

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3719509号
(P3719509)

(45) 発行日 平成17年11月24日(2005.11.24)

(24) 登録日 平成17年9月16日(2005.9.16)

(51) Int. Cl.⁷

F I

G06F	7/00	G06F	7/00	A
G06F	9/30	G06F	9/30	370
G06F	9/38	G06F	9/38	310J

請求項の数 13 (全 22 頁)

<p>(21) 出願番号 特願2002-99202 (P2002-99202)</p> <p>(22) 出願日 平成14年4月1日(2002.4.1)</p> <p>(65) 公開番号 特開2003-296096 (P2003-296096A)</p> <p>(43) 公開日 平成15年10月17日(2003.10.17)</p> <p>審査請求日 平成16年5月13日(2004.5.13)</p>	<p>(73) 特許権者 395015319 株式会社ソニー・コンピュータエンタテインメント 東京都港区南青山二丁目6番21号</p> <p>(74) 代理人 100099324 弁理士 鈴木 正剛</p> <p>(74) 代理人 100108604 弁理士 村松 義人</p> <p>(74) 代理人 100111615 弁理士 佐野 良太</p> <p>(72) 発明者 直井 純一 東京都港区赤坂7丁目1番1号 株式会社 ソニー・コンピュータエンタテインメント 内</p>
---	---

最終頁に続く

(54) 【発明の名称】 シリアル演算パイプライン、演算装置、算術論理演算回路およびシリアル演算パイプラインによる演算方法

(57) 【特許請求の範囲】

【請求項1】

各々独立に制御され他の算術論理演算回路と同時に動作可能な複数の算術論理演算回路をカスケード接続して成るシリアル演算パイプラインであって、

前記複数の算術論理演算回路の全部又は一部が、後段へデータを出力するための第1ラインと、前段へデータをフィードバックするための第2ラインと、各ライン上のデータをラッチするラッチ回路とを有しており、

前記複数の算術論理演算回路の各々が、外部から入力されたインストラクションの内容に応じた演算を含む処理を実行し、その実行結果を表すデータを所望のライン宛に出力するデコーダを有しており、

前記第1ラインが、第1の出力データおよび第2の出力データを後段に出力するための一対のラインであり、

前記ラッチ回路が、前記第1の出力データをラッチする第1のデータラッチ回路と、前記第2の出力データをラッチする第2のデータラッチ回路と、前記デコーダによる演算の結果生じるキャリーを次の桁の演算のためにラッチするキャリーラッチ回路と、前記第1の出力データおよび前記第2の出力データの一方を所定期間遅延させて他方との参照を可能にするシフトラッチ回路と、を含む、

シリアル演算パイプライン。

【請求項2】

前記ラッチ回路が、さらに、前段にフィードバックするリバーデータラッチするリ

バースラッチ回路と、前記第1および第2のデータラッチ回路および前記リバースラッチ回路にラッチされているデータの演算を制御するコントロールデータをラッチするコントロールラッチ回路とを含むことを特徴とする、

請求項1記載のシリアル演算パイプライン。

【請求項3】

前記デコーダにより処理されるデータが、単位処理サイズのデータであることを特徴とする、請求項1記載のシリアル演算パイプライン。

【請求項4】

前記単位処理サイズが、前記複数の算術論理演算回路の動作タイミングを定める一つのクロックで処理可能な最低ビット長ないし同等のビット長であることを特徴とする、

10

請求項3記載のシリアル演算パイプライン。

【請求項5】

前記第1ラインが前段から受け取ったデータないしそれに基づく演算結果を後段に出力するための順方向ラインであり、前記第2ラインが後段から受け取ったデータないしそれに基づく演算結果を前段に出力するための逆方向ラインであることを特徴とする、請求項1ないし4のいずれかの項記載のシリアル演算パイプライン。

【請求項6】

複数系統のデータ入出力ラインを切替接続する切替接続手段と、この切替接続手段に対して並列に接続された複数のシリアル演算パイプラインと、外部から入力された前記複数のシリアル演算パイプライン宛のインストラクションを受け付けるインストラクション受付機構とを備え、

20

前記複数のシリアル演算パイプラインの各々は、他の算術論理演算回路と同時に動作可能な複数の算術論理演算回路をカスケード接続して構成されたシリアル演算パイプラインであり、

前記複数の算術論理演算回路の各々は、前段から受け取ったデータないしそれに基づく演算結果を後段に出力するための順方向ラインと、後段から受け取ったデータないしそれに基づく演算結果を前段にフィードバックするための逆方向ラインと、入力された前記インストラクションに応じた処理を実行し実行結果を所望のラインに出力するデコーダと、他の算術論理演算回路との間で各ライン上のデータ出力の同期をとるためのラッチ回路とを有するものであり、

30

前記複数の算術論理演算回路の各々のデコーダにおける順方向ライン上のデータおよび逆方向ライン上のデータが単位処理サイズのデータである、

演算装置。

【請求項7】

n行m列の前記インストラクションの集合であるインストラクションアレイを保持し、このインストラクションアレイを列毎にm回前記複数の算術論理演算回路宛に出力するとともにパイプラインが後段に1段ずつn回移行させる際に、実行済みのインストラクションを1段毎に1つ減ずるコントローラをさらに備えてなる、

請求項6記載の演算装置。

【請求項8】

40

前記インストラクションアレイには、1パスで演算を実行するための演算手順が定められている、

請求項7記載の演算装置。

【請求項9】

複数系統のデータ入出力ラインを切替接続する切替接続手段と、この切替接続手段に対して並列に接続された複数のシリアル演算パイプラインと、外部から入力された前記複数のシリアル演算パイプライン宛のインストラクションを受け付けるインストラクション受付機構とを備え、

前記複数のシリアル演算パイプラインの各々は、他の算術論理演算回路と同時に動作可能な複数の算術論理演算回路をカスケード接続して構成されたシリアル演算パイプライン

50

であり、

前記複数の算術論理演算回路の各々は、前段から受け取ったデータないしそれに基づく演算結果を後段に出力するための順方向ラインと、後段から受け取ったデータないしそれに基づく演算結果を前段にフィードバックするための逆方向ラインと、入力された前記インストラクションに応じた処理を実行し実行結果を所望のラインに出力するデコーダと、他の算術論理演算回路との間で各ライン上のデータ出力の同期をとるためのラッチ回路とを有するものであり、

前記順方向ラインが、第1の出力データおよび第2の出力データを後段に出力するための一対のライン、前記逆方向ラインがリバースデータを出力するラインであり、

前記複数の算術論理演算回路の各々のラッチ回路が、後段に出力する第1の出力データをラッチする第1のデータラッチ回路と、

後段に出力する第2の出力データをラッチする第2のデータラッチ回路と、

前段に出力するデータをラッチするリバースラッチ回路と、

演算結果のキャリーを次の桁の演算のためにラッチするキャリーラッチ回路と、

前記第1の出力データおよび前記第2の出力データの一方を所定期間遅延させて他方との参照を可能にするシフトラッチ回路と、を含むことを特徴とする、演算装置。

【請求項10】

前記切替接続手段、前記複数のシリアル演算パイプライン、および前記インストラクション受付機構が一つの半導体デバイスの中に組み込まれていることを特徴とする、

請求項6記載の演算装置。

【請求項11】

シリアル演算パイプラインの構成要素となるカスケード接続可能な算術論理演算回路であって、前記シリアル演算パイプラインの前段から受け取ったデータないしそれに基づく演算結果を後段に出力するための1又は複数の順方向ラインと、後段から受け取ったデータないしそれに基づく演算結果を前段にフィードバックするための逆方向ラインと、外部から入力されたインストラクションに応じた処理を実行し実行結果を表すデータを所望のラインに出力するデコーダと、他の算術論理演算回路との間で各ライン上のデータ出力のタイミングを同期させるためのラッチ回路とを備え、

前記順方向ラインが、第1の出力データおよび第2の出力データを出力する一対のライン、前記逆方向ラインがリバースデータを一つの出力するラインであり、

前記ラッチ回路が、前記第1の出力データをラッチする第1のデータラッチ回路と、前記第2の出力データをラッチする第2のデータラッチ回路と、前記リバースデータをラッチするリバースラッチ回路と、前記デコーダによる演算結果のキャリーを次の桁の演算のためにラッチするキャリーラッチ回路と、第1の出力データおよび第2の出力データの一方を所定期間遅延させて他方との参照を可能にするシフトラッチ回路と、前記第1および第2のデータラッチ回路およびリバースラッチ回路にラッチされているデータの演算を制御するためのコントロールデータをラッチするコントロールラッチ回路とを含んで成る、算術論理演算回路。

【請求項12】

後段にデータを出力するための1又は複数の第1ラインと、前段にデータをフィードバックするための第2ラインとを有し、シリアル演算を行うとともに演算結果を表すデータの出力対象となるラインを各々独立に選択することができる算術論理演算回路を複数段カスケード接続してシリアル演算パイプラインを構築し、前記シリアル演算パイプラインに接続された外部コントローラで、前記複数の算術論理演算回路による1パスによる同時実行の手順を反映したm行n列のインストラクションの集合であるインストラクションアレイを列毎にm回前記複数の算術論理演算回路宛に個別に出力するとともに、パイプラインが後段に1段ずつn回移行させる際に、実行済みのインストラクションを1段毎に1つ減ずる制御を行うことを特徴とする、シリアル演算パイプラインによる演算方法。

【請求項13】

前記複数の算術論理演算回路の各々のデコーダにおける各ライン上のデータが、単位処

10

20

30

40

50

理サイズのデータであることを特徴とする、請求項 1 2 記載の演算方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、例えば頻繁に動くオブジェクトをコンピュータ・グラフィクスによって表現するような、突発的かつ爆発的に膨らむ離散演算を必要とする用途に適したシリアル演算パイプライン (pipeline) 及びその構成要素に関する。

ここで、シリアル演算パイプラインは、命令取り出し (fetch)、命令解読 (decode)、その実行 (execute) のような、別々の作業を順次同時に行なうことによって処理の高速化を図るものであり、少数の命令群を扱う演算器をカスケード接続して構成される。カスケード接続された複数の演算器のうち使用するものの組み合わせを適宜変えることにより、加減算、浮動小数点演算、比較、ブール代数、選択 (IF文) 等、様々な演算処理を実現することができる。

【0002】

【発明の背景】

コンピュータ・グラフィクスの作成に際しては、通常、2次元ピクセル内・ピクセル間の演算、空間把握 (コリジョン・デテクション等)、オブジェクト生成・合成、ジオメトリ演算等、多種の離散的演算を伴う。このような用途では、コンピュータのメインCPUだけでは、到底不可能なほどの大量演算パワーが必要となる。例えば、しばしば数100 [M polygon / sec]、数10 [Gpixel / sec] のオーダのレンダリング処理能力が必要になる。

そのため、従来より、演算パイプラインを搭載した専用のプロセッサが使用されてきた。

【0003】

従来のこの種のプロセッサの殆どは、単一のデバイスで構成され、見込まれる演算量に応じて演算パイプラインを並列化して組み込まれている。演算パイプラインを構築する個々の演算器にも所定の機能が固定的に割り当てられている。いわゆる「1機能1演算器」である。このような1機能1演算器によるパイプラインは、例えば固定長のデータサイズを短スループットで処理する用途には、非常に適している。

【0004】

しかし、「1機能1演算器」では、多様な用途には、使い回しが利きにくい。例えば用途に応じてパイプライン構成を変更しようとする場合は、演算対象となるデータの通り道と割り当てられた機能とを結びつけるためのセレクタ (バス) が別途必要となる。そのため、単一デバイス内での並列化には限界があった。また、演算器又は演算パイプラインを一定数以上並列化するとクラスタ化が避けられず、そのための制御やデータパス (キャッシュ又はバス) がそれぞれ必要となって、集積効率が悪化してしまう。

【0005】

用途の多様化に対応するために、プログラマブル・データ・フロー・グラフ (DFG) を構成することが考えられる。しかし、プログラマブルDFGは、演算器に対して、セレクタのような非演算要素の占める割合が比較的高い。演算のプログラマビリティを上げるほど、その比率が高まることは、よく知られていることである。プログラマブルDFGは、また、機能 (function) が細分化されているため、すべての機能を常時稼働させることは困難である。稼働効率を上げるためには、ある程度固定された機能の塊にせざるを得ず、様々なタイプのデータを処理すること用途には不向きとなる。

【0006】

一方、より高い演算能力を確保する観点からは、演算器の2次元並列化も考えられる。「2次元並列化」とは、演算器をパラレル及びカスケードに並列化することである。例えば、深パイプラインによるデータフローの並列化である。特殊なインプリメントとしては、並列化されたレンダリング・パイプラインがある。2次元並列化に際してはパイプライン内で必要な機能のみを盛り込み、プログラマビリティを極力排除して、専用演算器のカスケード接続で効率を上げている。

10

20

30

40

50

【 0 0 0 7 】

今後は、離散演算の多様化が予想される。その際、様々なデータサイズに対して突発的かつ爆発的な演算量を伴う複雑な処理フローを実現する汎用的なパイプラインが望まれる。汎用的な演算パイプラインを構築するためには、データパスを単純（一直線）にし、無駄なくカスケード接続する必要がある。また、多種多様な演算を1演算器で実現できるような仕組みが必要となる。

【 0 0 0 8 】

本発明は、このような仕組みを、コスト上昇を伴わずに実現しようとするものである。

【 0 0 0 9 】

【課題を解決するための手段】

本発明は、各々独立に制御され他の算術論理演算回路と同時に動作可能な複数の算術論理演算回路をカスケード接続して成るシリアル演算パイプラインを提供する。

このシリアル演算パイプラインにおいて、前記複数の算術論理演算回路の全部又は一部は、後段へデータを出力するための第1ラインと、前段へデータをフィードバックするための第2ラインと、各ライン上のデータをラッチするラッチ回路とを有しており、複数の算術論理演算回路の各々は、外部から入力されたインストラクションの内容に応じた演算を含む処理を実行し、その実行結果を表すデータを所望のライン宛に出力するデコーダを有している。

前記第1ラインは、第1の出力データおよび第2の出力データを後段に出力するための一対のラインであり、前記ラッチ回路は、前記第1の出力データをラッチする第1のデータラッチ回路と、前記第2の出力データをラッチする第2のデータラッチ回路と、前記デコーダによる演算の結果生じるキャリーを次の桁の演算のためにラッチするキャリーラッチ回路と、前記第1の出力データおよび前記第2の出力データ的一方を所定期間遅延させて他方との参照を可能にするシフトラッチ回路と、を含んでいる。

【 0 0 1 0 】

前記複数の算術論理演算回路の各々は、外部から入力されたインストラクション（命令）の内容に応じた演算を含む処理を実行し、その実行結果を所望のライン宛に出力するデコーダを有するものである。

また、前記ラッチ回路は、後段に出力する第1の出力データをラッチする第1のデータラッチ回路と、後段に出力する第2の出力データをラッチする第2のデータラッチ回路と、前記デコーダによる演算の結果生じたキャリーを次の桁の演算のためにラッチするキャリーラッチ回路と、第1の出力データおよび第2の出力データ的一方を所定期間遅延させて他方との参照を可能にするシフトラッチ回路とを含むものである。

前段にフィードバックするリバースデータをラッチするリバースラッチ回路と、前記第1および第2のデータラッチ回路およびリバースラッチ回路にラッチされているデータの出力タイミングを決めるコントロールデータをラッチするコントロールラッチ回路とをさらに含むようにしてもよい。

【 0 0 1 1 】

回路面積を縮小とするとともに、パイプライン全体の処理速度を高める観点からは、前記デコーダにより処理されるデータを、単位処理サイズのデータ、例えば、前記複数の算術論理演算回路の動作タイミングを定める一つのクロックで処理可能な最低ビット長ないし同等のビット長のデータとする。

【 0 0 1 2 】

本発明は、また、複数系統のデータ入出力ラインを切替接続する切替接続手段と、この切替接続手段に対して並列に接続された複数のシリアル演算パイプラインと、外部から入力された前記複数のシリアル演算パイプライン宛のインストラクションを受け付けるインストラクション受付機構とを備えた演算装置を提供する。

この演算装置において、前記複数のシリアル演算パイプラインの各々は、他の算術論理演算回路と同時に動作可能な複数の算術論理演算回路をカスケード接続して構成されたシリアル演算パイプラインであり、前記複数の算術論理演算回路の各々は、前段から受け取

10

20

30

40

50

ったデータないしそれに基づく演算結果を後段に出力するための順方向ラインと、後段から受け取ったデータないしそれに基づく演算結果を前段にフィードバックするための逆方向ラインと、入力された前記インストラクションに応じた処理を実行し実行結果を所望のラインに出力するデコーダと、他の算術論理演算回路との間で各ライン上のデータ出力の同期をとるためのラッチ回路とを有するものである。前記複数の算術論理演算回路の各々のデコーダにおける順方向ライン上のデータおよび逆方向ライン上のデータは、単位処理サイズのデータである。

【0013】

この演算装置にコントローラを備え、 n 行 m 列の前記インストラクションの集合であるインストラクションアレイをコントローラに保持しておき、コントローラで、インストラク

10

ションアレイを列毎に m 回前記複数の算術論理演算回路宛に出力するとともにパイプラインが後段に1段ずつ n 回移行させる際に、実行済みのインストラクションを1段毎に1つ減ずるようにしてもよい。

【0014】

また、前記複数の算術論理演算回路の各々のラッチ回路は、後段に出力する第1の出力データをラッチする第1のデータラッチ回路と、後段に出力する第2の出力データをラッチする第2のデータラッチ回路と、前段に出力するデータをラッチするリバースラッチ回路と、演算結果のキャリーを次の桁の演算のためにラッチするキャリーラッチ回路と、第

20

【0015】

この演算装置は、前記切替接続手段、前記複数のシリアル演算パイプライン、および前記インストラクション受付機構を一つの半導体デバイスの中に組み込んでおき、用途に応じて切替接続手段の制御信号とインストラクションアレイを入力することにより、用途の汎用性を持たせることができる。

【0016】

本発明は、シリアル演算パイプラインによる演算方法を提供する。この方法は、後段にデータを出力するための1又は複数の第1ラインと、前段にデータをフィードバックするための第2ラインとを有し、シリアル演算を行うとともに演算結果を表すデータの出力対象となるラインを各々独立に選択することができる算術論理演算回路を複数段カスケード接続してシリアル演算パイプラインを構築し、前記シリアル演算パイプラインに接続された外部コントローラで、前記複数の算術論理演算回路による1パスによる同時実行の手順を反映した n 行 m 列のインストラクションの集合であるインストラクションアレイを列毎に m 回前記複数の算術論理演算回路宛に個別に出力するとともに、パイプラインが後段に1段ずつ n 回移行させる際に、実行済みのインストラクションを1段毎に1つ減ずる制御を行うことを特徴とする演算方法である。

30

40

【0017】

【発明の実施の形態】

本発明の実施形態を説明するに当たり、図1を参照して、本発明の基本原理について説明する。

シリアル演算器は、例えば1~3ビット程度のシリアル演算を行うもので、16ビット程度の多ビット演算を行うパラレル演算器と比較して、スループットが、データ幅分の1(1ビットのシリアル演算の場合)となるが、演算器面積もその分小さい。スループットが小さい分をパイプラインの並列度を上げるようにすれば、両者の製造コストは実質的に変わらない。

例えば、図1(a)のように、16ビットレジスタを二つパラレルにし、その出力を一つ

50

の16ビットアダーで加算して、加算結果を一つの16ビットレジスタに格納する構成の16ビット演算器と、(b)のように1ビットレジスタを32個並列にし、それぞれ2つの1ビットレジスタの出力を加算する1ビット加算器を16個設け、各々の加算機の出力を格納する16個の1ビットレジスタからなる16個の1ビット演算器とは実質的に同一コストとなる。後者の場合は、パイプの各ステージが短くなるため、その分高速化が可能になる利点もある。

【0018】

この実施形態では、以上の原理に基づき、様々な多ビット演算命令の実行環境を、単位処理サイズのフィードバック付シリアル算術論理演算回路 (Serial-ALCell、以下「SALC」と称する) の組み合わせによって実現する場合の例を挙げる。単位処理サイズとは、SALCにおいて一つのクロックで処理可能な最低ビット長であり、最も単純な例では1ビット、上記のように、通常は1~3ビット程度である。

【0019】

<SALC>

SALCの構成例を図2に示す。図2からわかるように、この実施形態によるSALC1は、2系統のデータ入力端D1i、D2iおよび2系統のデータ出力端D1o、D2oと、1系統のリバースデータ入力端Riおよびリバースデータ出力端Roとを有し、前段(図の左側)から後段(図の右側)へ2系統のデータを出力するための順方向ラインと、後段から前段へデータをフィードバックするための逆方向ラインとが形成されている。

以後の説明では、データ出力端D1oから出力されるライン上のデータを「出力データ」、データ出力端D2oから出力されるライン上のデータを「参照データ」、リバースデータ出力端Roから出力(フィードバック)されるライン上のデータを「リバースデータ」とする。

SALC1において、後述するシリアル演算パイプラインを構成する他のSALCとの同期をとるため、出力データはデータラッチ回路11でラッチされ、参照データはリファレンスラッチ回路12でラッチされ、リバースデータはリバースラッチ回路12でラッチされる。

【0020】

SALC1は、また、インストラクション入力端CONから入力されるインストラクションの内容をデコードし、デコード結果に応じた処理を実行するとともに、実行結果を出力するラインの選定等を行うためのデコーダ10を有する。処理の例としては、例えば、四則演算、論理演算のような演算処理のほか、パス制御、ラッチ制御、条件付命令等の制御処理が挙げられる。デコーダ10は、また、前段から受け取ったデータが後段の方向に向かう順方向ラインと、後段から受け取ったデータが前段の方向に向かう逆方向ラインとの選択も行う。他の複数のSALC1のデコーダとの協働によって、マクロ命令の実行を行うことができる。

【0021】

デコーダ10には、上記の演算等を容易にするための種々のラッチ回路、すなわち、シフトラッチ回路14と、コントロールラッチ回路15と、キャリーラッチ回路16とが接続されている。

シフトラッチ回路14は、出力データのラインよりも参照データのラインが所定時間だけ遅れるようにするために、参照データをラッチし、例えば演算の際に次の桁でこれを出力するように動作する。キャリーラッチ回路15は、演算結果のキャリーを次の桁の演算時までラッチする。コントロールラッチ回路15は、出力データ、参照データ、リバースデータの出力タイミングをコントロールするコントロールデータをデコーダ10から出力の指示が出されるまでラッチする。

【0022】

以上のように構成されるSALC1では、デコーダ10によって順方向と逆方向の入れ替えができるため、複数のSALC1をカスケード接続してシリアル演算パイプラインを構築した場合に、順方向ラインと逆方向ラインとを適宜セレクトしてフレキシブルなデータ

10

20

30

40

50

レンジで論理演算を行うことができる。

また、乗算器や除算器、積和演算器、フロート演算器（指数・仮数を分けて演算し、それらの結果を互いに反映させる）などを容易に構築できるようになる。

【0023】

<SALP>

次に、上記のSALC1を用いたシリアル演算パイプラインの実施形態を説明する。

図3は、複数のSALC1をカスケード接続してなるシリアル演算パイプライン（Serial-ALPipeline、以下、「SALP」と称する）の構成例を示した図である。SALP2は、様々な多ビット演算命令を単純なシリアル命令の組み合わせで実現するものである。なお、1パスで論理演算を完了させる場合、SALC1の連結数は、加算器では1以上あれば済むが、乗算器では、桁数分のSALC1が必要となる。さらに、多項式演算などを行わせる場合は、同時演算分やデータラッチ分（これらは共通化できる場合がある）が必要である。

10

【0024】

多項式を実現する場合は、SALP2を図4のように、SALC1の出力を数段前のSALC1のデータ入力端にフィードバックすればよい。これにより、前に演算した結果と新たに投入するデータ間で、論理演算ができるようになり、各々の中間値（入出力間の中間のデータ）同士を演算させるなどの複雑な多項式を1パスで演算することができるようになる。

【0025】

SALP2では、複数のSALC1の協働によって種々のマクロ命令を実行することができる。この場合の動作例を具体的に説明する。ここでは、単純な例として、4ビット乗算（Multiplier）の例を挙げる。

20

【0026】

図5～図7は、4つのSALC1をカスケード接続したSALP2における各SALC1の状態とそのときのデータの内容を示した図である。

4ビット乗算は、図5(a)の0サイクルで最初のデータが入力され、1クロック毎に次のデータの入力・インストラクションのデコード・論理演算・ラッチ・出力が行われた後、図7(p)の16サイクル目で完結する。図中、個々のSALC1のうち面積が拾い部分はデコーダ10であり、デコーダ10の上部はキャリーラッチ回路16であり、デコーダ10の下部左側はシフトラッチ回路14、下部右側はコントロールラッチ回路15である。各々、空白になっている部分および破線部分にはデータ等が存在しないことを表している。0, A0～A3, B0～B3は演算対象データ、SM0～SM7は演算結果データである。デコーダ10の部分に記入されている文字はインストラクションであり、「SET」はデータセット（data set: シフトラッチ回路14、コントロールラッチ回路15へのデータ格納）、「THR」はスルー（through: データを格納せず、そのまま通過させる）、「ADS」はデータ加算&シフト（add&shift）を表している。「carry」はキャリーラッチ回路16にラッチされるキャリービットを表している。

30

【0027】

最初のサイクル（0サイクル）で最初のデータ（0, A0）が初段のSALC1に入力され、デコーダ10でインストラクション解読および論理演算がなされた後（スルーを含む）、演算結果データ（スルーを含む）が後段のSALC1に出力される。この動作をすべてのSALC1に共通のクロックタイミングで同時に入力されるインストラクションに従って繰り返し、最終的に4ビット乗算の結果が最終段のSALC1からSM1～SM7として出力される。

40

【0028】

マクロ命令の中には、フロート加算（Float Add）のような特殊な演算形態をとるものもある。この場合は、各SALC1に入力するインストラクションを変えるだけで、容易にそれを実現することができる。因みに、フロート加算において用いるインストラクションは、上記のADSのほか、FSUB（subtract for float）、SS（subtract & set）

50

、C C H G (controlled change)、C S F T (controlled shift)、I N V (inverse)、R E V (reverse)、S F T (shift)、O R、S U B (subtract&envset)、N S U B (subtract&RESET)を用いる。なお、上述したインストラクションの種類は例示であって、その他にもデコーダ10で実行可能な処理の種類に応じて任意に選定することができる。

【0029】

本実施形態のS A L P 2は、S A L C 1を基本構成とし、その連結状態を適宜変えながらフレキシブルなデータレンジで論理演算を行うことができるので、データレンジに比例した処理性能を実現することができる(例えば、8ビット演算は16ビット演算の倍の性能)。また、入出力間の中間データなどのビットを膨らませ、精度を保つことができるようになる。また、処理途上の上位・下位ビットを削除することで、データの質に合わせ効率的な有効桁数を持てるようになる。さらに、指数、仮数を別処理することで、フロート(float)演算が可能となる。

上述したように、S A L P 2では、パラレル演算パイプラインに比べてスループットがデータ幅分の1となるが、演算器面積もその分少ない。スループットが減少した分をパイプラインの並列度に割り振れば、原理的に(制御を考えなければ)コストは変わらない。その上、パイプ各ステージは短くなるため、その分を高速化できる。

【0030】

本実施形態のS A L P 2では、割り算、飽和演算、積和演算、多項式など、ほとんどの論理演算を実行することができる。単演算子としてのコストパフォーマンスをカスケード接続の演算器と比較すると、1要素の演算に対する比較では、スループットがデータ幅分の1であるためコスト上昇を伴うように見えるが、多量なデータに対する演算では、演算要素を並列に割り振ることによって同一コストとなる。また、クロック間のロジックは、単位処理サイズ(例えば1~3ビット)毎で切ることができ、また、いわゆるファンアウトが非常に小さいので、周波数を大幅に上げることができる。また、都合のいいことに、パイプラインの先頭に、非常に小さい構成のクロスバを接続することができる。これにより、演算要素の並列ラインの判定、パイプライン末尾ラインのビット判定を非常に小さい構成で実現することができる。アドレス計算、ブランチの条件を算出する等のスカラ処理、処理領域が小領域でパラメータがその領域ごとに変化するような処理に対しては、領域をバインドすることによって実現することができる。

そのため、このようなS A L P 2を複数用いることにより、効率的な論理演算を行う演算装置を構成することができる。

【0031】

<演算装置>

図8は、32個のS A L C (図1に示したS A L C 1)をカスケード接続したS A L P 2を256個含んでなる演算装置の構成図である。

各S A L P 2は、例えば外部コントローラ、あるいはネットワーク等と接続されるシリアルクロスバ(S C B)3に対して、並列に接続されている。S C B 3は、アドレスデコーダと複数のクロスバスイッチとを有し、小面積で256ラインの交換を行うもので、外部コントローラ等からのバスのアドレス投入(コントロールフェーズ)とデータ投入(データフェーズ)の2フェーズで、所望のラインにデータを分配することができる。

【0032】

256個の各々のS A L P 2には、インストラクションコントローラ(i C O N)4からn行m列のインストラクションの集合であるインストラクションアレイが入力され、このインストラクションアレイによって、論理演算およびデータの入出力ラインがそれぞれ独立に制御されるようになっている。

インストラクションアレイは、すべてのS A L P 2に共通内容のものを同時に入力してもよく、個々のS A L P 2のアドレスを指定したうえで共通内容のインストラクションアレイを同時に入力してもよく、個々のS A L P 2に個別的に個別内容のインストラクションアレイを入力してもよい。

10

20

30

40

50

i C O N 4 は、インストラクションアレイを列毎に m 回、各 S A L P 2 の個々の S A L C 宛に出力するとともにパイプラインが後段に n 回移行する際に、実行済みのインストラクションを減ずる。

【 0 0 3 3 】

このことを、1つの S A L P 2 の場合の例を挙げた図 9 により説明する。

図 9 の左側に破線ブロックで示されているインストラクションアレイは、i C O N 4 (図 8 参照) のメモリに保持されており、1クロック毎に、破線ブロックの右列のインストラクションから順次読み出されて S A L P 2 の S A L C に入力される。先頭のインストラクションが実行されると、そのインストラクションが 1 つ削除されて後段の S A L C に入力され、前段の S A L C には、次の列のインストラクションのすべてが入力される。これをインストラクションがなくなるまで繰り返す。

10

インストラクションアレイには、フェッチやデータループを埋め込むことができる。この場合には、従来技術において説明した複雑な D F G を 1 パスで記述することができるようになり、演算パイプライン動作の制御形態を簡略化させることができる。

【 0 0 3 4 】

次に、この演算装置の動作例を説明する。

ここでは、一例として、ある一つの S A L P 2 を用いて図 1 0 (a) に示す「1 0 1 0」(A データ) × 「1 1 0 0」(B データ) + 「0 0 1 0 1 1 0 1」(C データ) の乗算および加算を行う場合の例を挙げる。

出力される演算結果データは「1 1 0 0 0 1 0 1」となるのであるが、その演算過程が特徴的なので、最初のサイクルから演算結果データが出力されるまでの過程を図示して説明する。

20

用意されるデータは、図 1 0 (b) のように、2 段に並べられ、その右側から順次演算装置 (S A L P 2) に入力される。

なお、データサイズを合わせるために、データ取得時に M S B を 4 回取得し、これを A データの先頭に付加する。この付加したデータは、演算結果には影響を及ぼさないものである。図 1 0 (b) 右側のデータ説明において「intermediate data」は、演算過程で使用するいわゆる中間データであり、これも演算結果に影響を及ぼさないものである。

【 0 0 3 5 】

インストラクションアレイは、図 1 0 (c) のようなものが用意される。

30

この例において乗算と加算に使用するインストラクションは、「S E T」(データセット)、「A D S」(加算およびシフト)、「T H R」の 3 種類である。

「S E T」は、シフトラッチ回路 1 4 に参照データ (下段のデータ入力端に入力されたデータ)、コントロールラッチ回路 1 5 に出力データ (上段のデータ入力端に入力されたデータ)、キャリーラッチ回路 1 6 に「0」をセットすることを、その内容とする。

「A D S」はコントロールラッチ回路 1 5 にラッチされているデータが「0」の場合にはシフトのみ、それ以外は加算およびシフトを行うことを、その内容とする。シフトとは、参照データとしてシフトラッチ回路 1 4 にラッチされているデータを使用することをいう。加算のときには、キャリービットのチェンジが行われる (change carry)。

「T H R」は入力されたデータ (出力データおよび参照データ) をそのまま後段に送ることを、その内容とする。

40

【 0 0 3 6 】

最初のサイクルでは、図 1 1 に示すように、その先頭に「S E T」が割り当てられた 8 個インストラクションの列が初段の S A L C のインストラクション入力端 CON に入力され、また、図 1 0 (b) の最右列の 2 つのデータがデータ入力端 D1i、D2i の各々に入力される。インストラクションが「S E T」なので、これらの 2 つのデータ「0」、「0」が、S A L C のシフトラッチ回路 1 4 およびコントロールラッチ回路 1 5 にセットされる。キャリーラッチ回路 1 6 には、キャリービットが発生しないので、「0」がセットされる。

【 0 0 3 7 】

2 番目のサイクルでは、図 1 2 に示すように、次の列の 2 つのデータが初段の S A L C の

50

データ入力端D1i、D2iの各々に入力され、前のサイクルで入力された2つのデータ「0」、「0」がデータラッチ回路11およびリファレンスラッチ回路12から2段目のSALCに入力されるようになる。インストラクションは、前のサイクルで初段のSALCに入力されたもののうち既に実行された「SET」が除かれたものが2段目のSALCに出力され、初段のSALCには、図10(c)の次の列のインストラクションが入力される。初段および2段目のSALCのインストラクションとも「THR」なので、それぞれ入力された2つのデータ「0」、「0」は、後段のSALCにそのまま送られる。

【0038】

3番目のサイクルでは、図13に示すように、さらに次の列の2つのデータが初段のSALCのデータ入力端D1i、D2iの各々に入力される。インストラクションは、初段のSALC以外のSALCには、2番目のサイクルで入力されたもののうち既に実行された分が除かれたものが入力され、初段のSALCには、図10(c)のさらに次の列のインストラクションが入力される。初段および3段目のSALCのインストラクションは「THR」、2段目のSALCのインストラクションは「SET」なので、2段目のSALCに入力されたデータ「0」、「0」は、そのSALCのシフトラッチ回路14およびコントロールラッチ回路15にセットされ、初段および3段目のSALCについては、それぞれ入力された2つのデータ「0」、「0」が後段のSALCにそのまま送られる。4番目のサイクルでの動作は図14のようになり、上記と同様の手順でSALCによる演算実行およびデータ移動が行われる。

【0039】

5番目のサイクルでは、図15に示すように、「ADS」のインストラクションが初段のSALCが入力される点以外は、上記と同様の手順でSALC間による演算実行およびデータ移動が行われる。初段のSALCでは、コントロールラッチ回路15にラッチされているデータが「0」なので、シフトのみとなる。6番目のサイクル(図16)~11番目のサイクル(図22)についても、上記と同様の手順で、SALCによる演算実行およびデータ移動が行われる。

【0040】

12番目のサイクル(図23)からは、初段のSALCに入力されるデータがなくなり、前段のSALC(図23の場合は初段のSALC)のデータラッチ回路11およびリファレンスラッチ回路12にラッチされているデータが後段のSALC(図23の場合は2段目のSALC)に入力されるようになる。インストラクションについては、上記と同様、実行された分が削除されて、後段のSALCに入力される。最終段のSALCのデータラッチ回路11にラッチされているデータが最終演算結果を表すデータのLSBとなる。

【0041】

13番目のサイクル(図24)から20番目のサイクル(図31)まで、上記の動作を繰り返し、それぞれ最終段のSALC1のデータラッチ回路11にラッチされているデータを取り出すことにより、図31の上段に示される最終演算結果が得られる。

【0042】

このように、本実施形態の演算装置では、フィードバック機能付きのSALCが複数段カスケード接続されたSALP2をSCB3に対して複数ライン並列に接続し、個々のSALCによる演算動作をiCON4からのインストラクションアレイによって個別的に制御できるようにしたので、多種多様な論理演算(四則演算およびその組合せ、飽和演算、積和演算、多項式等)を共通のハードウェアによって容易かつフレキシブルに実現することができる。そのため、論理演算の用途が変わったときでもコストの増加を伴わずにそれに対応することができる。

【0043】

また、個々のSALC1がフィードバック機能を有しているので(ひいては、SALP2がそのような機能を有しているので)、様々なデータサイズに対して突発的かつ爆発的な演算量を伴う複雑な処理フローを実現する汎用的なシリアル演算パイプラインを容易に構築することができる。

10

20

30

40

50

【 0 0 4 4 】

また、インストラクションアレイの中に、データの並び、その実行手順等を任意に組み込むことができるので、データパスを単純（一直線）にすることができる利点がある。

【 0 0 4 5 】

上記の演算装置は、入出力データに対してインストラクションの量が相対的に多い論理演算に対しては、特に有効なツールとなり得る。この演算装置は、例えばテキストチャマッピングのような複雑な演算を何らのハードウェアを代えずに実施することができる。この場合、テキストチャが1フラグメントに収まらない場合は、周囲フラグメントが複数回投入される。

【 0 0 4 6 】

演算装置は、また、複数の関数を定義したルックアップテーブルと、アドレスサイクル分のスルーブットでルックアップテーブルの内容を書き換える機能とを有し、様々な関数を多重に演算することもできる。

【 0 0 4 7 】

【 発明の効果 】

以上の説明から明らかなように、本発明によれば、様々なデータサイズに対して突発的かつ爆発的な演算量を伴う複雑な処理フローを実現する汎用的なシリアル演算パイプラインを構築することができる。

【 図面の簡単な説明 】

【 図 1 】 本発明の原理を示す図で、(a) は1つの16ビット演算器、(b) は16個の1ビット演算器の例を示す。

【 図 2 】 本発明の一実施形態による算術論理演算回路 (SALC) の構成図。

【 図 3 】 本発明の一実施形態によるシリアル演算パイプライン (SALP) の構成図。

【 図 4 】 多項式を実行するときの S A L P の構成例を示した図。

【 図 5 】 (a) ~ (f) はシリアル演算パイプラインの動作の一例を示す説明図。

【 図 6 】 (g) ~ (l) はシリアル演算パイプラインの動作の一例を示す説明図。

【 図 7 】 (m) ~ (p) はシリアル演算パイプラインの動作の一例を示す説明図。

【 図 8 】 本発明の演算装置の一実施形態を表す構成図。

【 図 9 】 本発明の演算装置に含まれるシリアル演算パイプライン (SALP) の動作概要図。

【 図 1 0 】 (a) は本発明の演算装置の動作例を説明するための論理演算式、(b) は入力するデータ列の構造説明図、(c) は入力するインストラクションアレイの構造説明図

【 図 1 1 】 本発明の演算装置の動作状態説明図 (最初のサイクル) 。

【 図 1 2 】 本発明の演算装置の動作状態説明図 (2 番目のサイクル) 。

【 図 1 3 】 本発明の演算装置の動作状態説明図 (3 番目のサイクル) 。

【 図 1 4 】 本発明の演算装置の動作状態説明図 (4 番目のサイクル) 。

【 図 1 5 】 本発明の演算装置の動作状態説明図 (5 番目のサイクル) 。

【 図 1 6 】 本発明の演算装置の動作状態説明図 (6 番目のサイクル) 。

【 図 1 7 】 本発明の演算装置の動作状態説明図 (7 番目のサイクル) 。

【 図 1 8 】 本発明の演算装置の動作状態説明図 (8 番目のサイクル) 。

【 図 1 9 】 本発明の演算装置の動作状態説明図 (9 番目のサイクル) 。

【 図 2 0 】 本発明の演算装置の動作状態説明図 (10 番目のサイクル) 。

【 図 2 1 】 本発明の演算装置の動作状態説明図 (11 番目のサイクル) 。

【 図 2 2 】 本発明の演算装置の動作状態説明図 (12 番目のサイクル) 。

【 図 2 3 】 本発明の演算装置の動作状態説明図 (13 番目のサイクル) 。

【 図 2 4 】 本発明の演算装置の動作状態説明図 (14 番目のサイクル) 。

【 図 2 5 】 本発明の演算装置の動作状態説明図 (15 番目のサイクル) 。

【 図 2 6 】 本発明の演算装置の動作状態説明図 (16 番目のサイクル) 。

【 図 2 7 】 本発明の演算装置の動作状態説明図 (17 番目のサイクル) 。

【 図 2 8 】 本発明の演算装置の動作状態説明図 (18 番目のサイクル) 。

10

20

30

40

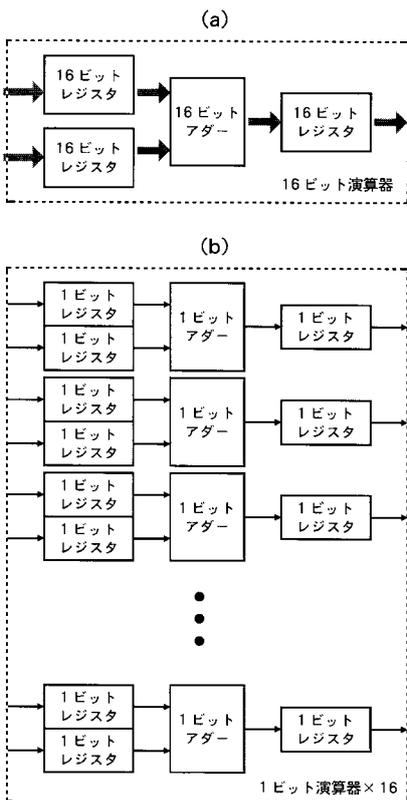
50

【図29】本発明の演算装置の動作状態説明図（19番目のサイクル）。
 【図30】本発明の演算装置の動作状態説明図（20番目のサイクル）。
 【図31】本発明の演算装置の動作状態説明図（21番目のサイクル）。

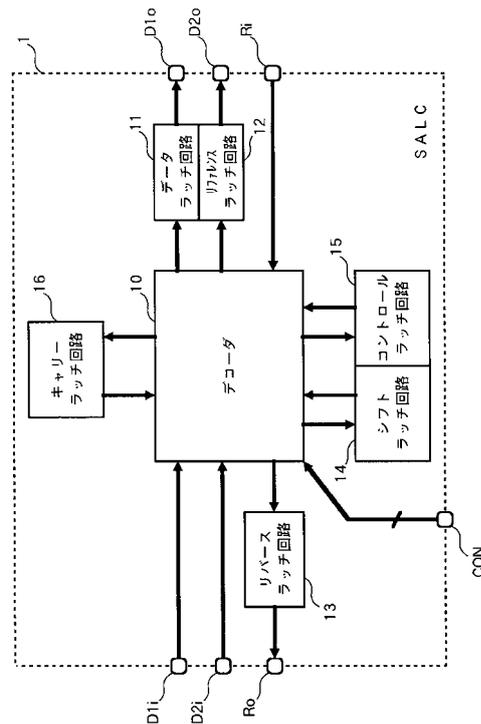
【符号の説明】

- 1 算術論理演算回路 (SALC)
- 10 デコーダ
- 11 データラッチ回路
- 12 リファレンスラッチ回路
- 13 リバースラッチ回路
- 14 シフトラッチ回路
- 15 コントロールラッチ回路
- 16 キャリーラッチ回路
- 2 シリアル演算パイプライン (SALP)
- 3 シリアル・クロスバ (SCB)
- 4 インストラクションコントローラ (iCON)

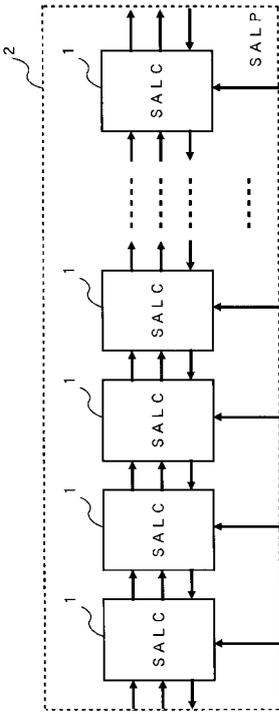
【図1】



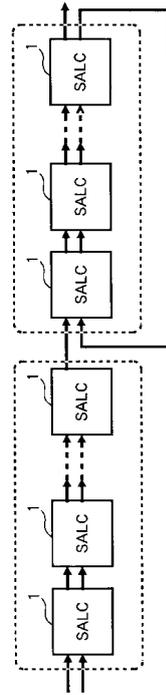
【図2】



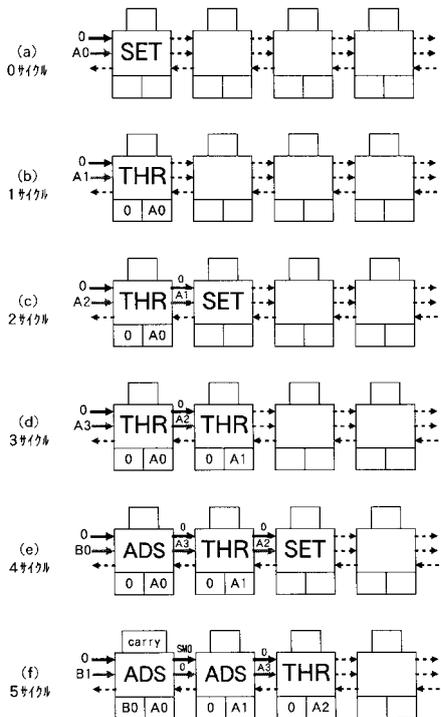
【 図 3 】



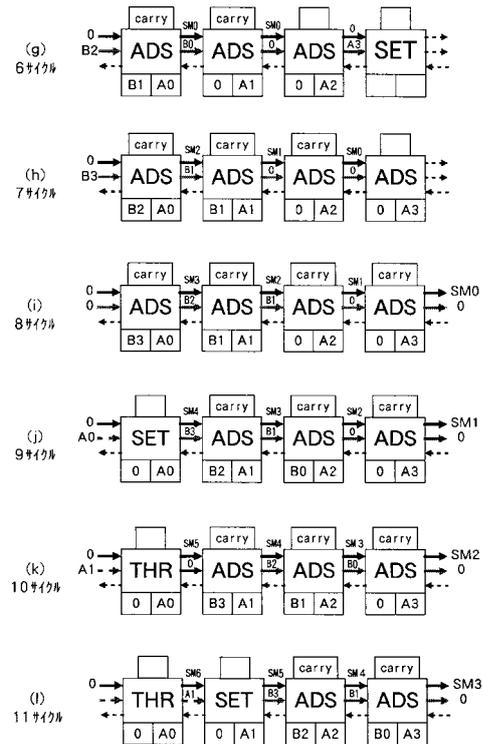
【 図 4 】



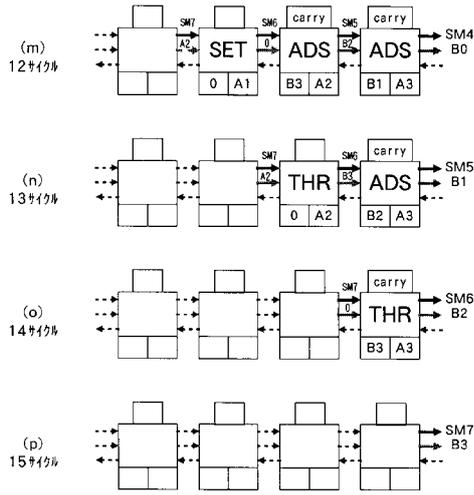
【 図 5 】



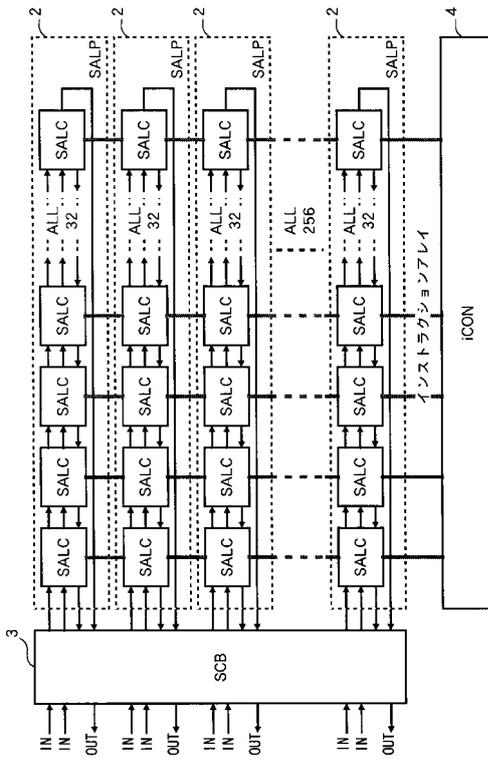
【 図 6 】



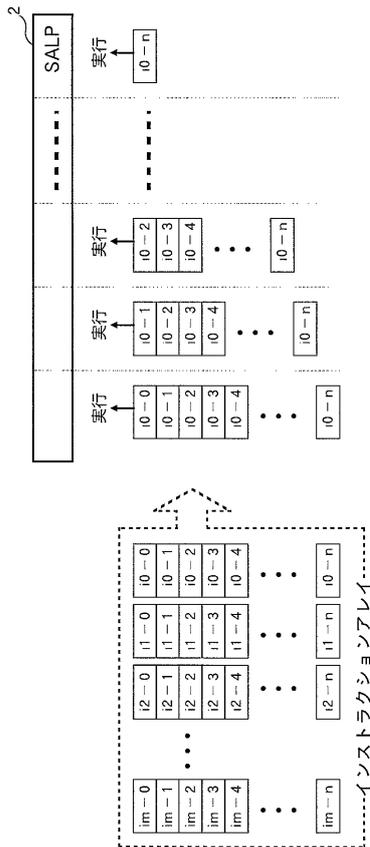
【 図 7 】



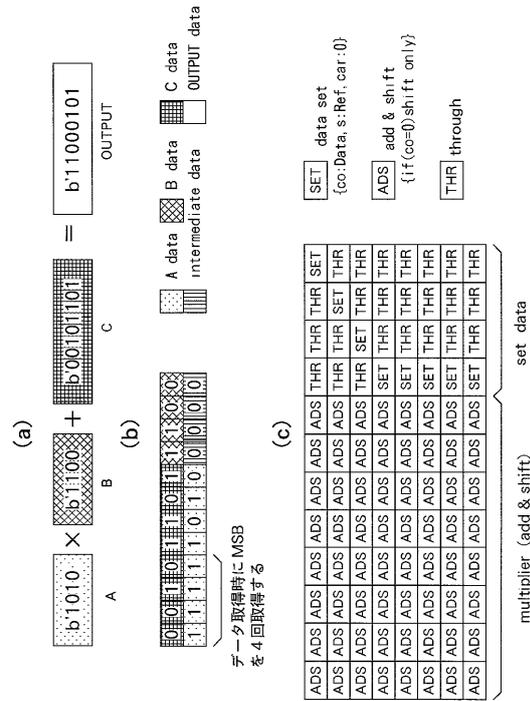
【 図 8 】



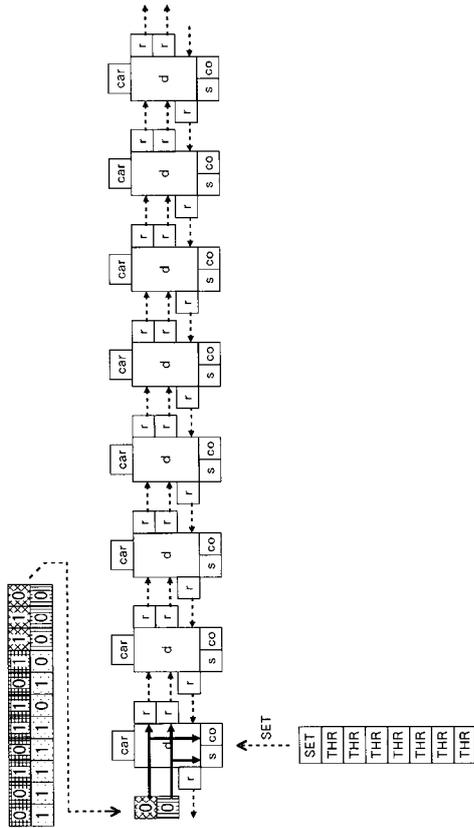
【 図 9 】



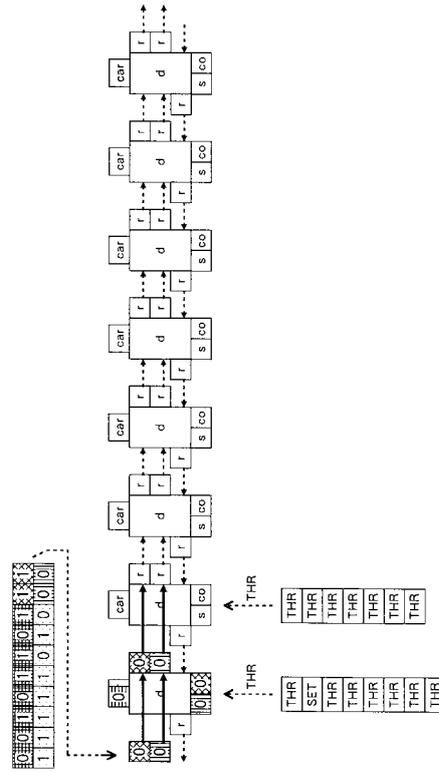
【 図 10 】



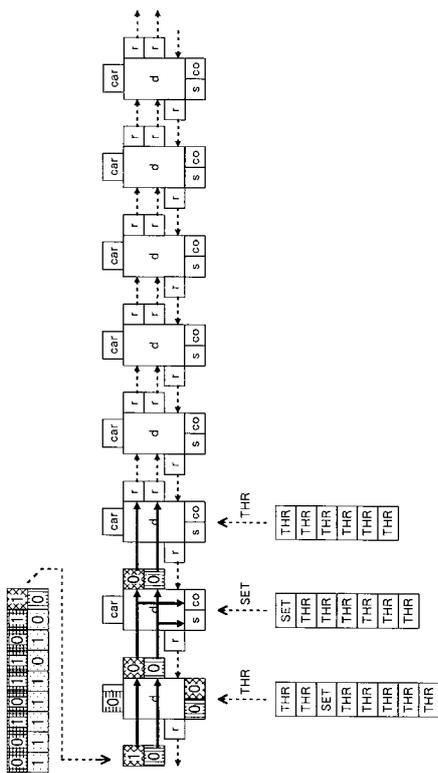
【 図 1 1 】



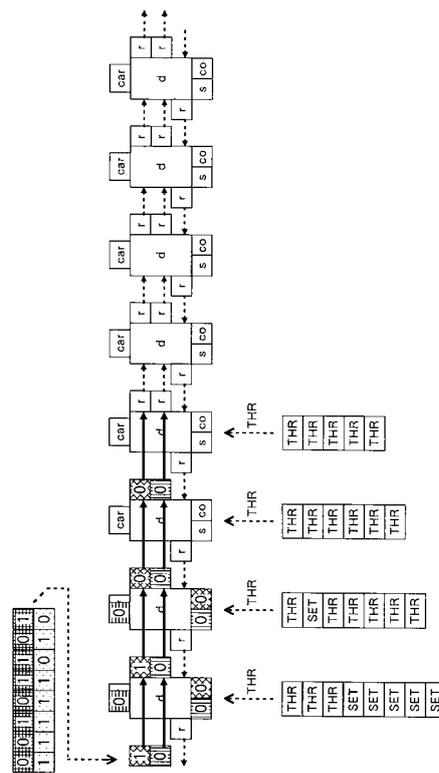
【 図 1 2 】



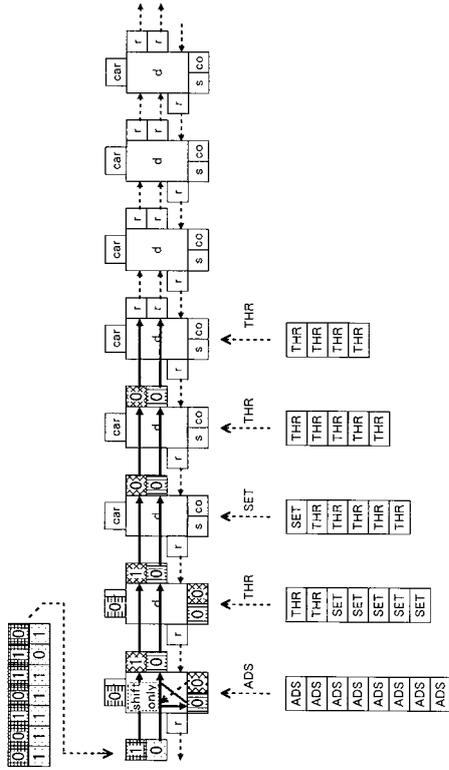
【 図 1 3 】



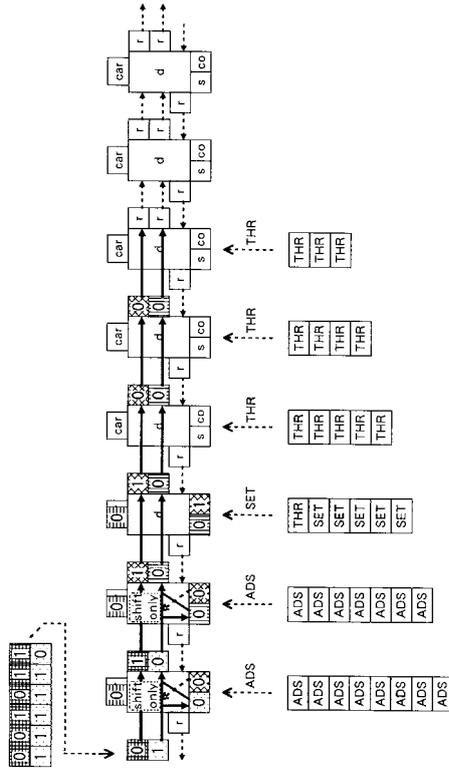
【 図 1 4 】



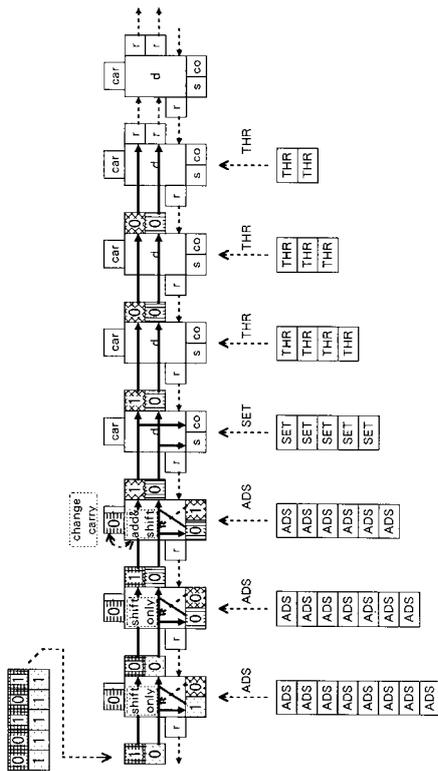
【 15 】



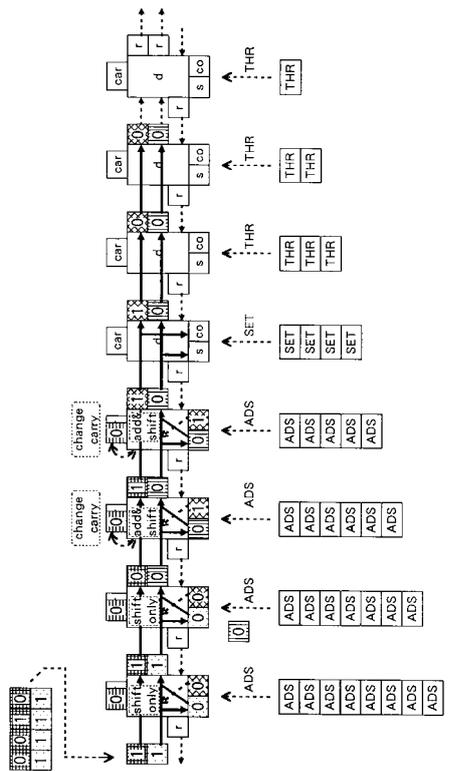
【 16 】



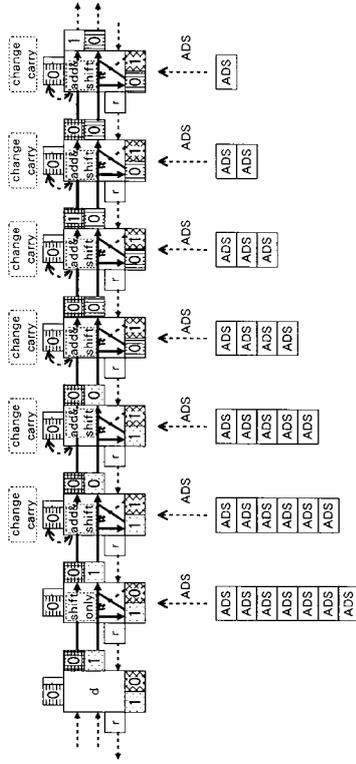
【 17 】



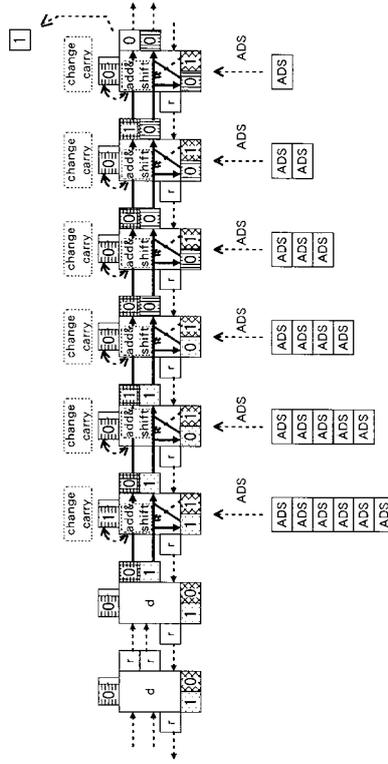
【 18 】



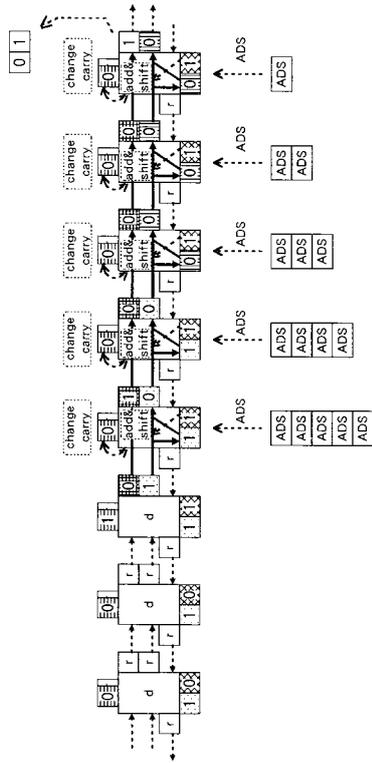
【 2 3 】



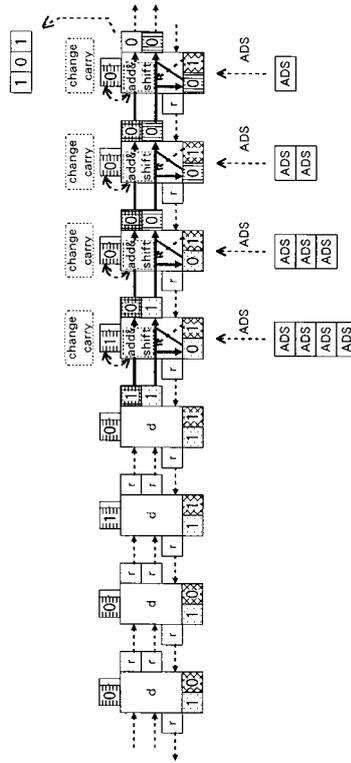
【 2 4 】



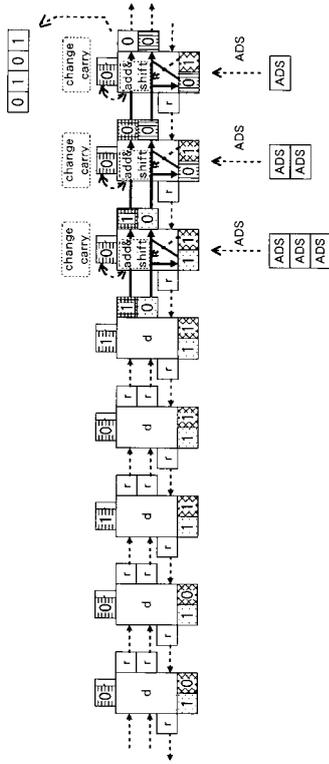
【 2 5 】



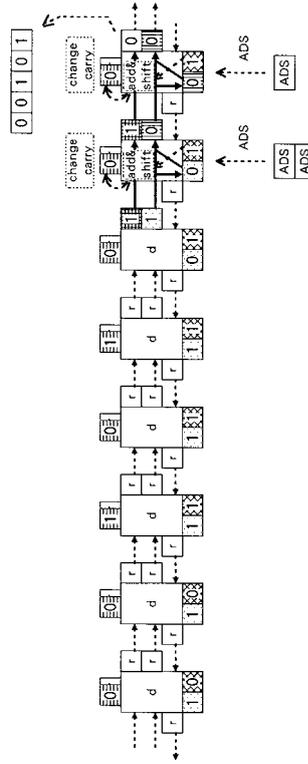
【 2 6 】



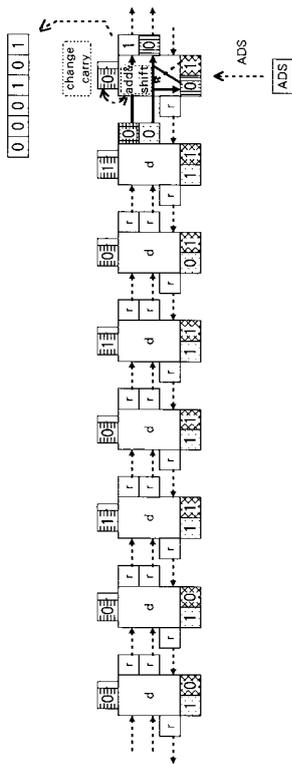
【 27 】



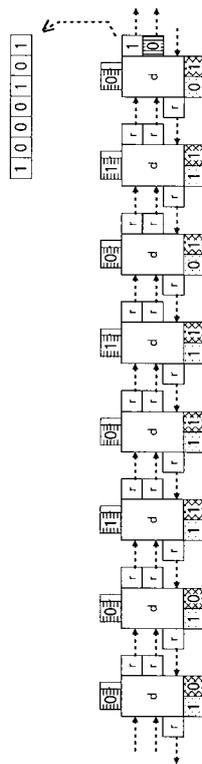
【 28 】



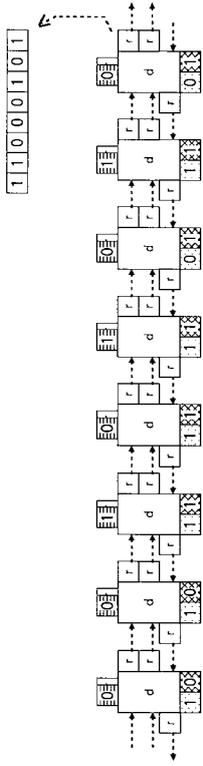
【 29 】



【 30 】



【 図 3 1 】



フロントページの続き

審査官 田中 友章

- (56)参考文献 特開平 1 1 - 2 1 9 2 7 9 (J P , A)
特開平 0 7 - 1 1 4 4 5 5 (J P , A)
特開平 0 4 - 2 4 5 5 3 3 (J P , A)
特開昭 5 6 - 0 1 9 1 5 2 (J P , A)

(58)調査した分野(Int.Cl.⁷, D B名)

G06F 7/00
G06F 9/30 370
G06F 9/38 310