(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3573506号
(P3573506)
(45)発行日　平成16年10月6日(2004.10.6)　(24)登録日　平成16年7月9日(2004.7.9)

| (51)Int.Cl.[7] | F I | | |
|---|---|---|---|
| G06F 9/38 | G06F | 9/38 | 370A |
| G06F 7/60 | G06F | 9/38 | 330J |
| G06F 9/32 | G06F | 7/60 | Z |
| | G06F | 9/32 | 320F |

請求項の数 27　(全 19 頁)

(21)出願番号　特願平6-308736
(22)出願日　平成6年12月13日(1994.12.13)
(65)公開番号　特開平7-244589
(43)公開日　平成7年9月19日(1995.9.19)
審査請求日　平成13年5月31日(2001.5.31)
(31)優先権主張番号　166582
(32)優先日　平成5年12月13日(1993.12.13)
(33)優先権主張国　米国(US)

(73)特許権者　398038580
ヒューレット・パッカード・カンパニー
HEWLETT－PACKARD COMPANY
アメリカ合衆国カリフォルニア州パロアルト　ハノーバー・ストリート　3000
(74)代理人　100063897
弁理士　古谷　馨
(74)代理人　100076680
弁理士　溝部　孝彦
(74)代理人　100087642
弁理士　古谷　聡
(72)発明者　マイケル・シュランスカー
アメリカ合衆国カリフォルニア州94024ロス・アルトス，ラ・プレンダ・409
最終頁に続く

(54)【発明の名称】述語、及びブール式を解くためのコンピュータ・システム、及び方法

(57)【特許請求の範囲】
【請求項１】
コンピュータ・システムにおいてブール式及び述語を解く方法であって、該コンピュータ・システムが、命令セットの命令を復号する命令ユニットを備えるプロセッサ、その復号された命令で指定された演算を実行する機能ユニット、１組のレジスタを備えるものにおいて、
前記命令ユニットで、前記命令セットから選択されるプログラムの命令を復号するステップと、
前記機能ユニット内で、前記復号された命令によって指定された演算を実行するステップと、
からなり、
少なくとも１つの機能ユニットのいずれか１つで、前記復号された命令の１つによって指定された対応する還元演算の実行が、
還元演算に対する少なくとも１つの命令指定された入力値を検索するステップ、
第１の命令指定された状態を有するブール結果値を生成するステップ、
前記還元演算の１以上の前記入力値の命令指定された条件関数を実行し、前記入力値の少なくとも１つの組合せに関して偽であるブール条件値を生成するステップ、
前記ブール条件値が第２の命令指定された状態であるとき、前記レジスタの組の命令指定された位置に前記結果値を条件付きで書き込むステップ、
からなることを特徴とする方法。

【請求項２】
前記還元演算がＡＮＤ還元演算であり、このＡＮＤ還元演算に対して、前記第１の命令指定された状態が真である、請求項１記載の方法。
【請求項３】
前記還元演算がＯＲ還元演算であり、このＯＲ還元演算に対して、前記第１の命令指定された状態が偽である、請求項１記載の方法。
【請求項４】
前記還元演算が補数の還元演算であり、この補数の還元演算に対して、前記第２の命令指定された状態が偽である、請求項１記載の方法。
【請求項５】
前記還元演算が非補数の還元演算であり、この非補数の還元演算に対して、前記第２の命令指定された状態が真である、請求項１記載の方法。
【請求項６】
前記還元演算が複数出力の還元演算であり、この複数出力の還元演算を実行するステップが、
前記ブール条件値が前記第２の命令指定された状態である場合、前記レジスタの組の複数の命令指定された位置に前記結果値を条件付きで書き込むステップからなる、請求項１記載の方法。
【請求項７】
前記命令指定された条件関数が、前記還元演算の少なくとも２つの入力値のブール関数である、請求項１記載の方法。
【請求項８】
前記少なくとも１つの入力値がブール述語値を含み、前記還元演算を実行するステップが、
前記ブール述語値の状態で前記還元演算の実行を述語化するステップからなる、請求項１記載の方法。
【請求項９】
第１の演算、及び前記プログラムの命令の部分集合に対する２以上からなる１組の還元演算を実行し、ブール式を解くステップからなり、前記還元演算を実行するステップが、
前記第１の還元演算を実行して、前記１組のレジスタの命令指定された位置に第１の状態を有するブール値を書き込むステップと、
前記還元演算の組を実行するステップと、からなり、この還元演算の組のそれぞれが、前記演算の前記条件関数によって生成された前記ブール条件値が前記命令指定された状態である場合に、前記指定された位置に第２の状態を有するブール値を書き込み、それによって前記指定された位置に記憶された前記値がブール式の値である、請求項１記載の方法。
【請求項１０】
前記還元演算の組がＡＮＤ還元演算であり、このＡＮＤ還元演算に対して、前記第１の命令指定された状態が真であり、前記第１の演算によって書き込まれた値が偽である、請求項９記載の方法。
【請求項１１】
前記還元演算の組がＯＲ還元演算であり、このＯＲ還元演算に対して、前記第１の命令指定された状態が偽であり、前記第１の演算によって書き込まれた値が真である、請求項９記載の方法。
【請求項１２】
前記還元演算の組の少なくとも１つが補数の還元演算であり、この補数の還元演算に対して、前記第２の命令指定された状態が偽である、請求項９記載の方法。
【請求項１３】
前記還元演算の組の少なくとも１つが非補数の還元演算であり、この非補数の還元演算に対して、前記第２の命令指定された状態が真である、請求項９記載の方法。
【請求項１４】

前記還元演算の組の少なくとも１つが複数出力の還元演算であり、この複数出力の還元演算を実行するステップが、
前記ブール条件値が前記第２の命令指定された状態である場合、前記レジスタの組の複数の命令指定された位置に前記複数出力の還元演算の結果値を条件付きで書き込むステップからなる、請求項９記載の方法。
【請求項１５】
前記還元演算の組の少なくとも１つの還元演算の前記命令指定された条件関数が、前記還元演算の少なくとも２つの入力値のブール関数である、請求項９記載の方法。
【請求項１６】
前記還元演算の組の少なくとも１つの還元演算の前記少なくとも１つの入力値が、ブール述語値を含み、前記還元演算を実行するステップが、
前記ブール述語値の状態で前記還元演算の実行を述語化するステップからなる、請求項９記載の方法。
【請求項１７】
復号化命令で指定される演算を同時に実行する複数の機能ユニットを備えるプロセッサ、及び２以上の機能ユニットによって同時に書き込まれるとき、画定された状態を有する値を記憶することができる前記レジスタの組の少なくとも１つのレジスタを有するコンピュータシステムにおける方法であって、この方法が前記機能ユニット内の前記還元演算の組を同時に実行するステップからなる、請求項９記載の方法。
【請求項１８】
値を記憶するための複数のレジスタと、
レジスタに結合され、コンピュータ・システムに対する命令セットの命令に対応する複数の演算を実行するように作動する機能ユニットと、この演算が還元演算からなることと、
前記機能ユニットを制御し、プログラムの命令に応答して演算を実行する命令ユニットと、
前記機能ユニットが以下のステップを実施することによって前記還元演算を実行することと、このステップが、
前記還元演算に対する、少なくとも１つの命令指定された入力値を検索するステップと、
第１の命令指定された状態を有するブール結果を生成するステップと、
前記演算の１以上の前記入力値の命令指定された条件関数を実行し、前記入力値の少なくとも１つの組合せに対して偽であるブール条件値を生成するステップと、
前記ブール条件値が第２の命令指定された状態であるとき、前記レジスタの組の命令指定された位置に、前記結果値を条件付きで書き込むステップとからなることと、
からなるコンピュータ・システム。
【請求項１９】
前記レジスタに結合され、コンピュータ・システムに対する命令セットの命令に対応する複数の演算を実行するようにそれぞれ作動する複数の機能ユニットと、この演算が還元演算からなることと、
複数のレジスタが、前記機能ユニットの２以上によって同時に書き込まれる複数の値を受け入れるレジスタからなり、このレジスタが、当該レジスタに書き込まれた前記ブール値のそれぞれが同じであるとき、前記レジスタに同時に書き込まれる複数のブール値の１以上に基づく値を記憶するように作動することとからなる、請求項１８記載のコンピュータ・システム。
【請求項２０】
前記レジスタに結合され、コンピュータ・システムに対する命令セットの命令に対応する複数の演算を実行するようにそれぞれ作動する複数の機能ユニットと、この演算が還元演算からなることと、
前記レジスタの少なくとも１つが、複数の同時に書き込まれた値を受け入れるための複数の入力ポートを備え、この入力ポートが、値と書き込み可能信号を各機能ユニットから受け取るために、１対１の関係で前記機能ユニットに接続され、前記機能ユニットの書き込

み可能信号が設定されると、機能ユニットから受け取った前記値が前記レジスタに書き込まれることと、
前記機能ユニットが、前記レジスタ内の位置を指定する還元演算を実行するとき、前記ブール条件値が前記第２の命令指定された状態である場合に、前記レジスタに前記ブール結果値を送り、前記書き込み可能信号を設定することと、
前記機能ユニットによって複数の同じ値が同時に前記レジスタに書き込まれるとき、前記値を導出し、前記位置に記憶する手段とからなる、請求項１８記載のコンピュータ・システム
【請求項２１】
前記還元演算がＡＮＤ還元演算であり、このＡＮＤ還元演算に対して、前記第１の命令指定された状態が真である、請求項１８記載のコンピュータ・システム。
【請求項２２】
前記還元演算がＯＲ還元演算であり、このＯＲ還元演算に対して、前記第１の命令指定された状態が偽である、請求項１８記載のコンピュータ・システム。
【請求項２３】
前記還元演算が補数の還元演算であり、この補数の還元演算に対して、前記第２の命令指定された状態が偽である、請求項１８記載のコンピュータ・システム。
【請求項２４】
前記還元演算が非補数の還元演算であり、この非補数の還元演算に対して、前記第２の命令指定された状態が真である、請求項１８記載のコンピュータ・システム。
【請求項２５】
前記還元演算が複数出力の還元演算であり、前記機能ユニットが、前記ブール条件値が前記第２の命令指定された状態である場合、前記レジスタの組の複数の命令指定された位置に前記結果値を条件付きで書き込むステップを実行することによって、該複数出力の還元演算を実行する、請求項１８記載のコンピュータ・システム。
【請求項２６】
前記命令指定された条件関数が、前記還元演算の少なくとも２つの入力値のブール関数である、請求項１８記載のコンピュータ・システム。
【請求項２７】
前記少なくとも１つの入力値がブール述語値を含み、前記機能ユニットが、前記ブール述語値の状態で前記還元演算の実行を述語化するステップを実行することによって、前記還元演算を実行する、請求項１８記載のコンピュータ・システム。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】
本発明は、一般的には、式を解くこと、及びコンピュータ・システムにおいてプログラムの実行を制御することに関し、より詳しくは、コンピュータ・システムでのブール還元（Ｂｏｏｌｅａｎ　Ｒｅｄｕｃｔｉｏｎ）を高速に解くこと、及び述語化実行のための構造的な機構、及び方法に関する。
【０００２】
【従来の技術】
プログラムは通常、多数のブール値を持った式を含む。この様な式によって生成される値は、データ値や分岐条件としての使用法や、コンピュータによっては、条件付き演算実行の述語としての使用を含む多数の使用法がある。
【０００３】
ブール値を持った式の一般的な応用例は、ブール還元を実行すること、即ち、ブール関数、例えばＡＮＤ、及びＯＲを使用して複数の値を単一の値に還元することである。ブール還元を実行するためのその様な式の一例は
ｒ＝ｕ∧¬ｖ∧ｗ∧ｘ（１）
である。

【０００４】
上式は、ＡＮＤ関数（∧で表される）を使用してｕ、ｖの補数（¬ｖで表される）、ｗ、及びｘのブール値を単一のブール値ｒに還元する。一般に、どんなブール式でも、多数のＡＮＤ還元とそれに続くＯＲ還元から成る、積和形で表すことができる。
【０００５】
通常、プログラムで使用されるブール値は、より小さい（＜）やより大きい（＞）等の比較演算や関係演算の結果として得られる。従って、以下のブール式は、コンピュータ・プログラムに用いられるＡＮＤ還元式として極めて典型的なものである。
ｒ＝（ａ＜ｂ）∧¬（ｃ＞ｄ）∧（ａ＜ｃ）∧（ｂ＞ｄ）（２）。
【０００６】
この種の還元式を解く速度は、プログラムを効率的に実行するために極めて重要である。このことは特に、超大型命令語（ＶＬＩＷ）やスーパースカラ・アーキテクチャ等の命令レベル並列処理を使用するコンピュータ・システム・アーキテクチャに当てはまる。ブール式は通常、分岐条件を計算する際に使用される。従って、ブール式を解くことは、プログラムのクリティカル・パスにあることが多い。又、並列処理（即ち、同時に実行できる演算群）を実施するための幾つかのプログラム変換がブール還元を導入している。その様な変換の効果は、これらの式をどれだけ速く解くことができるかに依存している。その様な変換の幾つかの例は、非ロール・ループから終了する複数の分岐を単一の分岐にまとめることと、非ロールｗｈｉｌｅループにおける制御依存性の高さ還元（ｈｅｉｇｈｔ－ｒｅｄｕｃｔｉｏｎ）が含まれる。
【０００７】
演算を並列に実行する機能がないコンピュータでは、ブール式は一般に、順次解く処理がなされる。即ち、還元式中の各演算は、一度に１つずつ実行される。一般に、ｎ個の比較演算を含む還元式を純粋に順次解いていくには、（２ｎ－１）α回のマシン・サイクルが必要である。ここで、αは、コンピュータ・システム上での１比較演算、又はＡＮＤ演算当たりのサイクル数である。例えば、上記還元式（２）は７α回のサイクルを必要とする。
【０００８】
命令レベル並列処理を伴うアーキテクチャにおいては、コンパイル時間高さ還元技法を使用してブール式を解くことを速くさせることが多い。速く解くための簡単な技法は、比較演算を並列に実行し、次いでＡＮＤ演算の２分木を使用して還元を実行することである。例えば、上記式（２）を検討する。コンピュータ・システム内のプロセッサが少なくとも４つの機能ユニットを有すると仮定すると、以下の表に示す様に式を解くことができる。
【０００９】
【表１】

| 高さ還元技法の例 | |
|---|---|
| サイクル | 命令 |
| 0 | $r_1 = a < b$；$r_2 = \neg(c > d)$；$r_3 = a < c$；$r_4 = b > d$； |
| α | $r_5 = r_1 \wedge r_2$；$r_6 = r_3 \wedge r_4$ |
| 2α | $r = r_5 \wedge r_6$ |
| 3α | 値ｒは使用可能である |

【００１０】
表１の例では、式（２）は簡単な高さ還元技法によって３α回のサイクルだけで解くことができる。一般に、この高さ還元技法では、ｎ個の比較演算を含む還元式は，コンピュータ・システムがｎ個の演算を並列に実行できるとすれば、（１＋（$\log_2 n$））α回のサイクルで解くことができる。しかし、更に並列処理を提供する資源が利用可能な時でも、この技法は少なくとも（１＋（$\log_2 n$））α回のサイクルを必要とする。

【００１１】
【発明が解決しようとする課題】
本発明は、コンピュータ・システムが十分な並列処理機能を有する場合、効果的にα回のサイクルだけでどんな還元式も解く機構、及び技法を提供する。この方法、及び技法による性能は、演算を並列に実行できる機能ユニット（資源）の数によってのみ制限され、式中の個々の演算間のどんな従属関係にも制限されない。更に、本発明によれば、演算を同時に実行することも、又どの様な所望の順序でも実行することができる。従って、例えば、コンパイラは自由に、比較演算とプログラム内の他の演算を並列に実行させることができる。これは、限られた資源をもつコンピュータ・システムにとって重要である。
【００１２】
【課題を解決するための手段】
本発明は、２つの主な態様を有する。本発明の第１の態様によれば、コンピュータ・システム内の１つ、又は複数のレジスタが、複数の演算によって書き込まれる全ての値が同じであれば、それらの演算が同時にそのレジスタに値を書き込むことを可能にする。その場合、そのレジスタに記憶される結果は、明確に定義されたものであり、レジスタに書き込まれるどの値とも等しい。しかし、複数の演算が同時に異なる値を１つのレジスタに書き込む場合、その結果記憶される値は未定義となる。書き込まれる値はブール値、整数値、浮動小数点値、その他の値でもよい。レジスタは、１ビット・レジスタでも、条件レジスタ、又は状況レジスタ内のビット位置でもよい。レジスタは又、汎用レジスタでも浮動小数点レジスタでもよい。コンピュータ・システム内の全レジスタが複数の同時書込みのためのこの機能を提供する必要があるわけではない。
【００１３】
本発明の第２の態様によれば、コンピュータ・システムは１組の還元演算を提供する。各還元演算の実行は一般に、演算の入力値の２つの関数、結果関数、及び許可関数によって定義される。結果関数は、値がある場合、どんな値が演算によって記憶されるかを決定する。許可関数は、その値が目標位置、又はレジスタに書き込まれ、あるいは記憶されるかどうかを決定する。従って、還元演算の結果$r_{out}$は

$$r_{out} = F_{out}(r_{in\ 1}, r_{in\ 2}, \cdots, r_{in\ n}) \quad \text{if } F_{en}(r_{in\ 1}, r_{in\ 2}, \cdots, r_{in\ n}) \quad (3)$$

で表すことができる。式中、$F_{out}$は結果関数であり、$F_{en}$は許可関数である。結果関数と許可関数は、少なくとも、整数入力値、又は浮動小数点入力値の比較関数、及びＡＮＤ、ＯＲ、逆関数、一致関数等のブール入力値の関数といったブール値関数を含むことができる。指定される目標レジスタは、複数の同時書込みを扱うタイプのものであることが好ましい。
【００１４】
本発明のこれら２つの態様によるコンピュータ・システムでは、どんな還元式も効果的にα回のサイクルだけで解くことができる。更に、積和形で表されたどんな一般ブール式も、効果的に２α回のサイクルで解くことができる。例えば、上記の還元式（２）は、以下の表に示した様に事前に１にセットされたレジスタに、比較の結果に従い条件付きでブール値の０を書き込む４つの演算を同時に実行することによって解くことができる。
【００１５】
【表２】

| 本発明による式（２）を解く例 | |
|---|---|
| サイクル | 命令 |
| | ｒ＝１；（前の演算と並列処理される） |
| 0 | ｒ＝ＡＮＤ－＜（ａ，ｂ）；<br>ｒ＝ＡＮＤ$_{c}$－＞（ｃ，ｄ）；<br>ｒ＝ＡＮＤ－＜（ａ，ｃ）；及び<br>ｒ＝ＡＮＤ－＞（ｂ，ｄ）． |
| α | ｒの値は使用可能である |

【００１６】
上記表では、レジスタｒは事前に１にセットされている。ＡＮＤで示された演算は、指定された比較演算（＜、又は＞）の結果が０である場合にレジスタｒにブール値の０を書き込む。ＡＮＤ$_{C}$で示された演算は、指定された比較演算の結果の補数が０である場合にレジスタｒにブール値の０を書き込む。従って、その演算の２つ以上が同時にレジスタに書き込むことができるが、書き込むことができる値は０だけである。
【００１７】
同時に実行される４つの演算は、効果的にα回のサイクルで還元式（２）を解く。（ここでは、例えば、機能ユニットの１つが別の状態、即ちアイドル状態となっている時のサイクルの間等で、レジスタを事前にセットする演算が、前の演算と並列処理されうると仮定している。従って、事前セットをする演算は効果的で、全く追加実行時間を必要としない）。ＡＮＤ演算によって実行された比較結果の何れか、又はＡＮＤ$_{C}$演算によって実行された比較結果の補数の何れかが０である場合、レジスタは０にセットされる。レジスタｒが本発明の第１の態様に従う時、複数の演算が同時にレジスタを０にセットすると、レジスタは定義値０にセットされる。そうでない場合、レジスタは１のままである。
【００１８】
加えて、積和形で表されるどんな一般ブール式も、２α回のサイクルで解くことができる。（この場合も、如何なる追加サイクルをも必要としない様に、事前設定の演算が前の演算と並列処理されると仮定している）。最初のα回のサイクルでは、式の各ＡＮＤ項が別々のＡＮＤ還元として実行される。次のα回のサイクルでは、ＡＮＤ還元の結果に対してＯＲ還元が実行されて式の値が得られる。
【００１９】
ブール式の値は特に述語化実行に有用である。命令の述語化実行とは、通常はブール値である入力値で条件付けされた実行を指す。例えば、述語化実行をサポートするマシン上では、第３の述語入力がある値を有するかどうかで、２つの入力値を加算する命令を条件付けることができる。２つの入力値の加算、又は少なくとも加算関数の結果の書込みは、述語入力が特定の値を有する時のみ実行される。従って、本発明の更なる特徴、及び利点によれば、本発明によって解かれた還元式で命令を述語化することができる。
【００２０】
本発明の他の特徴、及び利点は、添付の図面に関して進める以下の好適実施例の詳細な説明から明らかになろう。
【００２１】
【実施例】
図１を参照すると、本発明の好適実施例によるコンピュータ・システム２０は一般に、多重信号システム・バス３０によってメモリ２４、及び周辺装置２６に接続された中央演算処理装置（ＣＰＵ）２２を備えている。メモリ２４は、動的ランダム・アクセス・メモリ・チップで構成されていることが好ましいが、読取り専用メモリ、又はその他の電子メモリ装置で構成されていてもよい。メモリ２４は、コンピュータ・システム２０上で所望のタスク、又は計算を実施する１つ、又は複数のプログラム用のデータ、及び命令を記憶す

る。
【００２２】
図２を参照すると、コンピュータ・システム２０内のＣＰＵ２２は好適には命令レベル並列処理を提供する。（本発明の代替実施例では、コンピュータ・システムはパイプライン命令を提供するタイプのＣＰＵを有することができる。）より明確には、プロセッサ２２は、命令ユニット３４と、所望のタスク、又は計算を実行するためにプログラムの命令に応じて同時に演算を実行する複数の機能ユニット３６ないし３９を備えている。メモリ２４（図１）に記憶された命令は、命令ユニット３４に読み込まれる。バス・インタフェース回路４２は、システム・バス３０上のメモリ２４からデータ、及び命令を読み取り、かつメモリ２４にデータを書き込むために必要な信号を生成する。命令ユニット３４は、命令を復号して、その命令に従って演算を実行する様に１組の機能ユニット３６ないし３９に命令する制御信号を生成する。
【００２３】
命令ユニット３４は、所定の１組の命令（「命令セット」）に応答し、機能ユニット３６ないし３９を用いて対応する演算を実行するのに必要な制御信号を形成する様に構成されている。命令セットは一般に、論理演算命令、算術演算命令、及び浮動小数点演算命令を備えている。命令セットは又、メモリと１組のレジスタ４６の間、及びレジスタ同士の間でデータを転送するためのデータ転送命令も備えている。本発明によれば、命令ユニット３４が機能ユニットに実行する様命令できる演算の幾つかは、以下で更に詳細に説明する述語化プログラムの実行とブール還元を解くための演算である。
【００２４】
機能ユニット３６ないし３９は、命令ユニット３４によって指示された通りに命令を実施するための演算を実行する。機能ユニット３６ないし３９は一般に、演算で使用される１つ、又は２つのオペランドを、レジスタ４６の中から選択されたレジスタより読み取る。又、機能ユニット３６ないし３９は、１つ、又は複数の演算結果を、レジスタ４６の中から選択されたレジスタに書き込む。
【００２５】
好適実施例の命令ユニット３４、及び機能ユニット３６ないし３９は、従来の論理回路設計技法を使用した従来の論理ゲートによって実施される論理回路であり、以下で説明する演算を提供する。
【００２６】
次に、図３を参照すると、少なくともレジスタ４６（図２）の１つはブール値オペランドを記憶し、複数の同時書込みを扱うことができる。その様なレジスタ５４は、記憶装置、又は回路（記憶装置）５８、及び多数のポート６０ないし６３を備えることが好ましい（他のレジスタ４６は、整数値、又は浮動小数点値だけを記憶する等、他の目的専用にすることができる）。
【００２７】
一般に、記憶装置５８は、１つ、又は複数のラッチ回路、フリップ・フロップ回路、あるいは等価静的メモリ回路、又は等価動的メモリ回路を備えている。記憶装置５８は、単一ビットのみ記憶することもできるが、整数値や浮動小数点値等、複数ビット値を記憶することもできる。複数ビットを記憶する場合、任意のビットの組合せを各ブール状態に割り当てることによってブール値オペランドを表すことができる（例えば、全て０の場合はブール値の０を表し、他のあらゆるビットの組合せがブール値の１を表す）。複数ビット記憶を用いて、プログラム実行の必要性に応じて、ブール値オペランド、及び複数ビット整数値、又は複数ビット浮動小数点値を柔軟に記憶することができる。記憶装置５８は又、複数の単一ビットを備える複数ビット述語ベクトル、又は個別にあるいは集合的に書き込まれるブール述語値を記憶する様に実施することもできる。
【００２８】
ポート６０ないし６３は、機構ユニット３６ないし３９からレジスタ５４に書き込まれる値を受け取る。ポート６０ないし６３は一般に、データ・バス６６ないし６９と書込み可

能線７０ないし７３の両方がそれぞれ機能ユニット３６ないし３９に１対１の関係で接続されている。機能ユニット３６ないし３９は、演算の結果等の値をレジスタ５４に書き込むために、機能ユニット自身のデータ・バス６６ないし６９に書き込まれている値を利用し、機能ユニット自身の書込み可能線７０ないし７３上に書込み可能信号を設定する。ポート６０ないし６３は、書込み可能信号に応答して、ポート自身のそれぞれのデータ・バス６６ないし６９上の値を記憶装置５８にゲートする。例えば、機能ユニット３７がポート６１の書込み可能信号を設定すると、そのポート用のデータ・バス６７上の値が記憶装置５８に書き込まれる。ポート６１用の書込み可能信号が設定されない場合、値は記憶装置５８に書き込まれない。（述語ベクトルの記憶を実施する場合、ベクトル内の各述語の値に対して別々の可能線、及び可能信号が使用されうる）。

【００２９】

機能ユニット３６ないし３９のうち２つ以上が、演算が同時に実行されている間に、機能ユニット自身の書込み可能信号を設定すると、複数の値が同時にレジスタ５４に書き込まれる。従来型のレジスタでは、その様な複数の同時書込みは可能でなく、かつその様な書込みによって記憶装置５８内に記憶された値が不定になることがある。しかし、レジスタ５４はそれ自身に同時に書き込まれる全ての値が同じである場合、定義された値を記憶装置５８に記憶する様に動作する。その場合、その記憶される値は、書き込まれる値と同じであることが好ましい。

【００３０】

複数の値が同時に書き込まれた時に記憶されるべき値は、レジスタ５４内でポート６０ないし６３と記憶装置５８の間に接続された選択回路７６によって導出される。選択回路７６は、記憶される値を多数の方法で導出することができる。例えば、選択回路は、書き込まれた複数の値のＡＮＤ関数、又はＯＲ関数を実行でき、あるいは複数の値のうち１つを任意に選択して、記憶される値を導出することができる。同時に書き込まれた複数の値が異なるものである場合、結果は不定となる。従って、異なる値が同時に書き込まれた場合、選択回路７６によって記憶される値は重要なものではない。

【００３１】

再び図２を参照すると、機構ユニット３６ないし３９は、命令ユニット３４の制御下で、プログラムの命令に対応する一定の演算を実行する様に動作する。命令ユニットの制御下で機能ユニット３６ないし３９が実行する演算には、述語化プログラムの実行、及びブール還元式を解くのに有用な１組の還元演算が含まれる。還元演算はそれぞれ以下の形をとる：

$$r_{out} = operation(r_{in\ 1}, r_{in\ 2}, \ldots, r_{in\ n}) \quad (4)$$

言い換えると、各還元演算は、１つ、又は複数の入力（$r_{in\ 1}, r_{in\ 2}, \ldots, r_{in\ n}$）に作用し、指定されたレジスタ内に記憶される結果の新しい値（$r_{out}$）を決定する。より明確には、各演算の実行は、結果関数（$F_{out}$）と許可関数（$F_{en}$）の２つの関数によって厳密に定義される。ここで、結果（$r_{out}$）は以下の通りである：

$$r_{out} = F_{out}(r_{in\ 1}, r_{in\ 2}, \ldots, r_{in\ n}) \quad if \quad F_{en}(r_{in\ 1}, r_{in\ 2}, \ldots, r_{in\ n}) \quad (5)$$

結果関数（$F_{out}$）は、結果が記憶される場合、指定されたレジスタ内に記憶される結果の新しい値を決定する。許可関数（$F_{en}$）は、指定された出力レジスタ内に結果を記憶するかどうかを決定するブール値を算出する。

【００３２】

一実施例では、結果関数はブール値を有する。ブール値結果関数、及びブール値許可関数は各還元演算に固有のものである。各還元演算は、機能ユニット内で（１）１組の入力に対して演算の特定の結果関数を実行して結果のブール値を生成し、（２）指定された目標レジスタにブール値を書き込むかどうかを決定する演算の許可関数を実行することによって実行される。結果関数、又は許可関数に固有なブール値関数のタイプの例には、一般に、整数入力値に対する比較関数（例えば、より大きい、より小さい、等しい、それらの補

数等）、浮動小数点入力値に対する比較関数、及びブール入力値の関数（例えば、ＡＮＤ、ＯＲ、ＸＯＲ、一致、それらの補数等）が含まれる。

【００３３】

命令ユニット３４が機能ユニット３６ないし３９のうちの１つに実行する様命令する各還元演算ごとに、命令ユニット３４は、演算の特定のブール値結果関数用の入力値（又は入力値が取られてくるレジスタ）と、演算によって書き込まれるあらゆる値を記憶するための目標レジスタを指定する。指定された入力値は、演算の特定の結果関数に適当なものでなければならない。指定された目標レジスタは、ブール値を記憶できなければならず、好適実施例では、図３に示しかつ上記で説明した、複数の同時書込みを扱うタイプのもの（レジスタ５４）である。

【００３４】

許可関数は、結果関数によって生成される結果値を目標レジスタに書き込むかどうかを決定する。従って、各還元演算は、結果関数、及び許可関数の結果の組合せに応じて３つの動作をとりうる。これらの動作には、ブール値１を目標レジスタに書き込むこと、ブール値０を目標レジスタに書き込むこと、及び目標レジスタに書き込まないことが含まれる。

【００３５】

以下の表３は、本発明の好適実施例における還元演算の４つのクラスについての真理値表を示す。

【００３６】

【表３】

ＡＮＤ還元演算、及びＯＲ還元演算についての真理値表

| 条件 | 条件に対する演算 | | 条件の補数に対する演算 | |
|---|---|---|---|---|
| | ＯＲ | ＡＮＤ | ＯＲc | ＡＮＤc |
| 0 | -- | 0 | 1 | -- |
| 1 | 1 | -- | -- | 0 |

【００３７】

表３では、最も左の列は、ブール値比較関数の結果として生成される「条件」値の状態を表す。他の列は、条件の各状態ごとに還元演算の４つのクラス、ＯＲ、ＡＮＤ、$ＯＲ_C$、及び$ＡＮＤ_C$によって実行される動作を表す。ここで、「１」はブール値１を目標レジスタに書き込むことを、「０」はブール値０を目標レジスタに書き込むことを、「－－」は目標レジスタ内の値を変更しないことを表す。

【００３８】

図３を参照すると、機能ユニット３６ないし３９は、以下の表４の値の列に示した値を機能ユニット自身の各データ・バス６６ないし６９上に置き、書込み可能信号の列に示した様に機能ユニット自身の各書込み可能線を設定することによって、上述のＡＮＤ還元演算、及びＯＲ還元演算を実行する。

【００３９】

【表４】

ＡＮＤ還元演算、及びＯＲ還元演算の実行

| 還元演算 | 結果値 | 書込み可能信号 |
|---|---|---|
| ＡＮＤ | 0 | ¬条件 |
| ＡＮＤc | 0 | 条件 |
| ＯＲ | 1 | 条件 |
| ＯＲc | 1 | ¬条件 |

【００４０】
上記の表４は、表３の演算の４つのクラスのそれぞれの結果関数、及び許可関数によって生成される値を示す。これらの演算に関する結果関数は定数である。「値」の列に示した様に、ＡＮＤ演算クラス、及びＡＮＤ$_C$演算クラスに関する結果関数は常に結果として０を生成し、ＯＲ演算クラス、及びＯＲ$_C$演算クラスに関する結果関数は常に結果として１を生成する。「書込み可能信号」の列は、これらの演算クラスに関する許可関数が、条件値、又はその補数を生成する比較関数であることを示す。より明確には、ＯＲ還元演算クラス、及びＡＮＤ$_C$還元演算クラスに関する許可関数は、指定されたブール値比較関数（条件を生成する比較関数）である。ＯＲ$_C$還元演算クラス、及びＡＮＤ還元演算クラスの許可関数は、指定されたブール値比較関数の補数である。

【００４１】
前記の表３は、本発明の好適実施例で提供される還元演算の４つのクラス、明確にはＡＮＤ、ＡＮＤ$_C$、ＯＲ、及びＯＲ$_C$の還元演算クラスである。ＡＮＤ、及びＡＮＤ$_C$クラスでの還元演算は、ＡＮＤ還元式を解くうえで有用である。ＯＲ、及びＯＲ$_C$クラスでの還元演算は、ＯＲ還元式を解くうえで有用である。例えば、以下に再び示すＡＮＤ還元式（２）：

ｒ＝（ａ＜ｂ）∧¬（ｃ＞ｄ）∧（ａ＜ｃ）∧（ｂ＞ｄ）（２）

は、以下の４つのＡＮＤ、及びＡＮＤ$_C$クラスの還元演算によって解くことができる：

ｒ＝ＡＮＤ－＜（ａ，ｂ）（６）；
ｒ＝ＡＮＤ$_C$－＞（ｃ，ｄ）（７）；
ｒ＝ＡＮＤ－＜（ａ，ｃ）（８）；
ｒ＝ＡＮＤ－＞（ｂ，ｄ）（９）。

【００４２】
上記の還元演算（６ないし９）はそれぞれ、（括弧内の）指定された２つの入力値に対する演算固有の比較関数（＜、又は＞）と、指定されたレジスタ（ｒ）を目標とする演算固有の還元関数（ＡＮＤ、又はＡＮＤ$_C$）から成る。各還元演算の比較関数はブール値条件を生成する。次に、上記表３、及び４に示した様に、各還元演算の許可関数は、条件に基づき、ブール値０が目標レジスタに書き込まれるかどうかを決定する。還元演算（６）、及び（８）は同じ還元演算であるが、異なる入力値を指定することに留意されたい。

【００４３】
還元演算（６ないし９）によってＡＮＤ還元式（２）を解くために、まず事前にレジスタ（ｒ）をブール値１にセットする。次いで、還元演算（６ないし９）がそれぞれ実行される。必ずしも必要ではないが、機能ユニット３６ないし３９によって還元演算を並列に実行することが好ましい。どれかのＡＮＤ還元演算の比較関数が０を生成する場合、目標レジスタに０が書き込まれる。又ＡＮＤ$_C$の比較演算部分が１を生成する（比較の補数が０である）場合にも、目標レジスタに０が書き込まれる。従って、全ての還元演算（６）ないし（９）が実行された後、（効果的に式（２）を解いて）目標レジスタは、ＡＮＤ還元式（２）の全ての項が真である場合は１となり、そうでない場合は０となる。上記の表２に示した様に、還元演算（６ないし９）を並列に実行すると、式（２）は、効果的にコンピュータ・システム２０のα回のサイクルだけで解かれる。

【００４４】
本発明の代替実施例では、還元演算は、ブール値述語入力に基づいて目標レジスタに書き込まれる値を決定すると共に、比較関数によって生成される条件値も決定する。この代替実施例では、命令ユニット３４は、各還元演算ごとに述語値（又は述語値を含むレジスタ）を、演算の入力値、及び目標レジスタと共に指定する。以下の表５は、この代替実施例におけるいくつかの典型的な還元演算のクラスを示している。

【００４５】
【表５】

| 述語入力による還元演算の真理値表 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 述語入力 | 条件 | 条件に対する演算 | | | | 条件の補数に対する演算 | | | |
| | | U | C | OR | AND | $U_C$ | $C_C$ | $OR_C$ | $AND_C$ |
| 0 | 0 | 0 | − | − | − | 0 | − | − | − |
| 0 | 1 | 0 | − | − | − | 0 | − | − | − |
| 1 | 0 | 0 | 0 | − | 0 | 1 | 1 | 1 | − |
| 1 | 1 | 1 | 1 | 1 | − | 0 | 0 | − | 0 |

【００４６】
表５は、条件なし（Ｕ）、条件付き（Ｃ）、ＯＲ、及びＡＮＤ、ならびにそれらの補数（$U_C$、$C_C$、$OR_C$、$AND_C$）の還元演算の８つのクラスを示す。条件なしクラス（Ｕ、$U_C$）は、述語入力とその条件（又はその条件の補数）のブールＡＮＤ関数を目標レジスタに書き込む。条件付きクラスは、述語入力に応じて目標レジスタに条件を書き込む。最後に、ＯＲ、及びＡＮＤクラスが、表３に示した様に述語入力に応じて処理を行う。この場合も、表３に示した還元演算と同様に、これらの還元演算のそれぞれの実行は、結果値を生成する結果関数と、指定された目標レジスタに結果値が書き込まれるかどうかを決定する許可関数によって厳密に定義される。

【００４７】
表５に示したタイプの還元演算では、還元式を解く１組の還元演算の性能を、指定されたレジスタに含まれる述語入力の値で条件付けることができる。例えば、目標レジスタと、２つ以上のＡＮＤクラス演算を事前にセットするための条件付きクラス演算を含む１組の還元演算は、指定された述語入力が１である場合にＡＮＤ還元式（例えば、式（２））を解く様に準備され、述語入力が０である場合は何もしない。

【００４８】
１組の還元演算の実行から結果的に得られるブール値は、演算が実行された後に演算の目標レジスタに記憶され、プログラム内の演算の述語化実行に特に有用である。本発明の好適実施例では、演算は、ブール述語値が記憶されるレジスタを指定することができる。その場合、演算の実行は述語値で条件付けられる。好適実施例では、その演算が実行されるのは、それらの演算の指定された述語値が１の時だけである。

【００４９】
他の代替実施例では、同じ演算において所与の１組の入力値からそれぞれ複数の結果を生成する１組の還元演算が提供される。そのため、複数の結果を伴う還元演算は以下の形をとる：

$$r_{out,1}, r_{out,2}, \ldots, r_{out,m} = op(r_{in,1}, r_{in,2}, \ldots, r_{in,n}) \quad (10)$$

ここで、１組の入力値（$r_{in,1}, r_{in,2}, \ldots, r_{in,n}$）に対して演算を実行することによって複数の結果値（$r_{out,1}, r_{out,2}, \ldots, r_{out,m}$）が生成される。これらの還元演算の実行は、複数の結果値のそれぞれごとに２つの関数（$F_{out,i}$，$F_{en,i}$）によって厳密に定義される。より明確には、各ｉごとに、結果（$r_{out,i}$）は以下の値をとる：

$$r_{out,i} = F_{out,i}(r_{in,1}, r_{in,2}, \ldots, r_{in,n}) \quad if \ F_{en,i}(r_{in,1}, r_{in,2}, \ldots, r_{in,n}) \quad (11)。$$

【００５０】
複数の結果を生成するタイプの還元演算は、同じ１組の条件を使用する複数の還元式を解く状況で特に有用である。例えば、状況によっては、４つ１組の比較のＡＮＤ還元式と、同じ１組の比較のＯＲ還元式を解くことが必要になることがある。この様な他の代替実施例では、４つの機能ユニットだけで両方の還元式を同時に解くことができる。その４つの機能ユニットは、それぞれ、２つの別の目標レジスタに書き込むためにＡＮＤ還元関数と

ＯＲ還元関数を有する、４つの還元演算をそれぞれ実行する。この技法は、演算を述語化実行するために使用される２つの結果述語を条件分岐の真、及び偽の側から算出するうえで特に有用である。

【００５１】

好適実施例に関して本発明の原理を説明し図示してきたが、本発明は、そうした原理から逸脱することなく構成、及び細部を修正できることが認識されよう。例えば、図示された実施例は命令レベル並列コンピュータ・システムであるが、本発明は多重処理コンピュータ・システム（１つのプロセッサ内に別の機能ユニットを有するのではなく、演算を実行するための別の複数プロセッサを有する）に適用することもできる。その様な多重処理コンピュータ・システムの実施例においては、還元式は、隣接する同期点の間に１組の還元演算をそれぞれ別々のプロセッサ上で実行することによって解かれる。還元演算は、複数の同時書込みを扱う記憶レジスタ内に式の値を形成する。

【００５２】

他の例として、還元式に関連するブール値比較関数を、式を解く還元演算とは別に実行することができる。より明確には、演算の第１の組でブール値比較関数を実行して、所望の還元式に関する１組の条件を形成することができる。次に、還元演算の第２の組を実行して、条件を式の最終値に還元する。還元関数が２つの入力を持つＡＮＤ演算の２分木としてでなく、同時に実行されるので、この例でも従来の高さ還元技法を越える利点を提供することに留意されたい。

【００５３】

更に他の例として、本発明による還元演算は、それぞれが、複数のブール値を備えるベクトルに作用することができる。その様なベクトルは、入力値、述語値、及び／又は還元演算の関数の結果として機能することができる。

【００５４】

本発明の原理を適用可能な多数の実施例に関して言うと、ここで詳述された実施例は説明のみを目的としたものであり、本発明の範囲を制限するものではないということを認識されたい。むしろ、特許請求の範囲、及び意図に含まれる全ての実施例、及びその同等物は、本発明の範囲内であると言うことができる。

【００５５】

以下に本発明の実施態様を列挙する。

【００５６】

１．　コンピュータ・システムにおいてブール式、及び述語を解く方法であって、
命令セットの命令に対応する演算が、潜在的にパイプライン化された１つ、又は複数の機能ユニットによって実行されるコンピュータ・システムを提供するステップと、
機能ユニット内で演算を実行するステップを含み、
前記演算が１組の１つ、又は複数の入力値から、指定されたレジスタに記憶される値を決定する形の１つ、又は複数の還元演算を含み、前記機能ユニット内の還元演算の実行が、
機能ユニット内で１つ、又は複数の演算入力値の結果関数を実行し、結果値を生成するステップと、
機能ユニット内で１つ、又は複数の演算入力値の許可関数を実行し、ブール許可値を生成するステップと、
ブール許可値が真である場合に、指定されたレジスタ位置に結果値を書き込むステップを含むことを特徴とする方法。

【００５７】

２．　前記方法が更に、１組の還元演算を実行してブール式を解くステップを含み、１組の還元演算を実行する前記ステップが、
第１の還元演算を実行して、指定されたレジスタ位置に第１のブール値を書き込むステップと、
演算の許可関数によって生成されるブール許可値が真である場合に、１組の２つ以上の還元演算を実行し、そのそれぞれが指定されたレジスタ位置に第２のブール値をそれぞれ書

き込み、それによって、指定されたレジスタ位置に記憶される値をブール式の値とするステップを含むことを特徴とする、項番１に記載の方法。

【００５８】

３．　指定されたレジスタに記憶される値で条件付けられた演算を続けて実行するステップを更に含むことを特徴とする項番２に記載の方法。

【００５９】

４．　機能ユニット内の還元演算の少なくとも１つが、機能ユニット内で２つ以上の入力値の許可関数を実行し、入力値の少なくとも１つの組合せに関して偽である許可関数を実行することによって生成されるブール許可値を生成するステップを含むことを特徴とする項番１に記載の方法。

【００６０】

５．　１組の還元演算を実行してブール式を解くステップを更に含み、１組の還元演算を実行する前記ステップが、

第１の還元演算を実行して、指定されたレジスタ位置に第１のブール値を書き込むステップと、

演算の許可関数によって生成されるブール許可値が真である場合に、１組の２つ以上の還元演算を実行し、そのそれぞれが指定されたレジスタ位置に第２のブール値をそれぞれ書き込み、それによって、指定されたレジスタ位置に記憶された値をブール式の値とするステップを含むことを特徴とする、項番４に記載の方法。

【００６１】

６．　指定されたレジスタに記憶された値で条件付けられた演算を続けて実行するステップを更に含むことを特徴とする項番５に記載の方法。

【００６２】

７．　還元演算の少なくとも１つを機能ユニットで実行するステップが、機能ユニット内で少なくとも１つの入力値を共通に有する結果関数と許可関数を実行し、かつ入力値の少なくとも１つの組合せに関して偽である許可関数を実行することによって生成されるブール許可値を生成するステップを含むことを特徴とする項番１に記載の方法。

【００６３】

８．　１組の還元演算を実行してブール式を解くことを更に含み、１組の還元演算を実行する前記ステップが、

第１の還元演算を実行して、指定されたレジスタ位置に第１のブール値を書き込むステップと、

演算の許可関数によって生成されるブール許可値が真である場合に、１組の２つ以上の還元演算を実行し、そのそれぞれが指定されたレジスタ位置に第２のブール値を書き込み、それによって、指定されたレジスタ位置に記憶される値をブール式の値とするステップを含むことを特徴とする、項番７に記載の方法。

【００６４】

９．　指定されたレジスタに記憶された値で条件付けられた演算を続けて実行するステップを更に含むことを特徴とする項番８に記載の方法。

【００６５】

１０．　複数出力の還元演算を機能ユニット内で実行するステップを更に含み、それぞれの複数出力の還元演算は１つ、又は複数の入力値の組から、１つ、又は複数の指定されたレジスタ内の複数の値を決定する形を有し、機能ユニット内のそれぞれの複数出力の還元演算の実行が、演算によって決定される複数の値のそれぞれごとに、

機能ユニット内で演算の１つ、又は複数の入力値の結果関数を実行して、結果値を生成するステップと、

機能ユニット内で演算の１つ、又は複数の入力値の許可関数を実行して、ブール許可値を生成するステップと、

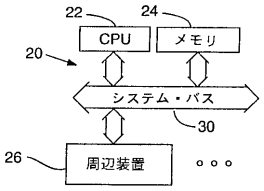
ブール許可値が真である場合に、指定されたレジスタ位置に結果値を書き込むステップを含むことを特徴とする項番１に記載の方法。

【００６６】
１１．　１組の還元演算を実行してブール式を解くステップを更に含み、前記組中の少なくとも１つの還元演算が複数出力の還元演算であり、その組の還元演算の実行が、
第１の還元演算を実行して、指定されたレジスタ位置に第１のブール値を書き込むことと、
演算の許可関数によって生成されるブール許可値が真である場合に、１組の２つ以上の還元演算を実行し、そのそれぞれが指定されたレジスタ位置に第２のブール許可値を書き込み、それによって、指定されたレジスタ位置に記憶される値をブール式の値とするステップを含むことを特徴とする、項番１０に記載の方法。
【００６７】
１２．　指定されたレジスタに記憶される値で条件付けられた演算を続けて実行するステップを更に含むことを特徴とする項番１１に記載の方法。
【００６８】
１３．　複数還元演算のうち少なくとも１つを機能ユニット内で実行するステップが、演算によって決定される少なくとも１つの値について、機能ユニット内で２つ以上の入力値の許可関数を実行し、入力値の少なくとも１つの組合せに関して偽である許可関数を実行することによって生成されるブール許可値を生成するステップを含むことを特徴とする項番１０に記載の方法。
【００６９】
１４．　１組の還元演算を実行してブール式を解くステップを更に含み、前記組中の少なくとも１つの還元演算が複数出力の還元演算であり、その組の還元演算を実行するステップが、
第１の還元演算を実行して、指定されたレジスタ位置に第１のブール値を書き込むステップと、
演算の許可関数によって生成されるブール許可値が真である場合に、１組の２つ以上の還元演算を実行し、そのそれぞれが指定されたレジスタ位置に第２のブール値を書き込み、それによって、指定されたレジスタ位置に記憶される値をブール式の値とするステップと、
指定されたレジスタに記憶された値で条件付けられた演算を続けて実行するステップを含むことを特徴とする、項番１３に記載の方法。
【００７０】
１５．　複数出力の還元演算の少なくとも１つを機能ユニットで実行するステップが、演算によって生成される少なくとも１つの値について、機能ユニット内で少なくとも１つの入力値を共通に有する結果関数と許可関数を実行し、かつ入力値の少なくとも１つの組合せに関して偽である許可関数を実行することによって生成されるブール許可値を生成するステップを含むことを特徴とする項番１０に記載の方法。
【００７１】
１６．　１組の還元演算を実行してブール式を解くステップを更に含み、前記組中の少なくとも１つの還元演算が複数出力の還元演算であり、その組の還元演算を実行するステップが、
第１の還元演算を実行して、指定されたレジスタ位置に第１のブール値を書き込むステップと、
演算の許可関数によって生成されるブール許可値が真である場合に、１組の２つ以上の還元演算を実行し、そのそれぞれが指定されたレジスタ位置に第２のブール値を書き込み、それによって、指定されたレジスタ位置に記憶される値をブール式の値とするステップと、
指定されたレジスタに記憶された値で条件付けられた演算を続けて実行するステップを含むことを特徴とする、項番１５に記載の方法。
【００７２】
１７．　コンピュータ・システムにおいて、

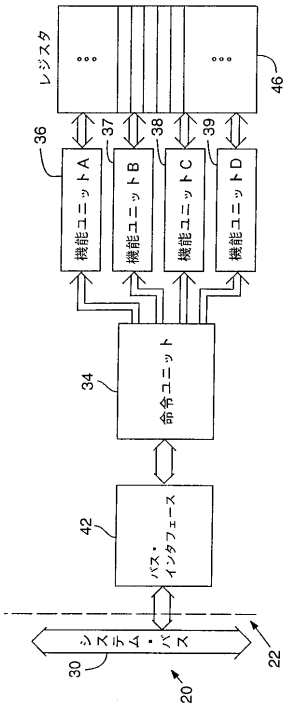
値を記憶するための複数のレジスタと、
レジスタに結合された１つ、又は複数の機能ユニットであって、各前記機能ユニットがコンピュータ・システム用の命令セットの命令に対応した複数の演算を実行し、前記演算がレジスタ内の指定された記憶位置に記憶された値を、１つ、又は複数の入力値から決定するための還元演算から成る機能ユニットと、
演算の１つ、又は複数の入力値の結果関数を実行することにより、結果値を生成し、演算の１つ、又は複数の入力値の許可関数を実行してブール許可値を生成し、ブール許可値が真である場合に、指定されたレジスタ位置に結果値を書き込むことによって還元演算を実行する機能ユニットを備えることを特徴とするコンピュータ・システム。
【００７３】
１８．　複数のレジスタが、レジスタに書き込まれた各ブール値が同じものである時、レジスタに同時に書き込まれた１つ、又は複数のブール値に基づいて値を記憶する様に動作する、同時に書き込まれた複数の値を受け入れるためのレジスタを備えることを特徴とする、項番１７に記載のコンピュータ・システム。
【００７４】
１９．　複数の入力ポートを有する、同時に書き込まれる値を受け入れるためのレジスタであって、前記入力ポートが値と書込み可能信号を機能ユニットから受け取るために機能ユニットに１対１の関係で接続されており、機能ユニットの書込み可能信号が設定された時に、機能ユニットから受け取った前記値が書き込まれるレジスタと、
レジスタ内のレジスタ位置を指定する還元演算を実行する時に、許可関数を実行することによって生成されたブール許可値が真である場合に、演算の結果関数を実行することによって生成された結果値をレジスタに送り、書込み可能信号を設定する機能ユニットと、
機能ユニットによって複数の同じ値が同時にレジスタに書き込まれた時に、レジスタに記憶する値を導出するための手段を更に備えることを特徴とする項番１８に記載のコンピュータ・システム。
【００７５】
２０．　機能ユニットが、演算によって決定された複数の値のそれぞれについて、演算の１つ、又は複数の入力値の結果関数を実行して結果値を生成し、演算の１つ、又は複数の入力値の許可関数を実行してブール許可値を生成し、ブール許可値が真である場合に、指定されたレジスタ位置に結果値を書き込むことによって、レジスタに記憶される複数の値を１つ、又は複数の入力値から決定するための還元演算を実行する様動作することを特徴とする項番１８に記載のコンピュータ・システム。
【００７６】
【発明の効果】
本発明に従う上述の２つの態様のコンピュータ・システムによって、どんな還元式も効果的にα回のサイクルだけで解くことができ、更に、積和形で表されたどんな一般ブール式も、効果的に２α回のサイクルで解くことが可能となる。
【図面の簡単な説明】
【図１】本発明の好適実施例によるコンピュータ・システムの概略ブロック図である。
【図２】命令レベル並列処理のための複数の機能ユニットを有する図１のコンピュータ・システム内の中央演算処理装置の概略ブロック図である。
【図３】図２の中央演算処理装置内のレジスタの概略ブロック図である。
【符号の説明】
２０　　コンピュータ・システム
２２　ＣＰＵ
２４　メモリ
３０　システム・バス
３４　命令ユニット
３６－３９　機能ユニット
４６、５４　レジスタ

６０－６３　ポート
６６－６９　　データ・バス
７０－７３　　書込み可能線
７６　　選択回路

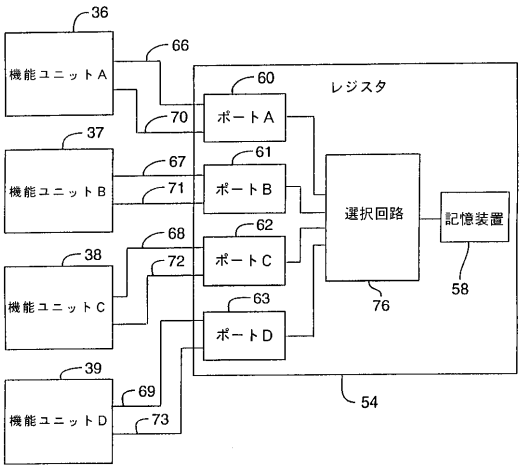
【図１】




【図２】



【図３】


36
機能ユニットA
66
60
レジスタ
ポートA
70
37
機能ユニットB
67
61
ポートB
71
選択回路
記憶装置
68
62
ポートC
38
72
機能ユニットC
63
ポートD
76
58
39
69
機能ユニットD
73
54

フロントページの続き

(72)発明者　ビー・ラマクリシュナ・ラウ
アメリカ合衆国カリフォルニア州９４０２４ロス・アルトス，ハイランズ・サークル・９００
(72)発明者　ヴィノッド・カサイル
アメリカ合衆国カリフォルニア州９５０１４クパーティノ，ナンバー・シー５，ノース・フォーシル・ブールヴァード・１０２７０

審査官　後藤　彰

(56)参考文献　特開平４－１８４５３５（ＪＰ，Ａ）
特開昭５８－４８１５３（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F　9/30　-　9/38
G06F　7/60
G06F 17/10　-　17/18