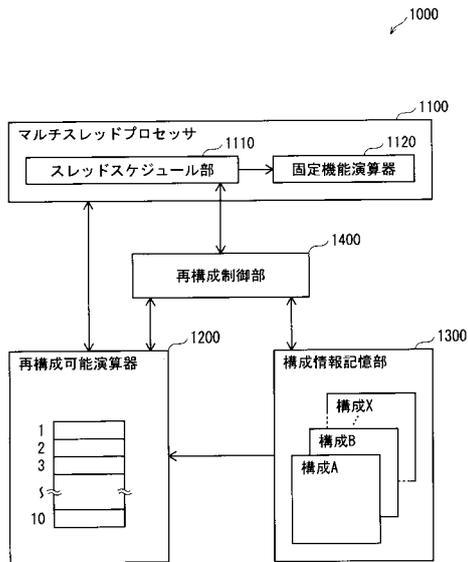
50

- 1 1 0 0 マルチスレッドプロセッサ
- 1 1 1 0 スレッドスケジュール部
- 1 1 1 0 旨スレッドスケジュール部
- 1 1 2 0 固定機能演算器
- 1 2 0 0 再構成可能演算器
- 1 3 0 0 構成情報記憶部
- 1 3 0 0 再構成情報記憶部
- 1 4 0 0 再構成制御部
- 1 4 1 0 1 4 2 0 スレッド情報テーブル
- 5 0 1 0 命令記憶部
- 5 1 0 0 命令フェッチ部
- 5 1 1 0 5 1 5 0 命令コード
- 5 1 1 1 オペコード
- 5 1 1 2 オペランド
- 5 2 0 0 命令デコード部
- 5 3 0 0 演算制御部
- 5 4 0 0 アドレステーブル記憶部
- 5 4 1 0 命令情報テーブル
- 5 4 1 3 使用エリア数
- 5 5 0 0 再構成情報記憶部
- 5 6 0 0 再構成可能演算器
- 5 7 0 0 固定機能演算器

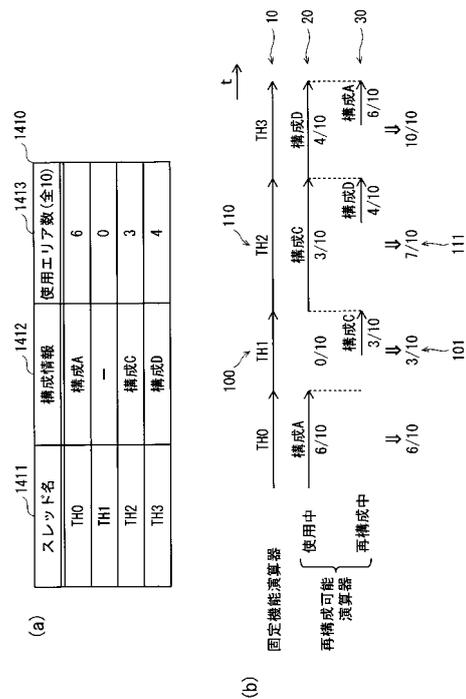
10

20

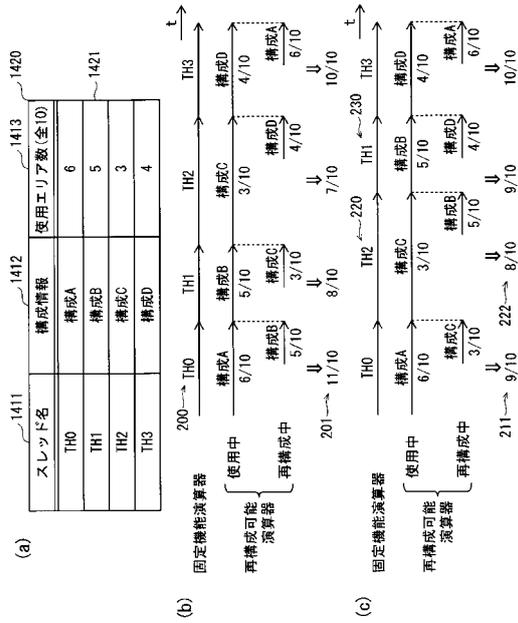
【図 1】



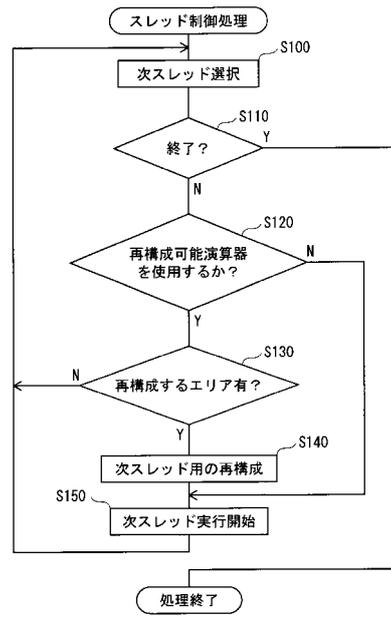
【図 2】



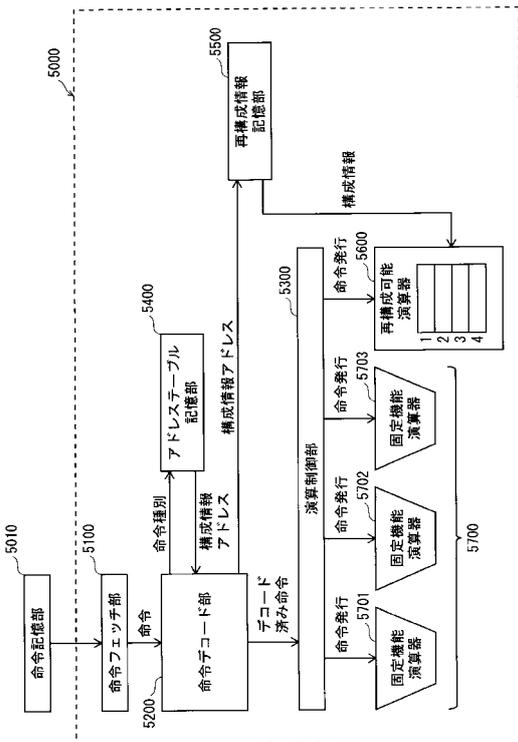
【図3】



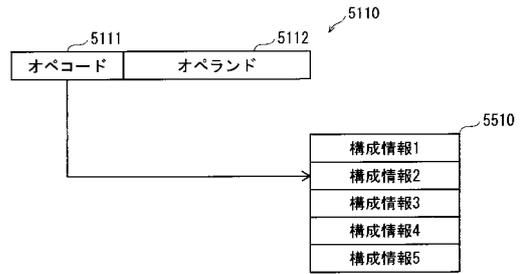
【図4】



【図5】



【図6】



【図7】

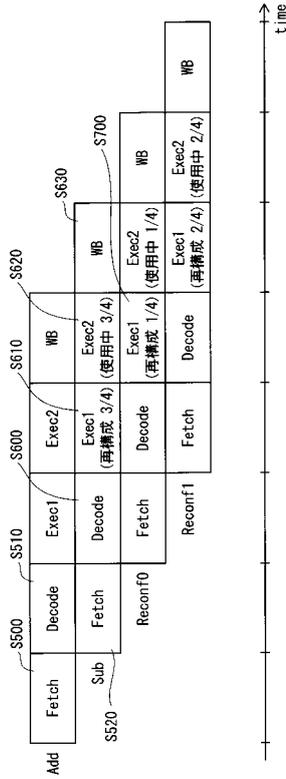
オペコード種別	アドレス	使用エリア数(全4)
Sub	addr1	3
Mul	addr2	4
Dev	addr3	4
Reconf0	addr4	1
Reconf1	addr5	2

【図8】

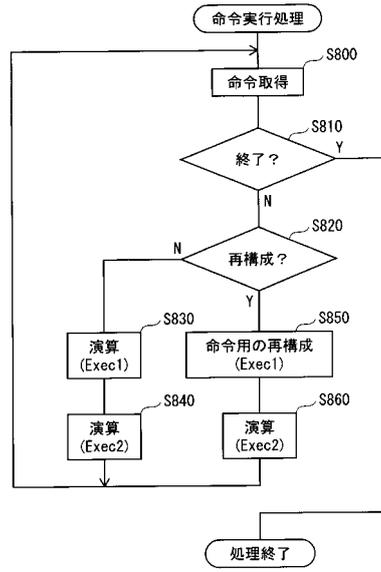
5111                      5112

Add	r0, r1, r2
Sub	r3, r1, r3
Reconf0	r2, r0, 0xfe
Reconf1	r3, r1, r3

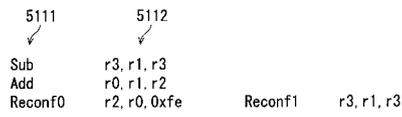
【 図 9 】



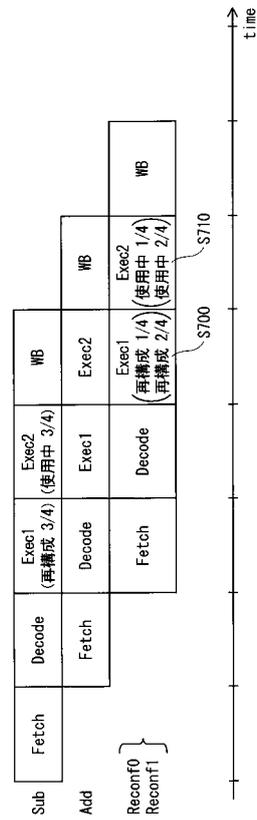
【 図 10 】



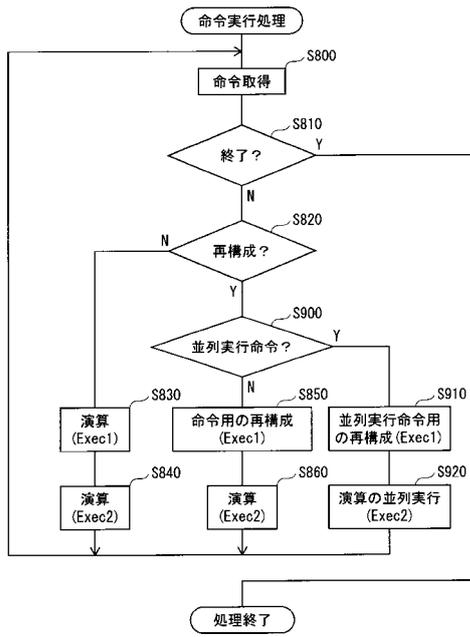
【 図 11 】



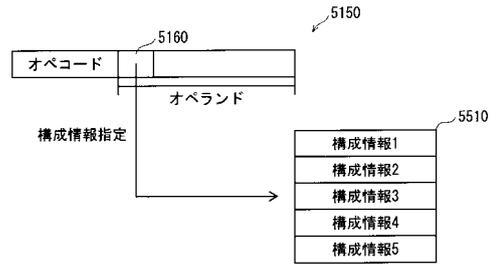
【 図 12 】



【図 13】



【図 14】



---

フロントページの続き

(72)発明者 清原 督三

大阪府門真市大字門真1006番地 松下電器産業株式会社内

審査官 高橋正徳

(56)参考文献 特開2000-242613(JP,A)

特開平08-286908(JP,A)

特開平10-320214(JP,A)

特開2001-068993(JP,A)

特開2004-070869(JP,A)

特開平08-069377(JP,A)

特開平10-083303(JP,A)

特開2007-334538(JP,A)

OKADA, M., et al., A Reconfigurable Processor based on ALU array architecture with limitation on the interconnection, Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International, IEEE, 2005年 4月 4日, pp. 1 - 6

飯田全広、外2名、マルチスレッド制御ライブラリのハードウェア化によるリコンフィギュラブルシステム、電子情報通信学会技術研究報告、日本、社団法人電子情報通信学会、1996年12月13日、第96巻、第426号、第135頁乃至第142頁

(58)調査した分野(Int.Cl., DB名)

G06F 9/38,

G06F 9/46,

G06F 15/80