(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2014-197169
(P2014-197169A)
(43)公開日　平成26年10月16日(2014.10.16)

(51)Int.Cl.		F I			テーマコード（参考）
G09C 1/00 (2006.01)		G09C	1/00	650Z	5J104

審査請求 未請求　請求項の数 19　ＯＬ　（全 27 頁）

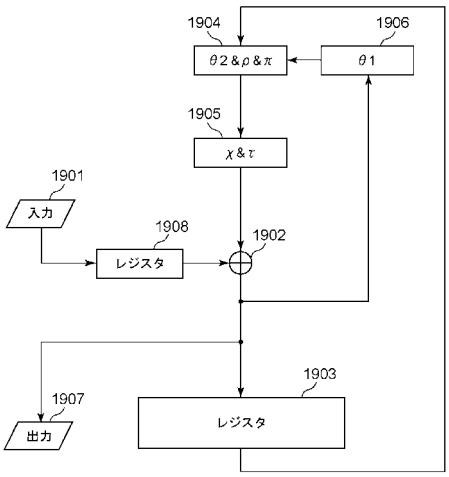
(21)出願番号　特願2014-17413(P2014-17413)
(22)出願日　平成26年1月31日(2014.1.31)
(31)優先権主張番号　特願2013-45574(P2013-45574)
(32)優先日　平成25年3月7日(2013.3.7)
(33)優先権主張国　日本国(JP)

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100126240
弁理士　阿部　琢磨
(74)代理人　100124442
弁理士　黒岩　創吾
(72)発明者　唐木　靖雅
東京都大田区下丸子３丁目３０番２号キヤノン株式会社内
(72)発明者　山田　真也
東京都大田区下丸子３丁目３０番２号キヤノン株式会社内

Ｆターム(参考)　5J104 AA12 AA18 AA32 AA41 EA08 JA01 NA12 NA39

(54)【発明の名称】ハッシュ値生成装置

(57)【要約】
【課題】　ハッシュ値生成のスループット向上を可能とする技術を提供する。
【解決手段】　ＳＨＡ－３アルゴリズムのラウンド処理に含まれるθ処理を実行するθ処理手段と、　前記ラウンド処理に含まれるρ処理を実行するρ処理手段と、前記ラウンド処理に含まれるπ処理を実行するπ処理手段と、前記ラウンド処理に含まれるχ処理を実行するχ処理手段と、前記ラウンド処理に含まれるι処理を実行するι処理手段を有し、前記π処理手段は、ｐｌａｎｅ単位でデータ入力し、ｓｈｅｅｔ単位でデータ出力する。
【選択図】　図１６



【特許請求の範囲】
【請求項１】
ＳＨＡ－３アルゴリズムのラウンド処理に含まれるθ処理を実行するθ処理手段と、
前記ラウンド処理に含まれるρ処理を実行するρ処理手段と、
前記ラウンド処理に含まれるπ処理を実行するπ処理手段と、
前記ラウンド処理に含まれるχ処理を実行するχ処理手段と、
前記ラウンド処理に含まれるι処理を実行するι処理手段を有し、
前記π処理手段は、ｐｌａｎｅ単位でデータ入力し、ｓｈｅｅｔ単位でデータ出力することを特徴とするハッシュ値生成装置。
【請求項２】
処理されたデータを保持する保持手段を有し、
前記保持手段は、ｓｈｅｅｔ単位でデータ入力、ｐｌａｎｅ単位でデータ出力を行うことを特徴とする請求項１に記載のハッシュ値生成装置。
【請求項３】
前記保持手段は、５つのｓｈｅｅｔが保持されたら、１つのｐｌａｎｅを出力することを特徴とする請求項２に記載のハッシュ値生成装置。
【請求項４】
前記θ処理手段は、ｃｏｌｕｍｎ和算出処理を行うθ１処理手段と、ｃｏｌｕｍｎ和加算処理を行うθ２処理手段とを有することを特徴とする請求項３に記載のハッシュ値生成装置。
【請求項５】
前記保持手段に、５つのｓｈｅｅｔが入力されている間に、前記θ１処理手段が実行されることを特徴とする請求項４に記載のハッシュ値生成装置。
【請求項６】
前記保持手段は、前記π処理手段で処理されたデータを保持することを特徴とする請求項２に記載のハッシュ値生成装置。
【請求項７】
前記ラウンド処理において、前記π処理手段は、前記θ２処理手段と前記ρ処理手段より前に、実行されることを特徴とする請求項４に記載のハッシュ値生成装置。
【請求項８】
前記θ２処理手段、前記ρ処理手段、前記χ処理手段、前記ι処理手段は、ｐｌａｎｅ単位で処理されることを特徴とする請求項７に記載のハッシュ値生成装置。
【請求項９】
前記保持手段は、前記ι処理手段で処理されたデータを保持することを特徴とする請求項２に記載のハッシュ値生成装置。
【請求項１０】
前記θ２処理手段、前記ρ処理手段は、ｐｌａｎｅ単位で処理され、
前記χ処理手段、前記ι処理手段は、ｓｈｅｅｔ単位で処理することを特徴とする請求項４に記載のハッシュ値生成装置。
【請求項１１】
前記θ手段、前記ρ手段、前記π手段、前記χ手段、前記ι手段を用いて、前記ラウンド処理を実行して得られたハッシュ値を出力する出力手段を有することを特徴とする請求項１乃至１０の何れか１項に記載のハッシュ値生成装置。
【請求項１２】
前記ｐｌａｎｅは、ｘ軸方向にｍビット、ｙ軸方向に１ビット、ｚ軸方向にｓビットの構造として表されるデータであり、前記ｓｈｅｅｔは、ｘ軸方向に１ビット、ｙ軸方向にｎビット、ｚ軸方向にｓビットの構造として表されるデータであることを特徴とする請求項１乃至１１の何れか１項に記載のハッシュ値生成装置。
【請求項１３】
前記θ処理手段は、ｘ軸方向のビットの和を算出し、算出した和を所定のビットに加算

し、
前記ρ処理手段は、各ビットの値をｚ軸方向にシフトし、
前記π処理手段は、ｘ－ｙ平面内で各ビットの値を入れ替え、
前記χ処理手段は、ｘ軸方向のビット列内での変換を行い、
前記ι処理手段は、各ビットに所定の値を加えることを特徴とする請求項１乃至１２の何れか１項に記載のハッシュ値生成装置。
【請求項１４】
ｘ軸方向にｍビット、ｙ軸方向にｎビット、ｚ軸方向にｓビットの構造を持つデータに対して処理を行うハッシュアルゴリズムのラウンド処理において、
ｘ軸方向のビットの和を算出し、算出した和を所定のビットに加算する第１の処理手段と、
前記ｚ軸方向にシフトする第２の処理手段と、
ｘ－ｙ平面内で各ビットの値の入替を行う第３の処理手段と、
ｘ軸方向のビット列内での変換を行う第４の処理手段と、
各ビットに所定の値を加算する第５の処理手段を有し、
前記第３の処理手段は、ｘ軸方向にｍビット、ｙ軸方向に１ビット、ｚ軸方向にｓビットの構造として表される単位でデータが入力され、ｘ軸方向に１ビット、ｙ軸方向にｎビット、ｚ軸方向にｓビットの構造として表される単位でデータが出力されることを特徴とするハッシュ値生成装置。
【請求項１５】
処理されたデータを保持する保持手段を有し、
前記保持手段は、ｘ軸方向に１ビット、ｙ軸方向にｎビット、ｚ軸方向にｓビットの構造として表される単位でデータ入力、ｘ軸方向にｍビット、ｙ軸方向に１ビット、ｚ軸方向にｓビットの構造として表される単位でデータ出力を行うことを特徴とする請求項１４に記載のハッシュ値生成装置。
【請求項１６】
前記保持手段は、ｘ軸方向に１ビット、ｙ軸方向にｎビット、ｚ軸方向にｓビットの構造として表される単位のデータが５つ保持された後に、ｘ軸方向にｍビット、ｙ軸方向に１ビット、ｚ軸方向にｓビットの構造として表される単位のデータを１つ出力することを特徴とする請求項１５に記載のハッシュ値生成装置。
【請求項１７】
前記第１の処理手段は、ｘ軸方向のビットの和を算出する第６の処理手段と、算出した和を所定のビットに加算する第７の処理手段とを有することを特徴とする請求項１６に記載のハッシュ値生成装置。
【請求項１８】
前記保持手段に、ｘ軸方向に１ビット、ｙ軸方向にｎビット、ｚ軸方向にｓビットの構造として表される単位のデータが５つ入力されている間に、前記第６の処理手段が実行されることを特徴とする請求項１７に記載のハッシュ値生成装置。
【請求項１９】
前記θ手段、前記ρ手段、前記π手段、前記χ手段、前記ι手段を用いて、前記ラウンド処理を実行して得られたハッシュ値を出力する出力手段を有することを特徴とする請求項１４乃至１８の何れか１項に記載のハッシュ値生成装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ハッシュ値を生成する技術に関するものである。
【背景技術】
【０００２】
データの改ざんがないかを検証するために、暗号学的ハッシュアルゴリズムを用いて算出されるハッシュ値が利用されている。暗号学的ハッシュアルゴリズムであるＳＨＡ－１

は安全性が確保できないことが既に証明されており、ＳＨＡ－２ファミリー（ＳＨＡ－２２４，ＳＨＡ－２５６，ＳＨＡ－３８４，ＳＨＡ－５１２）も安全性が崩れる可能性が指摘されている。そこで、アメリカ国立標準技術研究所（ＮＩＳＴ）は、次世代の暗号学的ハッシュアルゴリズム（ＳＨＡ－３）を策定すべく新しいアルゴリズムを公募した。そして、２０１２年１０月に、ＫＥＣＣＡＫアルゴリズム（非特許文献１）がＳＨＡ－３のアルゴリズムとして選定された。

【０００３】

ＳＨＡ－３では、任意の長さの入力メッセージ（データ）に対して固定長の暗号学的ハッシュ値を出力する。ＫＥＣＣＡＫアルゴリズムにおいては、５つのステップ（θ、ρ、π、χ、ι）を順に適用するラウンド処理を２４回繰り返す置換関数（ｐｅｒｍｕｔａｔｉｏｎ　ｆｕｎｃｔｉｏｎ）が用いられている。また、ラウンド処理は、”ｓｔａｔｅ”と呼ばれる１６００ビット長のデータに対して実行される。

【先行技術文献】

【非特許文献】

【０００４】

【非特許文献１】“Ｔｈｅ　ＫＥＣＣＡＫ　ｒｅｆｅｒｅｎｃｅ”，　Ｖｅｒｓｉｏｎ　３．０，　Ｊａｎｕａｒｙ　１４，　２０１１，　（ｈｔｔｐ：／／ｋｅｃｃａｋ．ｎｏｅｋｅｏｎ．ｏｒｇ／Ｋｅｃｃａｋ－ｒｅｆｅｒｅｎｃｅ－３．０．ｐｄｆ）

【発明の概要】

【発明が解決しようとする課題】

【０００５】

ところで、ラウンド処理の５つのステップのうち、θ処理とπ処理はそれぞれ行うために、先行する処理の結果をたくさん、一旦メモリに貯める必要がある。そのため、１回のラウンド処理において、２回、先行する処理の結果をたくさん、一旦メモリに貯める必要があり、高速化が困難であった。

【０００６】

本発明は、上述の問題点に鑑みなされたものであり、ハッシュ値生成のスループット向上を可能とする技術を提供することを目的としている。

【課題を解決するための手段】

【０００７】

上述の問題点を解決するため、本発明のハッシュ値生成装置は以下の構成を備える。すなわち、ＳＨＡ－３アルゴリズムのラウンド処理に含まれるθ処理を実行するθ処理手段と、　前記ラウンド処理に含まれるρ処理を実行するρ処理手段と、前記ラウンド処理に含まれるπ処理を実行するπ処理手段と、前記ラウンド処理に含まれるχ処理を実行するχ処理手段と、前記ラウンド処理に含まれるι処理を実行するι処理手段を有し、　前記π処理手段は、ｐｌａｎｅ単位でデータ入力し、ｓｈｅｅｔ単位でデータ出力することを特徴とする。

【発明の効果】

【０００８】

本発明によれば、ハッシュ値生成のスループット向上を可能とする技術を提供することができる。

【図面の簡単な説明】

【０００９】

【図１】ＫＥＣＣＡＫアルゴリズムを説明するための図である。

【図２】データ構造を説明する図である。

【図３】ステップθの処理を説明する図である。

【図４】ステップρの処理を説明する図である。

【図５】ステップπの処理を説明する図である。

【図６】ステップχの処理を説明する図である。

【図７】ステップιの処理を説明する図である。

【図８】ステップιにおけるラウンド定数を示す図である。
【図９】ラウンド処理Ｒ′の概要を説明する図である。
【図１０】ステップρ′の処理を説明する図である。
【図１１】ステップθ１の処理を説明する図である。
【図１２】ステップθ２の処理を説明する図である。
【図１３】ステップθ２′の処理を説明する図である。
【図１４】動作タイミングチャートである。
【図１５】ＫＥＣＣＡＫアルゴリズムをｌａｎｅを単位として処理する場合の実装例の概略構成を示す図である。
【図１６】第１実施形態に係るＫＥＣＣＡＫアルゴリズムの実装例の概略構成を示す図である。
【図１７】図１６の構成をより詳細に示した図である。
【図１８】レジスタ２００４の実装例を示す図である。
【図１９】θ２＆ρ＆π回路２００５の実装例を示す図である。
【図２０】χ＆ι回路２００６の実装例を示す図である。
【図２１】第２実施形態に係るＫＥＣＣＡＣアルゴリズムの実装例の概略構成を示す図である。
【図２２】θ２＆ρ′＆χ＆ι回路２４０６の実装例を示す図である。
【図２３】ステップπの処理の特性を説明する図である。
【図２４】レジスタ２００４によるｓｈｅｅｔ－ｐｌａｎｅ単位変換を説明する図である。
【図２５】π回路２２０５の実装例を示す図である。
【発明を実施するための形態】
【００１０】
以下に、図面を参照して、この発明の好適な実施の形態を詳しく説明する。なお、以下の実施の形態はあくまで例示であり、本発明の範囲を限定する趣旨のものではない。
【００１１】
（第１実施形態）
本発明に係るハッシュ値生成装置の第１実施形態として、ＳＨＡ－３（ＫＥＣＣＡＫアルゴリズム）のハッシュ値を生成する装置を例に挙げて以下に説明する。なお、以下の説明において、具体的なデータ長やビット値が示されている場合があるが、本発明はこれらの具体的な値に限定されるものでは無い。
【００１２】
まず、ＫＥＣＣＡＫアルゴリズムについて説明する。なお、より詳細な仕様については、背景技術で示した非特許文献１に記載されている。
【００１３】
図１（ａ）は、ＫＥＣＣＡＫアルゴリズムの全体の概要を示す図である。１０１は、メッセージブロック（m_1～m_t）を表している。メッセージブロック（m_1～m_t）は、ハッシュ値生成の対象となる入力メッセージを１０２４ビット毎に分割することにより生成される。
【００１４】
１０２と１０３は初期値を表しており、ここでは初期値は全ビットが０である。ここでは、初期値の全ビットが０である例で説明するが、これに限定されない。また、初期値１０２の長さは、上述のメッセージブロックの長さと同じ１０２４ビットであり、初期値１０２と初期値１０３の長さの合計は１６００ビットである。１０４は、ビット単位の排他的論理和（ＸＯＲ）演算部を表している。つまり、ＸＯＲ演算部１０４は、２つの１０２４ビットの入力データに対し、各ビットで排他的論理和を計算した結果を１０２４ビットのデータとして出力する。
【００１５】
１０５は、置換関数（ｐｅｒｍｕｔａｔｉｏｎ　ｆｕｎｃｔｉｏｎ）であるＫＥＣＣＡ

Ｋ－ｆを表しており、２つの入力データを受け取り、２つのデータを出力する。１０５の詳細については図１（ｂ）を参照して後述する。１０６は、切り取り部を表しており、１０２４ビットの入力データから、必要なサイズだけ切り出して、出力する。１０７は、このアルゴリズムの計算結果である暗号学的ハッシュ値（すなわち、ハッシュ値）を表している。

【００１６】

図１（ｂ）は、置換関数であるＫＥＣＣＡＫ－ｆ１０５の概要を説明する図である。２０１はラウンド処理Ｒを表しており、２４回実行される。ラウンド処理Ｒの詳細は後述する。２０２と２０３は、入力データを表している。入力データ２０２の長さは、１０２４ビットである。また、入力データ２０２と入力データ２０３の長さの合計は、１６００ビットである。入力データ２０２及び入力データ２０３の２つが結合されて、ラウンド処理Ｒ２０１に入力される。２０４と２０５は、出力データを表している。出力データ２０４の長さは、１０２４ビットである。また、出力データ２０４と出力データ２０５の長さの合計は、１６００ビットである。

【００１７】

図１（ｃ）は、ラウンド処理Ｒ２０１の概要を説明する図である。上述したように、ラウンド処理Ｒ２０１においては、入力データと出力データの長さは共に１６００ビットである。ラウンド処理Ｒ２０１で、入力データに対し、後述する５つのステップの処理（θ処理３０１，ρ処理３０２，π処理３０３，χ処理３０４，ι処理３０５）を順に適用して出力データを生成する。

【００１８】

以下では、ＫＥＣＣＡＫアルゴリズムのラウンド処理で用いられるデータ構造及び上述の５つのステップの詳細について説明する。

【００１９】

図２（ａ）は、ラウンド処理Ｒ２０１の入出力時のデータ構造である”ｓｔａｔｅ”を説明する図である。上述したように入力データと出力データはともに１６００ビットである。そして、当該１６００ビットのデータは、３次元配列において、幅（ｘ軸方向）５ビット、高さ（ｙ軸方向）５ビット、奥行き（ｚ軸方向）６４ビットの直方体として表される。この直方体のデータ構造を”ｓｔａｔｅ”と呼ぶ。なお、詳細は、図２（ｆ）を参照して後述するが、直方体として表されるｓｔａｔｅ構造に対して、１６００ビットのデータは、ｚ軸方向、ｘ軸方向、ｙ軸方向の順に割り当てられる。

【００２０】

図２（ｂ）は、データ構造”ｐｌａｎｅ”を説明する図である。ｐｌａｎｅ構造は、ｘ－ｚ平面に平行な、幅５ビット、高さ１ビット、奥行き６４ビットの平面構造として表される。つまり、上述のｓｔａｔｅ構造は、ｐｌａｎｅ構造をｙ軸方向に５個重ねたものとして考えることができる。

【００２１】

図２（ｃ）は、データ構造”ｓｈｅｅｔ”を説明する図である。ｓｈｅｅｔ構造は、ｙ－ｚ平面に平行な、幅１ビット、高さ５ビット、奥行き６４ビットの平面構造として表される。つまり、上述のｓｔａｔｅ構造は、ｓｈｅｅｔ構造をｘ軸方向に５個横に並べたものとして考えることができる。

【００２２】

図２（ｄ）は、データ構造”ｌａｎｅ”を説明する図である。ｌａｎｅ構造は、ｚ軸に平行な、幅１ビット、高さ１ビット、奥行き６４ビットの直線構造として表される。つまり、上述のｓｔａｔｅ構造は、ｌａｎｅ構造をｘ－ｙ平面に沿って２５個寄せ集めたものとして考えることができる。図２（ｆ）は、１個のｓｔａｔｅ構造を構成する２５個のｌａｎｅの順番を表す図である。

【００２３】

図２（ｅ）は、データ構造”ｃｏｌｕｍｎ”を説明する図である。ｃｏｌｕｍｎ構造は、ｙ軸に平行な、幅１ビット、高さ５ビット、奥行き１ビットの直線構造として表される

。つまり、上述のｓｈｅｅｔ構造は、ｃｏｌｕｍｎ構造をｚ軸方向に６４個並べたものとして考えることができる。

【００２４】

なお、第１実施形態では、入力データが１６００ビットである場合について説明するが、本発明はこれに限定されるものでは無い。また、ｓｔａｔｅ構造のデータを、幅（ｘ軸方向）５ビット、高さ（ｙ軸方向）５ビット、奥行き（ｚ軸方向）６４ビットの直方体のデータ構造として扱う例について説明するが、これに限定されない。例えば、入力データが８００ビットであり、ｓｔａｔｅ構造のデータを、幅５ビット、高さ５ビット、奥行き３２ビットの直方体のデータ構造として扱ってもよい。

【００２５】

また、ｐｌａｎｅ構造、ｓｈｅｅｔ構造、ｌａｎｅ構造、ｃｏｌｕｍｎ構造は、ｓｔａｔｅ構造の幅（ｘ軸方向）、高さ（ｙ軸方向）、奥行き（ｚ軸方向）の各ビット数に応じて変更される。すなわち、ｓｔａｔｅ構造のデータが、ｘ軸方向にｍビット、ｙ軸方向にｎビット、ｚ軸方向にｓビットである場合、ｐｌａｎｅ構造は、ｘ軸方向にｍビット、ｙ軸方向に１ビット、ｚ軸方向にｓビットである平面構造である。ｓｈｅｅｔ構造は、ｘ軸方向に１ビット、ｙ軸方向にｎビット、ｚ軸方向にｓビットである平面構造である。ｌａｎｅ構造は、ｘ軸方向に１ビット、ｙ軸方向に１ビット、ｚ軸方向にｓビットの直線構造である。ｃｏｌｕｍｎ構造は、ｘ軸方向に１ビット、ｙ軸方向にｎビット、ｚ軸方向に１ビットの直線構造である。

【００２６】

次に、ＫＥＣＣＡＫ－ｆ１０５に入力される入力データ２０２及び入力データ２０３から、１回目のラウンド処理Ｒ２０１の入力データを作成する方法について説明する。まず、入力データ２０２及び入力データ２０３を順に連結して、１６００ビットのデータブロックを生成する。次に、１６００ビットのデータを、６４ビット毎に分割し、２５個のｌａｎｅを生成する。最後に、２５個のｌａｎｅを、図２（ｆ）に示す番号順にｘ－ｙ平面に沿って配列し１個のｓｔａｔｅとして組み上げる。このようにして、生成したｓｔａｔｅ構造がラウンド処理Ｒ２０１に入力されることになる。なお、２４回目のラウンド処理Ｒ２０１の出力データから、出力データ２０４及び出力データ２０５を生成する方法についても同様であるため説明は省略する。

【００２７】

次に、ラウンド処理Ｒ２０１を構成する５つのステップ（ステップθ、ステップρ、ステップπ、ステップχ、ステップι）の処理について説明する。なお、各ステップにおいて、入力データと出力データのデータ構造は、ｓｔａｔｅ構造である。

【００２８】

図３（ａ）は、ステップθの処理（θ処理３０１）を説明する図である。ステップθは、各ビットに対して近傍の２つのｃｏｌｕｍｎの和を加える処理である。出力ｓｔａｔｅの各ビットは、入力ｓｔａｔｅのうち、”同じ場所にあるビットの値”と、”ｘ軸方向で－１の場所にあるｃｏｌｕｍｎのビットの和”と、”ｘ軸方向で＋１かつｚ軸方向で－１の場所にあるｃｏｌｕｍｎのビットの和”との３つ値の和として計算される。ここで、和とは、ＧＦ（２）上での和のことであり、排他的論理和の演算と同一の結果になる。式で書くと、次のようになる。

【００２９】

【数１】

$$a'[x][y][z] \leftarrow a[x][y][z] + \sum_{y'=0}^{4} a[x-1][y'][z] + \sum_{y'=0}^{4} a[x+1][y'][z-1]$$

【００３０】

ここで、ｘは０～４、ｙは０～４、ｚは０～６３である。

【００３１】

図３（ｂ）は、端の部分（例えばｘ＝０）のビットを求める場合におけるステップθの処理を説明する図である。ｘ＝０のビットを求めたい場合、”ｘ軸方向で－１の場所にあるｃｏｌｕｍｎ”は、ｓｔａｔｅの反対側、つまり”ｘ＝４の場所にあるｃｏｌｕｍｎ”に相当する。このように、ｓｔａｔｅからはみ出す座標については、ｓｔａｔｅの反対側の位置になる。つまり、座標値は同一ｓｔａｔｅ内で循環シフトする。このルールは、ｘ座標、ｙ座標、ｚ座標の何れも同じであり、また、他の４つのステップでも同様である。

【００３２】

図４は、ステップρの処理（ρ処理３０２）を説明する図である。ステップρは、ｚ軸方向に各ビットの値をシフトする処理である。より具体的には、図４（ａ）に示すように、ｓｔａｔｅの各ｌａｎｅ内の値を、指定されたビットだけｚ方向に循環シフトし出力する。各ｌａｎｅにおいてシフトするビット数は、予め定められており、図４（ｂ）に示している数字の通りである。尚、ρ処理を実行するために、予め、保持部に、図４（ｃ）に示すような、シフト量を示すテーブルを保持しておき、保持しているテーブルを用いて、ρ処理を実行する。

【００３３】

図５は、ステップπの処理（π処理３０３）を説明する図である。ステップπは、ｘ－ｙ平面内での各ビットの値の入れ替えを行う処理、つまり、同一ｓｔａｔｅ内の２５個のｌａｎｅを入れ替える処理を行う。尚、ｘ－ｙ平面は、”ｓｌｉｃｅ”とも呼ばれる。より具体的には、入力ｓｔａｔｅの各ｌａｎｅに対し図５（ａ）の上段に示すように番号をふった場合、出力ｓｔａｔｅは下段に示すようになる。尚、π処理を実行するために、予め、保持部に、図５（ｂ）に示すような、入替先を示すテーブルを保持しておき、保持しているテーブルを用いて、π処理を実行する。

【００３４】

図６は、ステップχの処理（χ処理３０４）を説明する図である。ステップχは、ｘ軸方向（”ｒｏｗ”とも呼ばれる）のビット列内での変換を行う処理であり、出力ｒｏｗの各ビットの値は、同一の入力ｒｏｗの３つのビットに基づき導出される。より具体的には、出力ｒｏｗの各ビットの値は、入力ｒｏｗの各ビットに対し、ｘ軸方向で＋１の場所にあるビットが０、かつ、ｘ軸方向で＋２の場所にあるビットが１の場合にビットの値が反転するように設定される。

【００３５】

図７は、ステップιの処理（ι処理３０５）を説明する図である。ステップιは、各ビットにラウンド定数を加える処理である。また、図８は、ステップιにおけるラウンド定数を示す図である。ステップιは、ｘ＝ｙ＝０のｌａｎｅのビット列に対して、ラウンド毎に予め定められたラウンド定数（６４ビット値）との排他的論理和（ＸＯＲ）を適用する。具体的には、ｘ＝ｙ＝０のｌａｎｅの６４ビット値（ｚ＝６３のビットをＭＳＢ、ｚ＝０のビットをＬＳＢとする）と、図８に示されるラウンド定数とのビット毎の排他的論理和を計算する。そして、その結果を、出力ｓｔａｔｅにおけるｘ＝ｙ＝０のｌａｎｅのビット列として設定する。

【００３６】

上述した、各ステップ（ステップθ，ステップρ，ステップπ，ステップχ，ステップι）の処理内容から、各ステップの処理を開始するにあたり以下の制約があることが分かる。

【００３７】

・ステップθは、ｓｔａｔｅ内の各ｌａｎｅの計算において、ｘ軸方向に関して－１のｓｈｅｅｔと＋１のｓｈｅｅｔのデータを使用する。そのため、最初の３つ分のｓｈｅｅｔが完全に揃う時、つまり、２５個のｌａｎｅのうち２３個のｌａｎｅを前段の処理から受け取った時、ステップθの処理を開始することができる。

【００３８】

・ステップρは、ｌａｎｅ毎に独立した計算である。そのため、前段（ステップθ）の計算結果が１個のｌａｎｅ分出力された時点で、ステップρの処理を開始することができ

る。
【００３９】
・ステップπは、ｓｔａｔｅ内の各ｌａｎｅを入れ替える。そのため、前段（ステップρ）の計算結果が１個のｓｔａｔｅ全体、すなわち２５個のｌａｎｅ分出力された時点で、ステップπの処理を開始することができる。
【００４０】
・ステップχは、ｓｔａｔｅ内の各ｌａｎｅの計算において、ｘ軸方向で＋１のｌａｎｅ及びｘ軸方向で＋２のｌａｎｅを使用する。そのため、３個目のｌａｎｅのデータを受け取った時点で、ステップχの処理を開始することできる。
【００４１】
・ステップιは、ｌａｎｅ毎に独立した計算である。そのため、前段（ステップχ）の計算結果が１個のｌａｎｅ分出力された時点で、ステップιの処理を開始することができる。
【００４２】
すなわち、ステップθ及びステップπ及びステップχにおいては、前段のステップの計算結果がそれぞれ２３個、２５個、３個のｌａｎｅ分出力されるまで、処理を開始することができない。このように、特にステップθ及びステップπの２処理の実行開始は、前段の処理開始から長い時間待った後でなければならない。
【００４３】
つまり、ステップθとステップπのどちらかの開始時間を早めることができれば、スループットが向上する。また、各ステップにおいて、ｌａｎｅ単位で処理するのではなく、ｐｌａｎｅやｓｈｅｅｔ単位で処理することで、スループットが向上する。
【００４４】
＜ラウンド処理Ｒ´８０１の説明＞
次に、ラウンド処理Ｒ´８０１について説明する。ラウンド処理Ｒ´８０１は、本実施形態で用いる処理であり、ラウンド処理Ｒ２０１と同じ結果になるように設計されている。
【００４５】
図９（ａ）は、ラウンド処理Ｒ´８０１の概要を説明する図である。ラウンド処理Ｒ´８０１は、処理結果がラウンド処理Ｒ２０１と同じになるように設計されている。ラウンド処理Ｒ´８０１は、入力データに対し、６つのステップの処理（θ１処理８０２、θ２処理８０３、ρ処理８０４、π処理８０５、χ処理８０６、ι処理８０７）を適用して出力データを生成する。
【００４６】
ここで、ρ処理８０４、π処理８０５、χ処理８０６、ι処理８０７は、それぞれ、ラウンド処理Ｒ２０１におけるρ処理３０２、π処理３０３、χ処理３０４、ι処理３０５と、処理は同じである。θ１処理８０２とθ２処理８０３は、ラウンド処理Ｒ２０１におけるθ処理３０１の処理を分離したものである。
【００４７】
ラウンド処理Ｒ´８０１内の処理のうち、ρ処理、χ処理、ι処理はラウンド処理Ｒ２０１内のものと同じ処理であるため、説明は省略する。
【００４８】
π処理８０５は、ラウンド処理Ｒ２０１内のものと同じ処理であるが、ｓｔａｔｅデータを保持してから処理するのはなく、ｐｌａｎｅデータ入力し、ｓｈｅｅｔデータ出力を行う。詳細は後述する。
【００４９】
以下では、θ１処理、θ２処理について説明する。
【００５０】
図１１は、ステップθ１の処理を説明する図である。ステップθ１は、ステップθの前半の演算に対応しており、ｃｏｌｕｍｎ和算出処理を実行するステップである。具体的に

は、ｃｏｌｕｍｎ毎に、”ｘ軸方向で－１の場所にあるｃｏｌｕｍｎのビットの和”と、”ｘ軸方向で＋１かつｚ軸方向で－１の場所にあるｃｏｌｕｍｎのビットの和”の２つの値の和（θ中間値と呼ぶことにする）を計算するための処理である。５個のｓｈｅｅｔデータを受け取った後に、各ｃｏｌｕｍｎに対して１ビットずつ、合計５×６４ビット分のθ中間値を出力する。θ中間値全体の構造は、ｘ－ｚ平面に平行な、幅５ビット、高さ１ビット、奥行き６４ビットの平面構造として表される。

【００５１】

図１２は、ステップθ２の処理を説明する図である。ステップθ２は、ステップθの後半の演算に対応しており、ｃｏｌｕｍｎ和加算処理を実行するステップである。つまり、ステップθ２は、ステップθ１で求めたθ中間値を、各ビットに加算するステップである。

【００５２】

上述した、各ステップ（ステップθ１，ステップθ２）の処理内容から、各ステップの処理を開始するにあたり以下の制約があることが分かる。

【００５３】

・ステップθ１は、和の計算であるため、ｓｔａｔｅ内の各ｓｈｅｅｔが入力される毎に、計算途中のθ中間値を更新していく処理となる。そのため、前段の計算結果が１つのｓｈｅｅｔデータが出力された時点で、ステップθ１の処理を開始することができる。

【００５４】

・ステップθ２は、ｓｔａｔｅ内の各ｐｌａｎｅの計算において、ステップθ１で計算したθ中間値を加算する。ステップθ２開始時点で、ステップθ１の実行は完了しているため、前段（レジスタ）の出力が１つのｐｌａｎｅデータが出力された時点で、ステップθ２の処理結果の出力を開始することができる。

【００５５】

・ステップρは、ｌａｎｅ毎に独立した計算である。そのため、前段（ステップθ２）の計算結果が１つのｐｌａｎｅデータが出力された時点で、ステップρの処理を開始することができる。

【００５６】

・ステップπは、ｓｔａｔｅ内の各ｌａｎｅを入れ替える。ただし、１個のｐｌａｎｅ分の入力に対し、１つのｓｈｅｅｔ分の出力が得られる。そのため、前段（ステップρ）の計算結果が１個のｐｌａｎｅデータが出力された時点で、ステップπの処理を開始することができる。

【００５７】

・ステップχは、ｓｔａｔｅ内の各ｌａｎｅの計算において、ｘ軸方向で＋１のｌａｎｅ及びｘ軸方向で＋２のｌａｎｅを使用する。そのため、３つのｓｈｅｅｔのデータがそろった時点で、ステップχの処理を開始することできる。

【００５８】

・ステップιは、ｌａｎｅ毎に独立した計算である。そのため、前段（ステップχ）の計算結果が１つのｓｈｅｅｔデータが出力された時点で、ステップιの処理を開始することができる。

【００５９】

ステップπにおいて、ｐｌａｎｅデータ入力し、ｓｈｅｅｔデータ出力を行うことで、ｓｔａｔｅデータを保持する必要がなくなり、スループットが向上する。

【００６０】

さらに、ステップθ２、ステップρ及びステップπについてはｐｌａｎｅ単位で、ステップχ及びステップιについてはｓｈｅｅｔ単位で行うことで、スループットが向上する。

【００６１】

以下では、ステップπは、ｐｌａｎｅデータ入力し、ｓｈｅｅｔデータ出力し、ラウンド処理をステップθ２及びステップρ及びステップπについてはｐｌａｎｅ単位で、ステ

ップχ及びステップιについてはｓｈｅｅｔ単位で行う構成について説明する。
【００６２】
＜装置構成および動作＞
図１６は、第１実施形態に係るＫＥＣＣＡＫアルゴリズムの実装例の概略構成を示す図である。１９０１は、入力データを表している。ここでは入力データは、ｌａｎｅ構造を単位として入力される。ただし、あらかじめ（ｘ，ｙ）＝（０，０），（０，１），（０，２），・・・のように、ｙ方向の順番で入力されることとする。１９０８は、入力データ１９０１である、ｌａｎｅ構造のデータを４個分保持し、５個目のｌａｎｅを受け取るタイミングで、ｓｈｅｅｔ構造を単位として、データを出力するレジスタである。なお、少なくとも、１つ目のｓｈｅｅｔ構造のデータが生成できるデータが保持されれば、１つ目のｓｈｅｅｔ構造のデータを出力するようにしてよい。１９０７は、出力データを表しており、計算が完了したときに、ｓｈｅｅｔ構造を単位として出力される。
【００６３】
１９０２は、排他的論理和（ＸＯＲ）演算部を表しており、ラウンド処理を２４回実行するたびに、メッセージブロックと内部データの排他的論理和を計算する。１９０３は内部データ全体を保持するレジスタである。尚、１９０３は、ｓｈｅｅｔ構造の入力データを入力して保持し、ｐｌａｎｅ構造の出力データを出力する。１９０４は、ステップθ２及びステップρ及びステップπを処理するための回路である。１９０４において、入力されるデータはｐｌａｎｅ構造のデータ、出力されるデータはｓｈｅｅｔ構造のデータとする。詳しくは後述する。１９０５は、ステップχ及びステップιを処理するための回路である。１９０６は、ステップθ１を処理するための回路である。
【００６４】
図１７は、図１６の構成をより詳細に示した図である。２００１は、入力データを表しており、図１６の入力データ１９０１と同じである。２００９は、レジスタであり、入力データ２００１のうち、少なくとも、１つ目のｓｈｅｅｔ構造のデータを保持して、ｓｈｅｅｔ構造を単位として出力する。２００９は、図１９のレジスタ１９０８と同じである。また、２００９は、１クロック毎に１個のｓｈｅｅｔ構造を単位として出力し、ここでは、ｘ座標の小さいｓｈｅｅｔから順に出力する。２００２は、マルチプレクサを表しており、入力データと内部ｓｔａｔｅとの間で排他的論理和の計算をするときは入力データをそのまま出力し、それ以外のときは０を出力する。
【００６５】
２００４は、内部データ全体を保持するレジスタであり、図１６のレジスタ１９０３と同じである。２００５はステップθ２及びステップρ及びステップπを計算するための回路（以下、θ２＆ρ＆π回路２００５と呼ぶ）である。上述のように、θ２＆ρ＆π回路２００５には、ｙ座標の小さい順にｐｌａｎｅ構造のデータが入力され（ｙ＝０，１，２，３，４の順番）、ｘ座標の小さい順にｓｈｅｅｔ構造のデータが出力される（ｘ＝０，１，２，３，４の順番）。
【００６６】
２００６は、ステップχ及びステップιを処理するための回路（以下、χ＆ι回路２００６と呼ぶ）であり、１個のｓｈｅｅｔ構造を単位として結果を出力する。２００７はマルチプレクサを表しており、ハッシュ値の計算開始時には初期化のため０を出力し、それ以外の時は計算途中のデータをそのまま出力する。
【００６７】
２００８は、ステップθ１の処理を行う回路（以下、θ１回路２００８と呼ぶ）であり、ｓｈｅｅｔデータを５個入力すると５×６４ビットの中間値（θ中間値）を出力する。
【００６８】
図１８は、レジスタ２００４の実装例を示す図である。２１０１は入力データを表している。２１０２は入力データ２１０１である、１つのｓｈｅｅｔデータを５つのｌａｎｅデータに分割し出力する組み合わせ回路である。２１０３はマルチプレクサを表しており、偶数ラウンドと奇数ラウンドでシフト方向を縦方向と横方向で交互に切り替える（図１

８内の同型記号の動作は全て２１０３と同様である）。２１０４は１段で１個のｌａｎｅの情報を記憶する５×５段のメッシュ構造のレジスタを表しており、１クロック毎に５個のｌａｎｅ分のデータ（１個のｓｈｅｅｔ分もしくはｐｌａｎｅ分のデータ）を入出力する。２１０５は２１０４の最終段から出力された５個のｌａｎｅ分のデータ（Ｒ０・Ｒ１・Ｒ２・Ｒ３・Ｒ４の５個、もしくはＲ０・Ｒ５・Ｒ１０・Ｒ１５・Ｒ２０の５個）を１個のｐｌａｎｅ分のデータとして出力する組み合わせ回路である。２１０６は出力を表している。

【００６９】

図２３は、ステップπの処理の特性を説明する図である。

【００７０】

図２３（ａ）は、ステップπの処理を実行する前のデータの例である。なお、πの処理により、各ｌａｎｅがどの位置に並び替えられるかをわかりやすくするため、各ｌａｎｅに連続する番号を割り振ってある。図２３（ｂ）は、ステップπの処理を実行した後のデータの例である。ここでｙ＝０の１個のｐｌａｎｅ分のデータ（図２３（ａ）の２３１）が全てｘ＝０の１個のｓｈｅｅｔ分のデータの位置（図２３（ｂ）の２３２）に並び替えられていることが分かる。また、同様の事がｙ＝１，２，３，４のデータにも言える。つまり、ステップπの処理を実行すると、ｙ＝ｉ（ｉ＝０，１，２，３，４）の１個のｐｌａｎｅ分のデータは、ｘ＝ｉの１個分のｓｈｅｅｔ分のデータとして出力される。

【００７１】

以上説明したステップπの処理の特性を用いることで、ステップπの処理を行う際に、全てのデータ（つまり、５つのｐｌａｎｅ構造のデータ）をレジスタに保持する必要はなく、１つのｐｌａｎｅ分の入力に対し、１個のｓｈｅｅｔ分の出力を得ることが可能となる。

【００７２】

図２４は、レジスタ２００４によるｓｈｅｅｔ－ｐｌａｎｅ単位変換を説明する図である。レジスタ２００４への入力データは、上述したように１クロックにつき１個のｓｈｅｅｔ分のデータである。図２４（ａ）は、１クロック目のｓｈｅｅｔデータが入力されたとき、図２４（ｂ）は、２クロック目のｓｈｅｅｔデータが入力されたとき、図２４（ｃ）は、５クロック目のｓｈｅｅｔデータが入力されたとき、の説明図である。図２４（ａ）、（ｂ）、（ｃ）に示した通り、５クロックかけて５個のｓｈｅｅｔ分のデータ（１個のｓｔａｔｅ分のデータ）をｙ方向（ｘ方向）に入力される。図２４（ｄ）は、５個のｓｈｅｅｔデータが入力された後、１つ目のｐｌａｎｅデータが出力されたとき、図２４（ｅ）は、２つ目のｐｌａｎｅデータが出力されるとき、の説明図である。５個のｓｈｅｅｔ分のデータが入力された後、ｘ方向（ｙ方向）に１クロックにつき５個のｌａｎｅ分のデータを出力する事で、１個のｐｌａｎｅ分のデータを得る。レジスタ２００４は以上の流れでｓｈｅｅｔ－ｐｌａｎｅ変換を行っている。

【００７３】

尚、レジスタへのデータ入力は、例えば、ｙ軸方向から５ｐｌａｎｅ入力し、その後、ｘ軸方向から５ｓｈｅｅｔ入力するなど、入力方向を、ｘ軸、ｙ軸交互に変えて行う。

【００７４】

図１９は、θ２＆ρ＆π回路２００５の実装例を示す図である。θ２＆ρ＆π回路２００５は、注目ｐｌａｎｅに対して２００８であらかじめ計算しておいたθ中間値を使用して、出力データを算出する。ステップπの処理を含むため、出力データはｓｈｅｅｔとなる。

【００７５】

２２０１は入力データを表している。上述したように、入力データとしては、１クロック毎に１個のｐｌａｎｅが入力される。２２０２は、θ１回路２００８からの入力を表しており、θ中間値に相当する。

【００７６】

２２０３は、排他的論理和（ＸＯＲ）演算部を表しており、この処理が上述したステッ

プθ２の演算に相当する。２２０４は、ステップρの演算を行う論理回路（ρ回路）である。２２０５は、ステップπの演算を行う論理回路（π回路）であり、１個のｐｌａｎｅ分の入力に対し、１個のｓｈｅｅｔ分の出力を得る。２２０６は出力データを表しており、１クロック毎に１個のｓｈｅｅｔが出力される。

【００７７】

図２５は、π回路２２０５の実装例を示す図である。２６０１は、入力データを表している。上述したように、入力データとしては、１クロック毎に１個のｐｌａｎｅが入力される。２６０２は入力データ２６０１である、１個のｐｌａｎｅ分のデータを５個のｌａｎｅ分のデータに分割し出力する組み合わせ回路である。

【００７８】

２６０３は、ステップπの並び替えを行う組み合わせ回路である。ｘ軸方向に並ぶ５個のｌａｎｅ分のデータを、図５（ｂ）のテーブルに従い、ｙ軸方向に並ぶ５個のｌａｎｅ分のデータに並び変え出力する。

【００７９】

２６０４は、５個のｌａｎｅ分のデータを１個のｓｈｅｅｔ分のデータとして出力する組み合わせ回路である。２６０５は出力データを表しており、１クロック毎に１個のｓｈｅｅｔが出力される。

【００８０】

図２０は、χ＆ι回路２００６の実装例を示す図である。χ＆ι回路２００６は、注目ｓｈｅｅｔと当該注目ｓｈｅｅｔに対してｘ軸方向に＋１と＋２にある２個のｓｈｅｅｔを使用して、出力データを算出する。

【００８１】

２３０１は、入力データを表している。上述したように、入力データとしては、１クロック毎に１個のｓｈｅｅｔが入力される。２３０２は、マルチプレクサを表しており、処理開始から最初の５クロックでは入力データ２３０１をそのまま出力し、引き続く２クロックは２３０４からのデータを出力する。

【００８２】

２３０３は、１段で１個のｓｈｅｅｔの情報を記憶する２段構成のレジスタを表している。２３０４は、１段で１個のｓｈｅｅｔの情報を記憶する２段構成のレジスタを表しており、ここでは、ｘ＝０とｘ＝１のｓｈｅｅｔの情報を記憶している。

【００８３】

２３０５は、組み合わせ回路を表しており、上述したステップχ及びステップιの演算を行う論理回路である。２３０６は、出力データを表しており、１個のｓｈｅｅｔを単位として出力される。

【００８４】

上述したように、θ２＆ρ＆π回路２００５をｐｌａｎｅ単位、χ＆ι回路２００６をｓｈｅｅｔ単位で処理することにより、θ２＆ρ＆π回路２００５の出力からχ＆ι回路２００６の入力までの経路は組み合わせ回路のみ接続されている。すなわち、ラッチ回路を含まない構成であるため、１クロック内でデータを通過させることが可能な構成となっている。

【００８５】

以上説明したように、第１実施形態では、πの処理を行う際に、ｐｌａｎｅ入力に対し、ｓｈｅｅｔ出力とし、ラウンド内の処理単位をｐｌａｎｅとｓｈｅｅｔにすることで、ラウンド内でのパイプライン化が可能となる。

【００８６】

また、以上説明したように、第１実施形態では、ラウンド開始時の入力処理単位はｐｌａｎｅ単位で、ラウンド終了時の出力処理単位はｓｈｅｅｔ単位である。が、レジスタ２００４の入出力時にｐｌａｎｅ単位からｓｈｅｅｔ単位変換を行うことで、次のラウンド開始の入力処理単位をｐｌａｎｅ単位にすることができる。

【００８７】

尚、第１実施形態ではθ１回路２００８への入力単位をｓｈｅｅｔ構造としたが、入力単位をｐｌａｎｅ構造としてもよい。その場合、θ１回路２００８は、ｐｌａｎｅデータを５個入力すると５×６４ビットのθ中間値を出力する。ただし、θ１回路２００８への入力単位をｓｈｅｅｔ構造とした方が、入力される度に、順次、θ中間値を計算できる効果がある。

【００８８】

図１４（ａ）は、第１実施形態に係る実装例における各モジュールの動作タイミングチャートである。なお、θ２＆ρ＆π回路２００５及びχ＆ι回路２００６はパイプライン処理されるよう構成されている。θ１回路２００８は、５個のｓｈｅｅｔが入力されて、θ中間値が算出できるので、５クロックかかる。θ２＆ρ＆π回路２００５は、１個のｓｈｅｅｔを出力してから2クロック後に、χ＆ι回路２００６が１個のｓｈｅｅｔ出力できる。つまり、χ＆ι回路２００６は、３個のｓｈｅｅｔが入力されたとき処理を開始でき、θ２＆ρ＆π回路２００５とχ＆ι回路２００６との並列動作が実現できる。また、１回のラウンド処理にかかる時間が、７クロックである。

【００８９】

＜仕様通りのアルゴリズムでｌａｎｅ単位で処理を行った場合の例＞

以下では、上述の第１実施形態の実装例に対する比較対象として、仕様通りのアルゴリズムでｌａｎｅを単位として処理する実装例について説明する。

【００９０】

図１５は、ＫＥＣＣＡＫアルゴリズムを仕様通りにｌａｎｅを単位として処理する場合の実装例の概略構成を示す図である。なお、５つのステップ（θ，ρ，π，χ，ι）の処理は上述したものと同じであるため説明は省略する。

【００９１】

１８０１は、入力データを表している。入力データ１８０１から、１クロック毎に１個のｌａｎｅ（６４ビット長のデータ）を受信する。なお、ｌａｎｅは、１個のｓｔａｔｅの中から図２（ｆ）に示される順に受信される。

【００９２】

１８０２は、排他的論理和の処理を表しており、ラウンド処理を２４回実行するたびに、メッセージブロックと内部データの排他的論理和を計算する演算部である。

【００９３】

１８０３は、ｓｔａｔｅで表される内部データ全体を保持するレジスタである。１８０４は、ステップπを実行する処理ブロック（π回路）である。ただし、上述したように、ステップπの処理は、ステップρの処理を完了した後でのみ実行可能となる。１８０５は、ステップθを実行する処理ブロック（θ回路）、１８０６は、ステップρを実行する処理ブロック（ρ回路）である。

【００９４】

１８０７は、ステップχを実行する処理ブロック（χ回路）、１８０８は、ステップιを実行する処理ブロック（ι回路）である。１８０９は、マルチプレクサであり、ラウンド処理の前半は１８０６からのデータを出力し、後半は１８０８からのデータを出力する。１８１０は、出力データを表しており、計算が完了したときに、１個のｌａｎｅを単位として出力される。

【００９５】

図１４（ｂ）は、仕様通りのアルゴリズムでｌａｎｅを単位として処理する場合の各モジュールの動作タイミングチャートである。θ回路１８０５＆ρ回路１８０６４とχ回路１８０７＆ι回路１８０８とは別々の期間に動作し、同時には動作しない。また、１回のラウンド処理にかかる時間は、５１クロックである。

【００９６】

＜比較＞

図１４（ａ）と図１４（ｃ）とを比較すると分かるように、第１実施形態の実装例の構成を用いることにより処理スループットが向上することがわかる。

【００９７】
すなわち、
・θ２＆ρ＆π回路及びχ＆ι回路の２つの処理回路の並列動作により回路利用効率が向上可能となる。
【００９８】
・より少ないクロック数（時間）で１回のラウンド処理を実行することが可能となる。
【００９９】
（第２実施形態）
本発明に係るハッシュ値生成装置の第２実施形態として、ＳＨＡ－３（ＫＥＣＣＡＫアルゴリズム）のハッシュ値を生成する装置を例に挙げて以下に説明する。なお、以下の説明において、具体的なデータ長やビット値が示されている場合があるが、本発明はこれらの具体的な値に限定されるものでは無い。ＫＥＣＣＡＫアルゴリズム、データ構造については第１実施形態と同様であり、以降では、第１実施形態と異なる箇所について説明する。
【０１００】
＜ラウンド処理Ｒ´９０１の説明＞
ラウンド処理Ｒ´９０１について説明する。ラウンド処理Ｒ´９０１は、本実施形態で用いる処理であり、ラウンド処理Ｒ２０１と同じ結果になるように設計されているが、ＫＥＣＣＡＫアルゴリズムの仕様とは処理内容が異なる。
【０１０１】
図９（ｂ）は、ラウンド処理Ｒ´９０１の概要を説明する図である。ラウンド処理Ｒ´９０１は、処理結果がラウンド処理Ｒ２０１と同じになるように設計されている。ラウンド処理Ｒ´９０１は、入力データに対し、６つのステップの処理（θ１処理９０２、π処理９０３、θ２´処理９０４、ρ´処理９０５、χ処理９０６、ι処理９０７）を適用して出力データを生成する。
【０１０２】
ここで、π処理９０３、χ処理９０６、ι処理９０７は、それぞれ、ラウンド処理Ｒ２０１におけるπ処理３０３、χ処理３０４、ι処理３０５と、処理は同じである。ρ´処理９０５は、ラウンド処理Ｒ２０１におけるρ処理３０２と同じように、ｚ軸方向に各ビットをシフトする処理であるが、シフトするビット数が異なる。θ１処理９０２とθ２´処理９０４は、ラウンド処理Ｒ２０１におけるθ処理３０１の処理を分離したものである。
【０１０３】
ラウンド処理Ｒ´９０１内の処理のうち、χ処理とι処理はラウンド処理Ｒ２０１内のものと同じ処理であるため、説明は省略する。
【０１０４】
π処理９０５は、ラウンド処理Ｒ２０１内のものと同じ処理であるが、ｓｔａｔｅデータを保持してから処理するのはなく、ｐｌａｎｅデータ入力し、ｓｈｅｅｔデータ出力を行う。詳細は第１の実施形態と同じである。
【０１０５】
また、ラウンド処理Ｒ´９０１のθ１処理は、ラウンド処理Ｒ´８０１のθ１処理と同じであるため、説明は省略する。以下では、ρ´処理、θ２´処理について説明する。
【０１０６】
図１０（ａ）は、ステップρ´の処理（ρ´処理９０５）を説明する図である。ステップρ´は、ステップρと同様に、ｚ軸方向に各ビットの値を循環シフトする処理である。ただし、各ｌａｎｅにおいて循環シフトするビット数はステップρと異なり、図１０（ｂ）に示している数字の通りである。尚、ρ´処理を実行するために、予め、保持部に、図１０（ｃ）に示すような、シフト量を示すテーブルを保持しておき、保持しているテーブルを用いて、ρ´処理を実行する。このテーブルは、π処理を考慮したテーブルである。詳細は後述する。

【０１０７】
ここで、ラウンド処理Ｒ´９０１の処理結果とラウンド処理Ｒ２０１の処理結果が同じであることを説明するために、まず、ラウンド処理Ｒ２０１の処理結果と、ラウンド処理Ｒ´´９１１の処理結果が同じであることを説明する。
【０１０８】
図９（ｃ）は、ラウンド処理Ｒ´´９１１の図である。ラウンド処理Ｒ´´９１１は、入力データに対し、５つのステップの処理（θ処理９１２、π処理９１３、ρ´処理９１５、χ処理９１６、ι処理９１７）を適用して出力データを生成するとする。ここで、θ処理９１２、π処理９１３、χ処理９１６、ι処理９１７は、ラウンド処理Ｒ２０１におけるθ処理３０１，π処理３０３，χ処理３０４，ι処理３０５と、それぞれ同じ処理である。ρ´処理９１５は、ラウンド処理Ｒ´９０１におけるρ´処理９０５と同じ処理である。
【０１０９】
ラウンド処理Ｒ２０１とラウンド処理Ｒ´´９１１を比較すると、ラウンド処理Ｒ２０１では、ρ処理３０２、π処理３０３の順に実行するのに対し、ラウンド処理Ｒ´´９１１では、π処理３０３、ρ´処理９０５の順で実行する点が異なる。
【０１１０】
ここで、ラウンド処理Ｒ２０１における、ステップρは、ｌａｎｅ毎に決められたルールで、ｚ軸方向にシフトするステップで、ステップπは、各ｌａｎｅを入れ替えるステップである。それに対して、ラウンド処理Ｒ´´９１１では、各ｌａｎｅを入れ替えるステップ（ステップπの処理）を先に行い、その後、入替処理を考慮したｌａｎｅ毎に決められたルールで、ｚ軸方向にシフトするステップ（ステップρ´の処理）を行う。つまり、ラウンド処理Ｒ´´９１１では、ステップπを先に行うが、ステップρ´で、ｚ軸方向にシフトするシフト量を、ステップπの処理を考慮して変更することで、ラウンド処理Ｒ´´９１１の処理結果とラウンド処理Ｒ２０１の処理結果が同じになる。
【０１１１】
図１０（ｃ）は、ステップρ´を行う際に用いる各ｌａｎｅのシフト量を示すテーブルである。
【０１１２】
図１０（ｃ）に示しているテーブルの生成方法について具体的に解説する。まず、ラウンド処理Ｒ２０１について考える。ラウンド処理Ｒ２０１では、ρ処理３０２とπ処理３０３を順に行う。図４（ｂ）に示している数字は、ステップρにおけるシフト量であり、たとえば、ｘ＝０，ｙ＝４の位置のｌａｎｅのシフト量は１８ｂｉｔであることを表している。次に、π処理によるｌａｎｅの入れ替えを、図５を使って確認すると、ｘ＝０，ｙ＝４の位置のｌａｎｅは、ｘ＝４，ｙ＝２の位置に移動する。
【０１１３】
次に、ラウンド処理Ｒ´´９１１について考える。ラウンド処理Ｒ´´９１１では、π処理９１３とρ´処理９１５を順に行う。ρ´処理の前にπ処理が行われているので、ρ´処理において１８ｂｉｔシフトしなければならないｌａｎｅは、ｘ＝４，ｙ＝２の位置にあるｌａｎｅとなる。よって、図１０（ｂ）に示している数字の、ｘ＝４，ｙ＝２の位置にある数字は、１８となる。他のｌａｎｅのシフト量も、同様にして求めることで、図１０（ｂ）に示している数字となる。
【０１１４】
つまり、図１０（ｃ）が示す、ステップρ´を行う際の各ｌａｎｅのシフト量を示すテーブルは、π処理の入替処理を考慮したテーブルである。
【０１１５】
次に、ラウンド処理Ｒ´´９１１の処理結果とラウンド処理Ｒ´９０１の処理結果が同じであることを説明する。
【０１１６】
ここで、π処理９０３、ρ´処理９０５、χ処理９０６、ι処理９０７は、ラウンド処

理Ｒ′′９１１におけるπ処理９１３、ρ′処理９１５、χ処理９１６、ι処理９１７と、それぞれ同じ処理である。θ１処理９０２、θ２′処理９０４は、θ処理９１２を分離した処理である。

【０１１７】

ラウンド処理Ｒ′′９１１とラウンド処理Ｒ′９０１を比較すると、ラウンド処理Ｒ′′９１１では、θ処理９１２、π処理９１３の順に実行するのに対し、ラウンド処理Ｒ′９０１では、θ１処理９０２、π処理９１３、θ２′処理９０４の順で実行する点が異なる。

【０１１８】

ここで、ラウンド処理Ｒ′′９１１において、ステップθは、各ビットに対して近傍の２つのｃｏｌｕｍｎの和を加えるステップであり、ステップπは、各ｌａｎｅを入れ替えるステップである。それに対して、ラウンド処理Ｒ′９０１では、各ビットに対して近傍の２つのｃｏｌｕｍｎの和を求める（ステップθ１の処理）。そして、その後、各ｌａｎｅを入れ替え（ステップπの処理）、各ｌａｎｅの入れ替えを考慮したビットにｃｏｌｕｍｎの和を加える処理（ステップθ２′の処理）を行う。

【０１１９】

図１３（ａ）は、ステップθ２′の処理を説明する図である。ステップθ２′は、ステップθの後半の演算に対応しており、ｃｏｌｕｍｎ和加算処理を実行するステップである。つまり、ステップθ２′は、ステップθ１で求めたθ中間値を、各ビットに加算するステップである。

【０１２０】

ただし、ステップθ２′においては、すでにステップπが実行されていることに注意する必要がある。具体的には、ラウンド処理Ｒ′′９１１のステップθ（つまり、ラウンド処理Ｒ２０１のステップθ）の場合は、各ビットのｘ座標と各ビットの計算に使用するθ中間値のｘ座標は等しいものとなる。しかし、ラウンド処理Ｒ′９０１のステップθ２′の場合は、各ビットのｘ座標と各ビットの計算に使用するθ中間値のｘ座標は異なり、ステップπの各ｌａｎｅの入れ替えを考慮したｘ座標となる。各ビットの計算に使用するθ中間値のｘ座標は、図１３（ｂ）に示している数字の通りである。尚、θ２′処理を実行するために、予め、保持部に、図１３（ｃ）に示すような、各ビットの計算に使用するθ中間値のｘ座標を示すテーブルを保持しておき、保持しているテーブルを用いて、θ２′処理を実行する。

【０１２１】

図１３（ｃ）に示しているテーブルの生成方法について具体的に解説する。まず、ラウンド処理Ｒ′′９１１について考える。ステップθにおける各ビットの計算に必要なθ中間値のｘ座標は、各ビットのｘ座標と等しい。たとえば、ステップθにおいて、ｘ＝０，ｙ＝４の位置のビットは、ｘ＝０のθ中間値を使って演算を行う。次に、ステップπによるｌａｎｅの入れ替えを、図５を使って確認すると、ｘ＝０，ｙ＝４の位置のビットは、ｘ＝４，ｙ＝２の位置に移動する。

【０１２２】

次に、ラウンド処理Ｒ′９０１について考える。ステップθ２′ではステップπがすでに行われているので、ステップθ２′において、ｘ＝４，ｙ＝２の位置にあるビットの計算に必要なθ中間値のｘ座標は、ｘ＝０であることがわかる。そのため、図１３（ｂ）に示している数字の、ｘ＝４，ｙ＝２の位置にある数字は、０となる。他のビットにおけるθ中間値のｘ座標も、同様にして求めることで、図１３（ｂ）に示している数字となる。

【０１２３】

つまり、図１３（ｃ）が示す、ステップθ２′を行う際のθ中間値のｘ座標を示すテーブルは、π処理の入替処理を考慮したテーブルである。

【０１２４】

以上説明したように、ラウンド処理Ｒ２０１の処理結果とラウンド処理Ｒ′′９１１の処理結果は同じであり、また、ラウンド処理Ｒ′′９１１の処理結果とラウンド処理Ｒ′

９０１の処理結果は同じである。従って、ラウンド処理Ｒ´９０１の処理結果とラウンド処理Ｒ２０１の処理結果は同じとなる。

【０１２５】

上述した、各ステップ（ステップθ１，ステップθ２´、ステップρ´）の処理内容から、各ステップの処理を開始するにあたり以下の制約があることが分かる。

【０１２６】

・ステップθ１は、和の計算であるため、ｓｔａｔｅ内の各ｐｌａｎｅが入力される毎に、計算途中のθ中間値を更新していく処理となる。そのため、前段の計算結果が１個のｐｌａｎｅデータが出力された時点で、ステップθ１の処理を開始することができる。

【０１２７】

・ステップθ２´は、ｓｔａｔｅ内の各ｐｌａｎｅの計算において、ステップθ１で計算したθ中間値を加算する。ステップθ２´開始時点で、ステップθ１の実行は完了しているため、前段（ステップπ）の計算結果が１個のｐｌａｎｅデータが出力された時点で、ステップθ２´の処理結果の出力を開始することができる。

【０１２８】

・ステップρ´は、ｌａｎｅ毎に独立した計算である。そのため、前段（ステップθ２´）の計算結果が１個のｐｌａｎｅデータが出力された時点で、ステップρ´の処理を開始することができる。

【０１２９】

すなわち、ステップθ１及びステップθ２´及びステップρ´においては、前段のステップの計算結果のうち１個のｐｌａｎｅデータが出力された時点で、処理を開始することができる。

【０１３０】

また、第２実施形態におけるステップπ、ステップχ、ステップιの処理内容は、第１実施形態で述べたものと同じであることから、各ステップの処理を開始するにあたり以下の制約があることが分かる。

【０１３１】

・ステップπは、ｓｔａｔｅ内の各ｌａｎｅを入れ替える。ただし、１個のｐｌａｎｅ分の入力に対し、１個のｓｈｅｅｔ分の出力が得られる。そのため、前段（ステップι）の計算結果が１個のｐｌａｎｅ分出力された時点で、ステップπの処理を開始することができる。

【０１３２】

・ステップχは、ｓｔａｔｅ内の各ｌａｎｅの計算において、ｘ軸方向で＋１のｌａｎｅ及びｘ軸方向で＋２のｌａｎｅを使用する。そのため、前段（ステップρ´）の計算結果が１個のｐｌａｎｅデータが出力された時点で、ステップχの処理を開始することできる。

【０１３３】

・ステップιは、ｌａｎｅ毎に独立した計算である。そのため、前段（ステップχ）の計算結果が１個のｐｌａｎｅデータが出力された時点で、ステップιの処理を開始することができる。

【０１３４】

ステップπにおいて、ｐｌａｎｅデータ入力し、ｓｈｅｅｔデータ出力を行うことで、ｓｔａｔｅデータを保持する必要がなくなり、スループットが向上する。

【０１３５】

さらに、ｐｌａｎｅ単位で処理を行い、さらに、ラウンド処理Ｒ２０１の代わりにラウンド処理Ｒ´９０１を用いることで、スループットが向上する。そこで、以下では、ステップπは、ｐｌａｎｅデータ入力し、ｓｈｅｅｔデータ出力し、ラウンド処理をｐｌａｎｅ単位で行う構成について説明する。

【０１３６】

＜装置構成および動作＞

図２１は、第２実施形態に係るＫＥＣＣＡＫアルゴリズムの実装例の構成を示す図である。２４０１は、入力データを表している。ここでは入力データは、あらかじめ（ｘ，ｙ）＝（０，０），（１，０），（２，０）・・・のように、ｘ方向の順番で入力されることとする。２４０９は、レジスタであり、入力データ２４０１である、ｌａｎｅ構造のデータを４個分保持し、５個目のｌａｎｅを受け取るタイミングで、ｐｌａｎｅ構造を単位として、データを出力する。なお、レジスタ２４０９では、少なくとも、１つ目のｐｌａｎｅ構造のデータが生成できるデータが保持されれば、１つ目のｐｌａｎｅ構造のデータを出力するようにしてよい。２４１０は、出力データを表しており、計算が完了したときに、ｓｈｅｅｔ構造を単位として出力される。

【０１３７】

２４０２は、マルチプレクサを表しており、入力データと内部ｓｔａｔｅとの間で排他的論理和の計算をするときは入力データをそのまま出力し、それ以外のときは０を出力する。

【０１３８】

２４０４は、ステップπを計算するための回路（以下、π回路２４０４と呼ぶ）である。π回路２４０４は、図１９のπ回路２２０５と同じであり、ｐｌａｎｅ構造のデータが入力され、ｓｈｅｅｔ構造のデータが出力される。具体的には、ｙ座標の小さい順にｐｌａｎｅ構造のデータが入力され（ｙ＝０，１，２，３，４の順番）、ｘ座標の小さい順にｓｈｅｅｔ構造のデータが出力される（ｘ＝０，１，２，３，４の順番）。

【０１３９】

２４０５は、内部データ全体を保持するレジスタである。レジスタ２４０５は、第１実施形態のレジスタ２００４と同じものであるため、説明は省略する。

【０１４０】

２４０６は、ステップθ２′及びステップρ′及びステップχ及びステップιを計算するための回路（以下、θ２′＆ρ′＆χ＆ι回路２４０６と呼ぶ）であり、１個のｐｌａｎｅ構造を単位として結果を出力する。２４０７は、マルチプレクサを表しており、ハッシュ値の計算開始時には初期化のため０を出力し、それ以外の時は計算途中のデータをそのまま出力する。

【０１４１】

２４０８は、ステップθ１の処理を行う回路（以下、θ１回路２４０８と呼ぶ）であり、ｐｌａｎｅデータを５個入力すると５×６４ビットの中間値（θ中間値）を出力する。

【０１４２】

図２２は、θ２′＆ρ′＆χ＆ι回路２４０６の実装例を示す図である。θ２′＆ρ′＆χ＆ι回路２４０６は、注目ｐｌａｎｅに対して２４０８であらかじめ計算しておいたθ中間値を使用して、出力データを算出する。

【０１４３】

２５０１は、入力データを表している。入力データとしては、１クロック毎に１個のｐｌａｎｅが入力される。２５０２はθ１回路２４０８からの入力を表しており、θ中間値に相当する。

【０１４４】

２５０３は、排他的論理和（ＸＯＲ）演算部を表しており、この処理が上述したステップθ２′の演算に相当する。２５０４は組み合わせ回路を表しており、上述したステップρ′及びステップχ及びステップιの演算を実際に行う論理回路であり、１クロック毎に１個のｐｌａｎｅが出力される。２５０５は出力データを表しており、１クロック毎に１個のｐｌａｎｅが出力される。

【０１４５】

上述したように、π回路２４０４をｐｌａｎｅ単位で処理し、出力されたｓｈｅｅｔ構造のデータをレジスタでｓｈｅｅｔ構造へ変換し、θ２′＆ρ′＆χ＆ι回路２４０６をｐｌａｎｅ単位で処理する。そして、θ２′＆ρ′＆χ＆ι回路２４０６の出力からπ回路２４０４の入力までの経路は組み合わせ回路のみ接続されている。すなわち、ラッチ回

路を含まない構成であるため、１クロック内でデータを通過させることが可能な構成となっている。

【０１４６】

尚、第２実施形態ではθ１回路２４０８への入力単位をｐｌａｎｅ構造としているが、入力単位をｓｈｅｅｔ構造としてもよい。その場合、θ１回路２４０８は、ｓｈｅｅｔデータを５個入力すると５×６４ビットのθ中間値を出力する。

【０１４７】

図１４（ｂ）は、第２実施形態に係る実装例における各モジュールの動作タイミングチャートである。なお、θ２′＆ρ′＆χ＆ι回路２４０６の出力からπ回路２４０４はパイプライン処理されるよう構成されている。ラウンド処理内の全てのステップの処理において、１個のｐｌａｎｅが入力されたとき処理を開始できる。つまり、ラウンド処理内の全てのステップの並列動作が実現可能である。また、１回のラウンド処理にかかる時間は、平均５クロックである。

【０１４８】

＜比較＞

以下では、上述の第２実施形態の実装例に対する比較対象として、仕様通りのアルゴリズムでｌａｎｅを単位として処理する実装例について説明する。仕様通りのアルゴリズムでｌａｎｅを単位として処理する実装例については、第１実施形態の内容と重複するため説明は省略する。

【０１４９】

図１４（ｃ）は、仕様通りのアルゴリズムでｌａｎｅを単位として処理する場合における各モジュールの動作タイミングチャートである。

【０１５０】

図１４（ｂ）と図１４（ｃ）とを比較すると分かるように、第２実施形態の実装例の構成を用いることにより処理スループットが向上することがわかる。

【０１５１】

すなわち、

・π回路及びθ２′＆ρ′＆χ＆ι回路の２つの処理回路の並列動作により回路利用効率が向上可能となる。

【０１５２】

・より少ないクロック数（時間）で１回のラウンド処理を実行することが可能となる。

【０１５３】

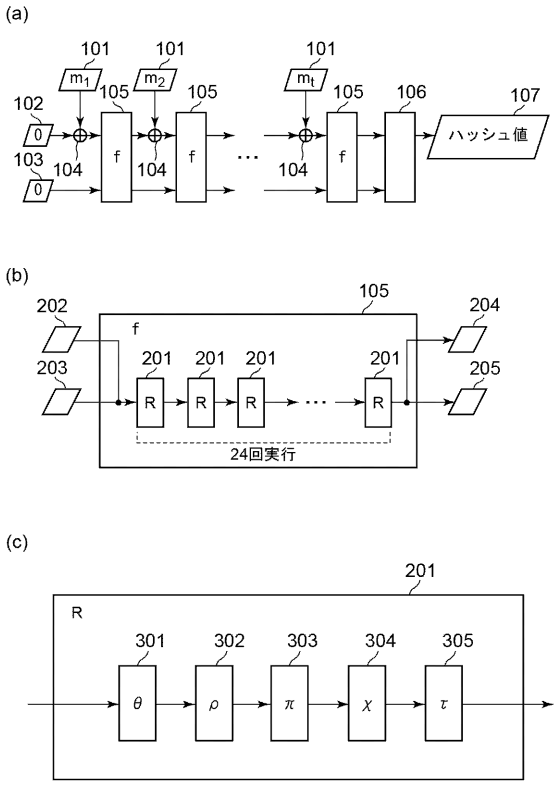
以上の本実施形態で説明したように、θ１処理を行っている間に、処理単位を変換することで、データ保持するための時間を削減することが可能である。

【０１５４】

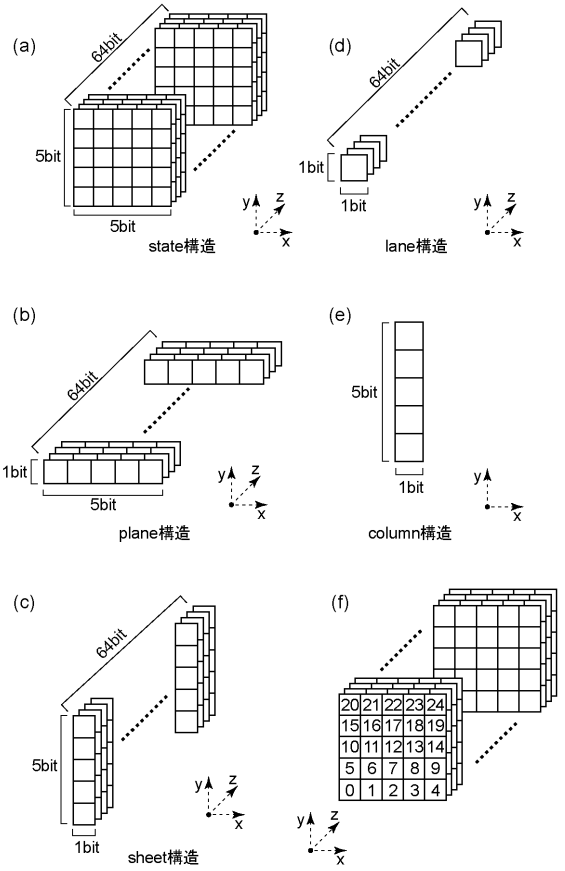
（その他の実施例）

また、本発明は、以下の処理を実行することによっても実現される。即ち、上述した実施形態の機能を実現するソフトウェア（プログラム）を、ネットワーク又は各種記憶媒体を介してシステム或いは装置に供給し、そのシステム或いは装置のコンピュータ（またはＣＰＵやＭＰＵ等）がプログラムを読み出して実行する処理である。

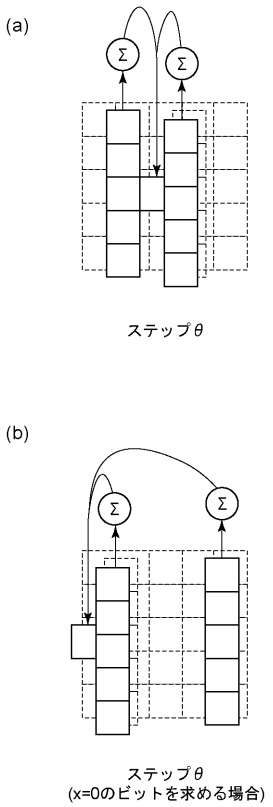
【図１】



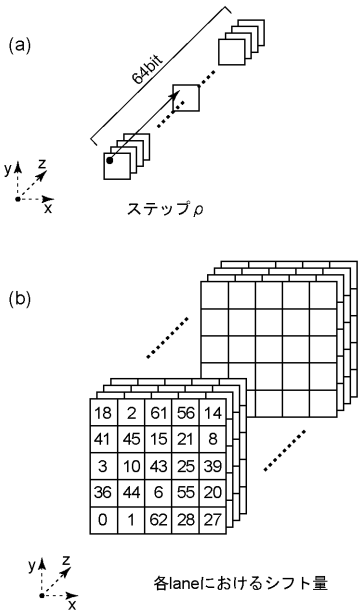
【図２】



【図３】



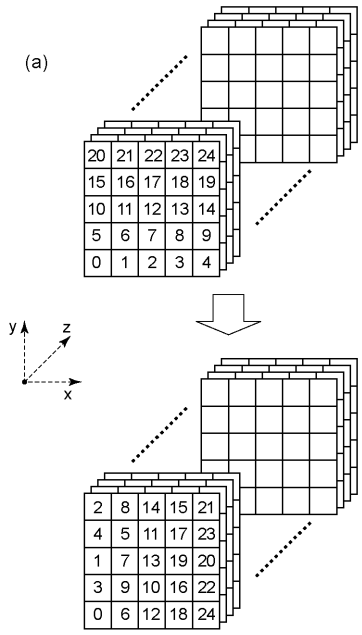
【図４】



(c)

18	2	61	56	14
41	45	15	21	8
3	10	43	25	39
36	44	6	55	20
0	1	62	28	27

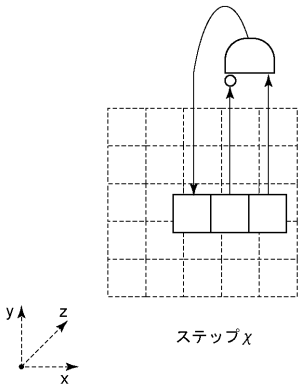
【図５】



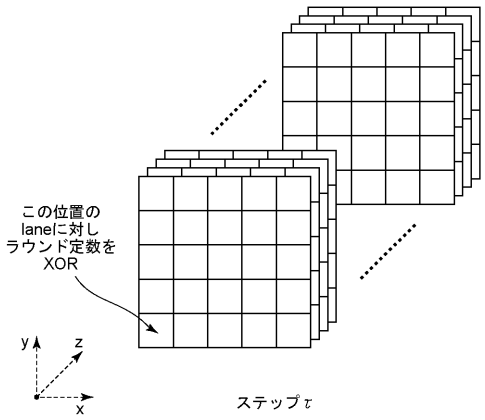
(b)

2	8	14	15	21
4	5	11	17	23
1	7	13	19	20
3	9	10	16	22
0	6	12	18	24

【図６】



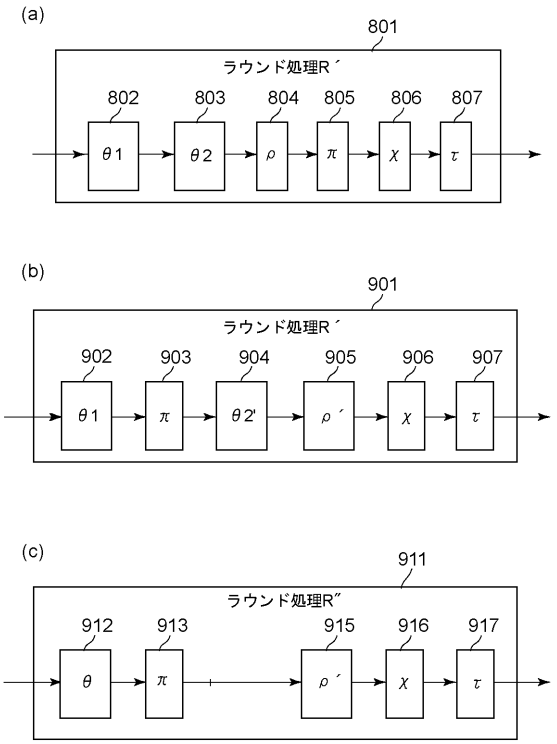
【図７】



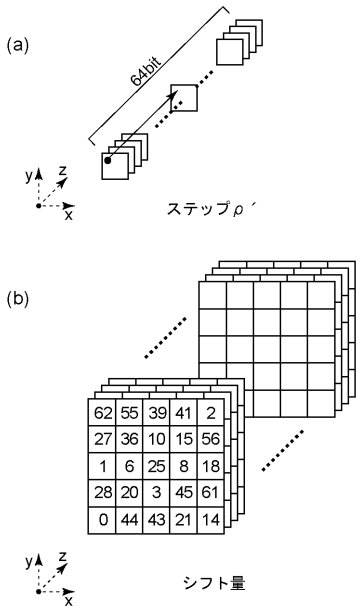
【図８】

ラウンド回数	ラウンド定数
1	0x0000000000000001
2	0x0000000000008082
3	0x800000000000808A
4	0x8000000080008000
5	0x000000000000808B
6	0x0000000080000001
7	0x8000000080008081
8	0x8000000000008009
9	0x000000000000008A
10	0x0000000000000088
11	0x0000000080008009
12	0x000000008000000A
13	0x000000008000808B
14	0x800000000000008B
15	0x8000000000008089
16	0x8000000000008003
17	0x8000000000008002
18	0x8000000000000080
19	0x800000000000800A
20	0x800000008000000A
21	0x8000000080008081
22	0x8000000000008080
23	0x0000000080000001
24	0x8000000080008008

【図９】



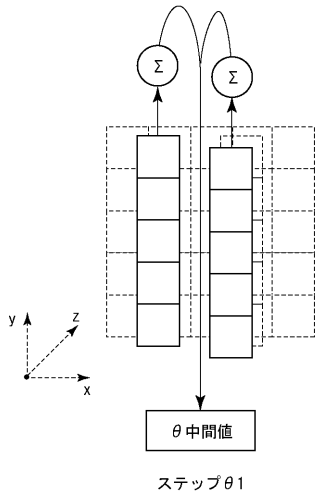
【図１０】



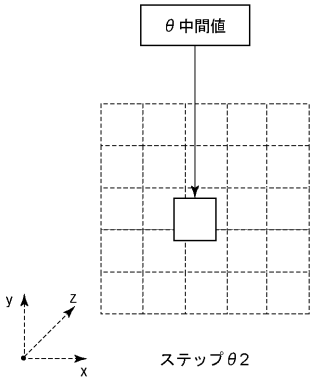
(c)

62	55	39	41	2
27	36	10	15	56
1	6	25	8	18
28	20	3	45	61
0	44	43	21	14

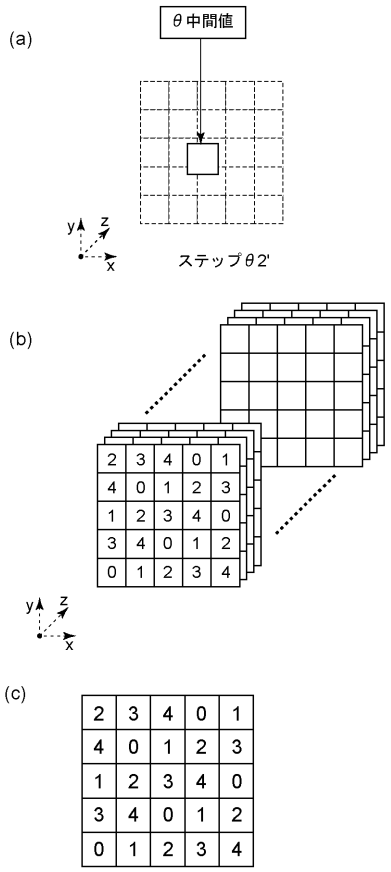
【図１１】



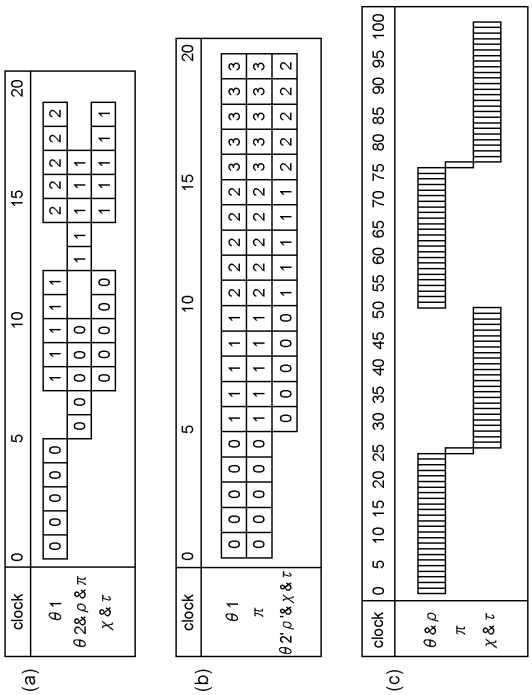
【図１２】



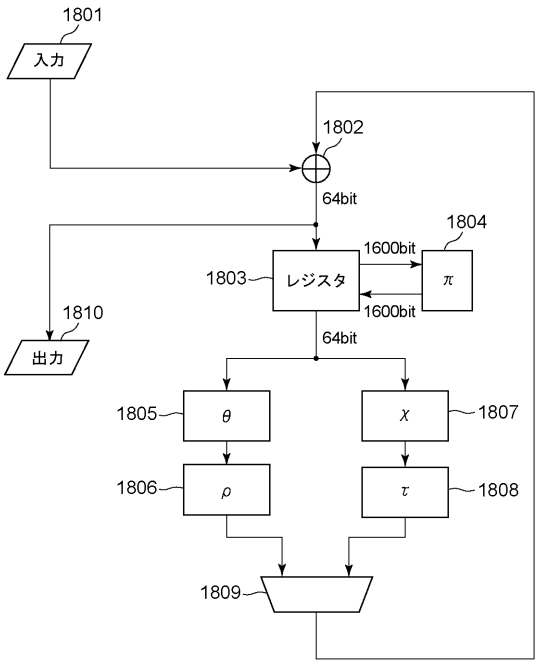
【図１３】



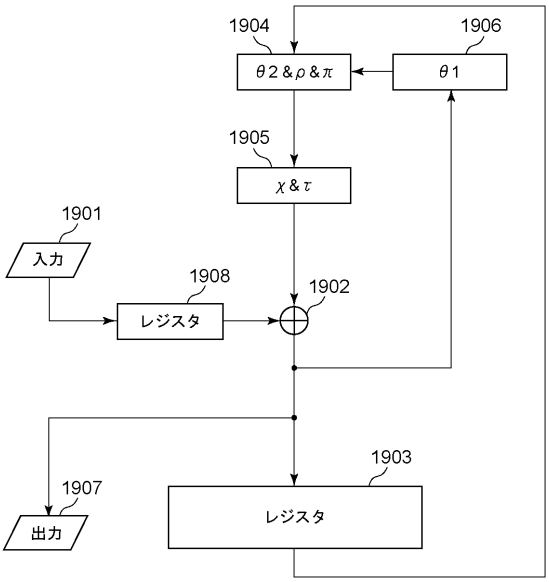
【図１４】



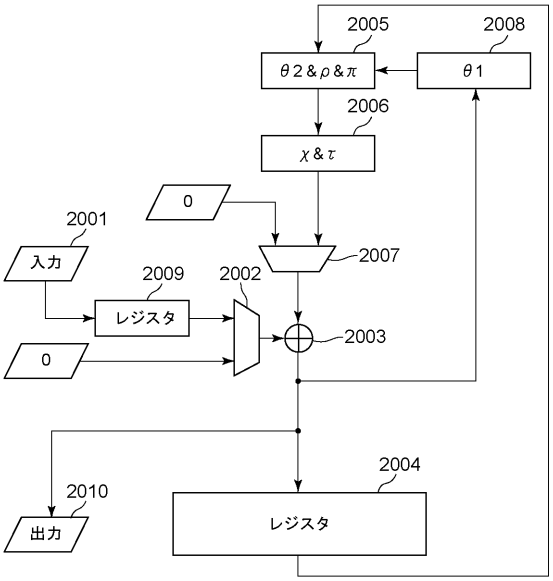
【図１５】



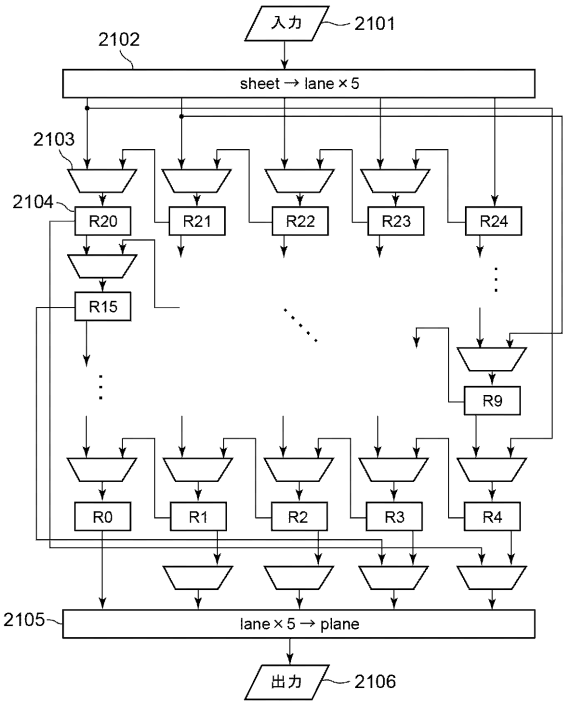
【図１６】



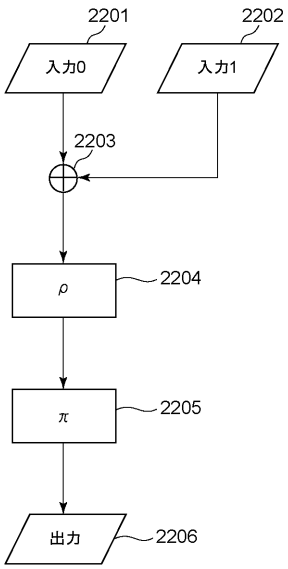
【図１７】



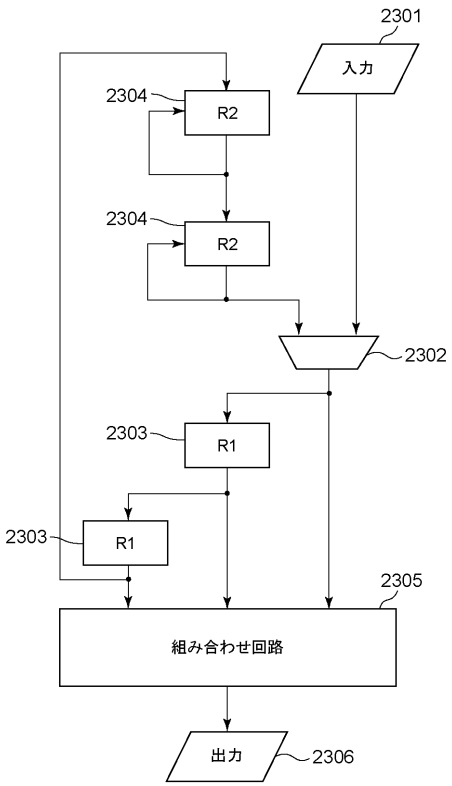
【図１８】



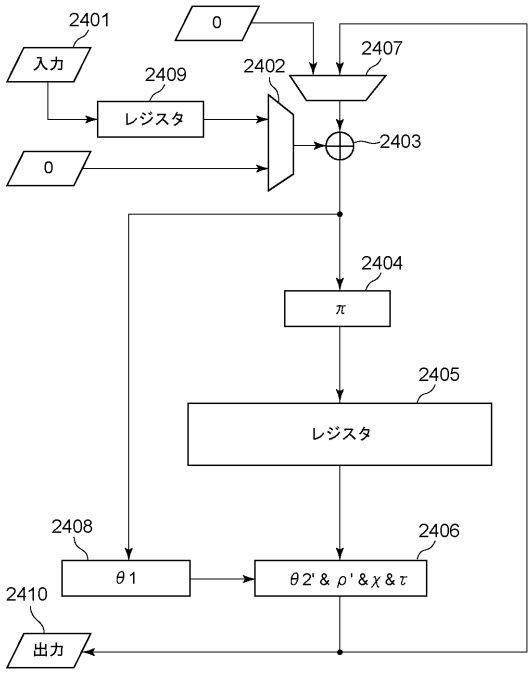
【図１９】



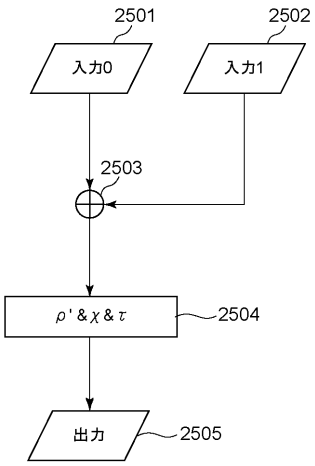
【図２０】



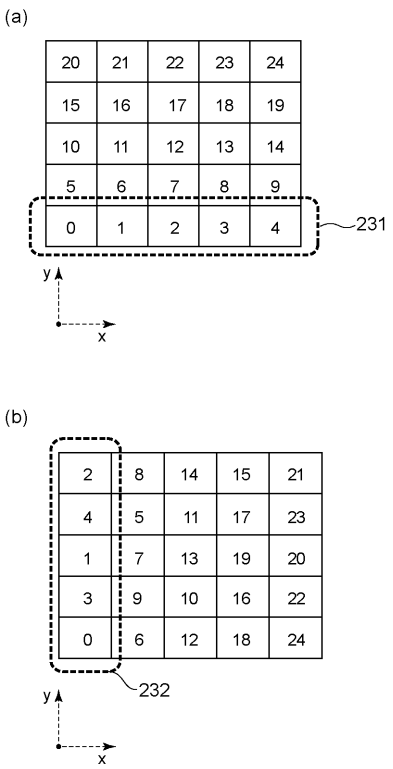
【図２１】



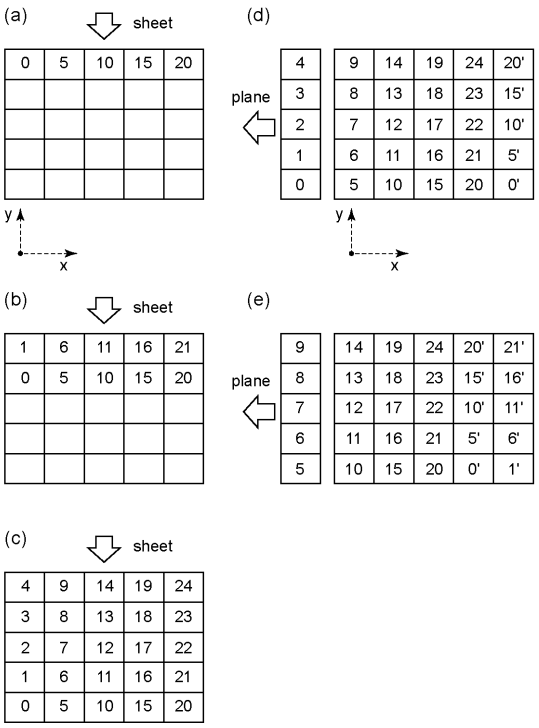
【図２２】



【図２３】



【図２４】



【図２５】