(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2009-282992
(P2009-282992A)
(43)公開日　平成21年12月3日(2009.12.3)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 17/10	(2006.01)	G06F	17/10	M	5B056
G09C 1/00	(2006.01)	G09C	1/00	650A	5J104

審査請求 有　請求項の数 1　ＯＬ　（全 13 頁）

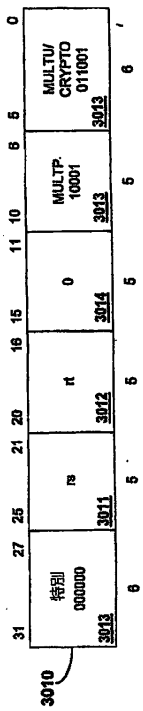
(21)出願番号　特願2009-162679 (P2009-162679)
(22)出願日　平成21年7月9日 (2009.7.9)
(62)分割の表示　特願2002-568191 (P2002-568191)の分割
原出願日　平成14年2月15日 (2002.2.15)
(31)優先権主張番号　09/788,684
(32)優先日　平成13年2月21日 (2001.2.21)
(33)優先権主張国　米国 (US)

(71)出願人　501166038
ミップス　テクノロジーズ　インコーポレイテッド
アメリカ合衆国カリフォルニア州９４０８５－４５２１，　サニーヴェール，イースト・アルケス・アヴェニュー　９５５
(74)代理人　100140109
弁理士　小野　新次郎
(74)代理人　100089705
弁理士　社本　一夫
(74)代理人　100075270
弁理士　小林　泰
(74)代理人　100080137
弁理士　千葉　昭男

最終頁に続く

(54)【発明の名称】多項式演算オペレーション

(57)【要約】　（修正有）
【課題】多項式演算インストラクションの実行性能を増大させる。
【解決手段】多項式演算インストラクション３０１０が、インストラクション設定アーキテクチャ（ＩＳＡ）中に提供される。乗算－加算多項式（ＭＡＤＤＰ）インストラクション、及び乗算－多項式（ＭＵＬＴＰ）インストラクション３０１３が提供される。
【選択図】図３Ａ



【特許請求の範囲】
【請求項１】
インストラクション・セット・アーキテクチャの一部分であるインストラクションであって、多項式演算を実行するためのインストラクションにおいて、
インストラクションを、多項式演算オペレーションを実行するためのインストラクションであるとして認識するための１又は複数のオペレーション・コードと、
１又は複数のレジスタ識別子と
を含み、１又は複数のレジスタ識別子を用いて多項式演算オペレーションを実行することによって処理されることを特徴とするインストラクション。
【請求項２】
請求項１記載のインストラクションにおいて、多項式演算オペレーションは、２進多項式加算であることを特徴とするインストラクション。
【請求項３】
請求項２記載のインストラクションにおいて、２進多項式加算は、マルチプライヤを用いて実行されることを特徴とするインストラクション。
【請求項４】
請求項１記載のインストラクションにおいて、多項式演算オペレーションの結果は、１又は複数のレジスタに格納されることを特徴とするインストラクション。
【請求項５】
請求項４記載のインストラクションにおいて、多項式演算オペレーションは、
１又は複数のレジスタ識別子によって識別されるレジスタの内容を乗算して中間値を得、
該中間値に１又は複数のレジスタの内容を加算して結果を得る
ことを含んでいることを特徴とするインストラクション。
【請求項６】
請求項５記載のインストラクションにおいて、得られた結果は、１又は複数のレジスタに格納されることを特徴とするインストラクション。
【請求項７】
請求項１記載のインストラクションにおいて、多項式演算オペレーションの結果は、上位結果レジスタ及び下位結果レジスタに格納されることを特徴とするインストラクション。
【請求項８】
請求項１記載のインストラクションにおいて、多項式演算オペレーションは、多項式乗算であることを特徴とするインストラクション。
【請求項９】
請求項８記載のインストラクションにおいて、１又は複数のレジスタ識別子によって識別されるレジスタはそれぞれ、多項式を含んでいることを特徴とするインストラクション。
【請求項１０】
請求項９記載のインストラクションにおいて、各多項式は、２進表記の係数として符号化されていることを特徴とするインストラクション。
【請求項１１】
請求項１記載のインストラクションにおいて、インストラクションのセットは、ＲＩＳＣインストラクションのセットであることを特徴とするインストラクション。
【請求項１２】
インストラクションを用いて多項式演算を実行する方法において、
インストラクションを受け取るステップであって、インストラクションが、
インストラクションを、多項式演算オペレーションを実行するためのインストラクションであるとして認識するための１又は複数のオペレーション・コードと、
１又は複数のレジスタ識別子と
を含んでいる、インストラクション受信ステップと、
インストラクションを処理することによって、１又は複数のレジスタ識別子を用いて多項式演算オペレーションを実行するステップと

からなることを特徴とする方法。
【請求項１３】
請求項１２記載の方法において、多項式演算オペレーションを実行するするテップは、２進多項式加算を実行するステップであることを特徴とする方法。
【請求項１４】
請求項１３記載の方法において、２進多項式加算はマルチプライヤを用いて実行されることを特徴とする方法。
【請求項１５】
請求項１２記載の方法において、該方法はさらに、多項式演算オペレーションの結果を、１又は複数のレジスタに格納するステップを含んでいることを特徴とする方法。
【請求項１６】
請求項１５記載の方法において、多項式演算オペレーションを実行するステップは、
１又は複数のレジスタ識別子によって識別されるレジスタの内容を乗算して中間値を得るステップと、
該中間値に１又は複数のレジスタの内容を加算して結果を得るステップと
を含んでいることを特徴とする方法。
【請求項１７】
請求項１６記載の方法において、該方法はさらに、得られた結果を、１又は複数のレジスタに格納するステップを含んでいることを特徴とする方法。
【請求項１８】
請求項１２記載の方法において、該方法はさらに、多項式演算オペレーションの結果を、上位結果レジスタ及び下位結果レジスタに格納するステップを含んでいることを特徴とする方法。
【請求項１９】
請求項１２記載の方法において、多項式演算オペレーションを実行するステップは、多項式乗算を実行するステップであることを特徴とする方法。
【請求項２０】
請求項１９記載の方法において、１又は複数のレジスタ識別子によって識別されるレジスタはそれぞれ、多項式を含んでいることを特徴とする方法。
【請求項２１】
請求項２０記載の方法において、各多項式は、２進表記の係数として符号化されていることを特徴とする方法。
【請求項２２】
請求項１２記載の方法において、インストラクションは、ＲＩＳＣインストラクションのセットの一部であることを特徴とする方法。
【請求項２３】
ソフトウエアにおいて実現されたマイクロプロセッサ・コアを含んだコンピュータにより読み取り可能な記憶媒体であって、マイクロプロセッサ・コアが、多項式演算を実行するためのインストラクションを含んでいる、コンピュータ読取可能記憶媒体において、インストラクションが、
インストラクションを、多項式演算オペレーションを実行するためのインストラクションであるとして認識するための１又は複数のオペレーション・コードと、
１又は複数のレジスタ識別子と
を含んでおり、インストラクションが、１又は複数のレジスタ識別子を用いて多項式演算オペレーションを実行することによって処理されることを特徴とするコンピュータ読取可能記憶媒体。
【請求項２４】
請求項２３記載のコンピュータ読取可能記憶媒体において、多項式演算オペレーションは、２進多項式加算あることを特徴とするコンピュータ読取可能記憶媒体。
【請求項２５】

請求項２４記載のコンピュータ読取可能記憶媒体において、２進多項式加算はマルチプライヤを用いて実行されることを特徴とするコンピュータ読取可能記憶媒体。
【請求項２６】
請求項２３記載のコンピュータ読取可能記憶媒体において、多項式演算オペレーションの結果は、１又は複数のレジスタに格納されることを特徴とするコンピュータ読取可能記憶媒体。
【請求項２７】
請求項２６記載のコンピュータ読取可能記憶媒体において、多項式演算オペレーションは、
１又は複数のレジスタ識別子によって識別されるレジスタの内容を乗算して中間値を得、
該中間値に１又は複数のレジスタの内容を加算して結果を得る
ことを含んでいることを特徴とするコンピュータ読取可能記憶媒体。
【請求項２８】
請求項２７記載のコンピュータ読取可能記憶媒体において、得られた結果は、１又は複数のレジスタに格納されることを特徴とするコンピュータ読取可能記憶媒体。
【請求項２９】
請求項２７記載のコンピュータ読取可能記憶媒体において、多項式演算オペレーションの結果は、上位結果レジスタ及び下位結果レジスタに格納されることを特徴とするコンピュータ読取可能記憶媒体。
【請求項３０】
請求項２３記載のコンピュータ読取可能記憶媒体において、多項式演算オペレーションは、多項式乗算であることを特徴とするコンピュータ読取可能記憶媒体。
【請求項３１】
請求項３０記載のコンピュータ読取可能記憶媒体において、１又は複数のレジスタ識別子によって識別されるレジスタはそれぞれ、多項式を含んでいることを特徴とするコンピュータ読取可能記憶媒体。
【請求項３２】
請求項３１記載のコンピュータ読取可能記憶媒体において、各多項式は、２進表記の係数として符号化されていることを特徴とするコンピュータ読取可能記憶媒体。
【請求項３３】
請求項２３記載のコンピュータ読取可能記憶媒体において、インストラクションは、ＲＩＳＣインストラクションのセットの一部であることを特徴とするコンピュータ読取可能記憶媒体。
【請求項３４】
公開キー暗号化システムにおける、公開キーにより情報を暗号化する方法において、該方法は多項式演算を実行するためのインストラクションを含み、該インストラクションは、
インストラクションを、多項式演算オペレーションを実行するためのインストラクションであるとして認識するための１又は複数のオペレーション・コードと、
１又は複数のレジスタ識別子と
を含み、インストラクションが、１又は複数のレジスタ識別子を用いて多項式演算オペレーションを実行することによって処理されることを特徴とする暗号化方法。
【請求項３５】
請求項３４記載の方法において、多項式演算オペレーションは、２進多項式加算であることを特徴とする方法。
【請求項３６】
請求項３５記載の方法において、２進多項式加算はマルチプライヤを用いて実行されることを特徴とする方法。
【請求項３７】
請求項３４記載の方法において、多項式演算オペレーションの結果は、１又は複数のレジスタに格納されることを特徴とする方法。

【請求項３８】
請求項３７記載の方法において、多項式演算オペレーションは、
１又は複数のレジスタ識別子によって識別されるレジスタの内容を乗算して中間値を得、
該中間値に１又は複数のレジスタの内容を加算して結果を得る
ことを含んでいることを特徴とする方法。
【請求項３９】
請求項３８記載の方法において、得られた結果は、１又は複数のレジスタに格納されることを特徴とする方法。
【請求項４０】
請求項３４記載の方法において、多項式演算オペレーションの結果は、上位結果レジスタ及び下位結果レジスタに格納されることを特徴とする方法。
【請求項４１】
請求項３４記載の方法において、多項式演算オペレーションは、多項式乗算であることを特徴とする方法。
【請求項４２】
請求項４１記載の方法において、１又は複数のレジスタ識別子によって識別されるレジスタはそれぞれ、多項式を含んでいることを特徴とする方法。
【請求項４３】
請求項４２記載の方法において、各多項式は、２進表記の係数として符号化されていることを特徴とする方法。
【請求項４４】
請求項３４記載の方法において、インストラクションは、ＲＩＳＣインストラクションのセットの一部であることを特徴とする方法。
【請求項４５】
マイクロプロセッサにおいて多項式演算を実行するためのインストラクションであって、
インストラクションを、多項式演算オペレーションを実行するためのインストラクションであるとして認識するための１又は複数のオペレーション・コードと、
１又は複数のレジスタ識別子と
を含み、１又は複数のレジスタ識別子を用いて多項式演算オペレーションを実行することによって処理されることを特徴とするインストラクション。
【請求項４６】
請求項４５記載のインストラクションにおいて、多項式演算オペレーションは、２進多項式加算であることを特徴とするインストラクション。
【請求項４７】
請求項４６記載のインストラクションにおいて、２進多項式加算は、マルチプライヤを用いて実行されることを特徴とするインストラクション。
【請求項４８】
請求項４５記載のインストラクションにおいて、多項式演算オペレーションは、
１又は複数のレジスタ識別子によって識別されるレジスタの内容を乗算して中間値を得、
該中間値に１又は複数のレジスタの内容を加算して結果を得る
ことを含んでいることを特徴とするインストラクション。
【請求項４９】
請求項４５記載のインストラクションにおいて、多項式演算オペレーションは、多項式乗算であることを特徴とするインストラクション。
【請求項５０】
多項式演算を実行するための１又は複数のインストラクションを提供するマイクロプロセッサにおいて、
インストラクション記憶部と、
インストラクション記憶部からマイクロプロセッサ・インストラクションを取得し、取得したインストラクションを処理する実行ユニットと、

実行ユニットによって取得したインストラクションを処理する際に、該インストラクションが多項式演算を実行するための１又は複数のインストラクションである場合に、使用される多項式演算ユニットと
からなることを特徴とするマイクロプロセッサ。
【請求項５１】
請求項５０記載のマイクロプロセッサにおいて、該マイクロプロセッサはさらに、乗算／除算ユニットを含んでいることを特徴とするマイクロプロセッサ。
【請求項５２】
請求項５１記載のマイクロプロセッサにおいて、多項式演算ユニットは、乗算／除算ユニットの構成要素であることを特徴とするマイクロプロセッサ。
【請求項５３】
請求項５０記載のマイクロプロセッサにおいて、多項式演算ユニットは、２進多項式加算を実行するよう動作可能であることを特徴とするマイクロプロセッサ。
【請求項５４】
請求項５０記載のマイクロプロセッサにおいて、多項式演算ユニットは、２進多項式乗算を実行するよう動作可能であることを特徴とするマイクロプロセッサ。
【請求項５５】
請求項５０記載のマイクロプロセッサにおいて、該マイクロプロセッサはさらに、多項式演算ユニットからの結果を記憶するための結果レジスタを備えていることを特徴とするマイクロプロセッサ。
【請求項５６】
請求項５５記載のマイクロプロセッサにおいて、多項式演算ユニットは、２進多項式乗算を実行して中間値を決定し、該中間値を結果レジスタの内容に加算することによって、２進多項式乗算及び加算オペレーションを実行するよう動作可能であることを特徴とするマイクロプロセッサ。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、多項式演算を実行するためのマイクロプロセッサ・インストラクション（命令）に関し、特に、多項式乗算を実行するためのマイクロプロセッサ・インストラクションに関する。
【背景技術】
【０００２】
産業傾向がより大きくより複雑なインストラクションのセットに傾いていくにつれ、縮小インストラクション・セット・コンピュータ（ＲＩＳＣ）アーキテクチャが開発された。インストラクションセット設計の単純化によって、ＲＩＳＣアーキテクチャは、パイプライン化（ｐｉｐｅｌｉｎｉｎｇ）及びキャッシング（ｃａｃｈｉｎｇ）等の技術の使用を容易にして、その結果、システム性能を増大させている。
【０００３】
ＲＩＳＣアーキテクチャは、通常、インストラクション形式に少しのバリエーションしか持たない固定長インストラクション（例えば１６ビット、３２ビット又は６４ビット）を持つ。インストラクション・セット・アーキテクチャ（ＩＳＡ）の各インストラクションは、常に同じ記憶位置にソース・レジスタを持つ。例えば、３２ビットのＩＳＡは、常にビット１６～２０及び２１～２５に指定されたソース・レジスタを持つ。こうすることで、指定されたレジスタが、いかなる複雑なインストラクションを解読することもなく、すべてのインストラクションに対して取り出されることが可能になる。
【発明の概要】
【発明が解決しようとする課題】
【０００４】
暗号化システム（「暗号システム」）は、トランザクションを保護し、通信を暗号化し

、ユーザを認証し、かつ情報を守るためにますます使用されている。デジタル・エンクリプション・スタンダード（ＤＥＳ）のような多くの秘密鍵暗号方式は、計算が比較的単純で、かつデータのブロック上で一連のＸＯＲ、ローテーション及び入替えを実行するハードウェア・ソリューションを縮小することが可能である。しかしながら、公開鍵暗号システムは、秘密鍵システムより数学的に難解で、計算がより困難である。

【０００５】

異なる公開鍵暗号化スキームは、数学的に異なる基盤を持つだけでなく、１０２４ビットという非常に大きな範囲の値での整数計算を一般的に必要とする傾向がある。この拡張精度算術は、多くの場合、モジュール方式（すなわち、ある値範囲を法として演算が実行される）であり、そして、ある場合には、２の補数ではなく多項形態である。例えば、ＲＳＡ公開鍵暗号システムは、情報を暗号化し復号化するために拡張精度モジュールの指数化を使用し、楕円曲線暗号システムは拡張精度モジュールの多項式乗算を使用する。

【０００６】

秘密鍵暗号システムは、通信チャンネルを暗号化するために広範囲に使用されているのに対して、公開鍵暗号システムは、ユーザ認証及び保護キーの交換に対して広範囲に使用されている。しかしながら、公開鍵暗号システムの使用が増加するにつれて、拡張精度モジュールの算術計算の性能を増大させることが望まれるようになった。

【課題を解決するための手段】

【０００７】

１つの一般的な側面において、インストラクション・セット・アーキテクチャは、多項式演算を行うためのインストラクションを含んでいる。該インストラクションは、１又は複数のオペレーション（演算）・コードを含み、該コードにより、該インストラクションを多項式演算オペレーションを実行するためのインストラクションとして、識別する。さらに、インストラクションは、１又は複数のレジスタを特定する。特定されたレジスタを用いて多項式演算オペレーションを実行することによって、該インストラクションは処理される。

【０００８】

実施例においては、２進多項式加算を実行するためのインストラクションも提供し、これは、乗算器（マルチプライヤ）を用いて実現される。多項式演算オペレーションの結果は、１又は複数の結果レジスタに格納される。多項式演算オペレーションは、乗算を含み、識別されたレジスタの内容が共に乗算される。オペレーションはまた、多項式乗算加算を含み、これにより、特定されたレジスタの内容がともに乗算され、そして１又は複数の結果レジスタの内容に加算される。結果レジスタは、上位レジスタ及び下位レジスタを含んでいる。多項式演算オペレーションは、レジスタに記憶された多項式上で実行される。これら多項式は、係数の２進表現として符号化されている。

【０００９】

１又は複数の実施例の詳細を、添附図面及び以下において説明する。他の機能及び利点は、その説明及び図面、並びに特許請求の範囲から明白になるであろう。

【図面の簡単な説明】

【００１０】

【図１】図１は、ＲＩＳＣアーキテクチャで使用される典型的な５ステージのパイプラインのブロック図である。

【図２】図２は、実行ユニット及び乗算／除算ユニットを含むプロセッサ・コアのブロック図である。

【図３Ａ】図３Ａは、多項式乗算及び加算を実行するインストラクションを例示するインストラクション符号化の図である。

【図３Ｂ】図３Ｂは、多項式乗算及び加算を実行するインストラクションを例示するインストラクション符号化の図である。

【発明を実施するための形態】

【００１１】

多くの公開鍵暗号システムは、データを暗号化し復号化するために、拡張精度モジュール算術を使用する。例えば、多くの楕円曲線（ＥＣ）暗号システムは、データを暗号化し復号化するために、広範囲に多項式の乗算及び加算を使用する。楕円曲線暗号システムの性能は、プログラム可能なＣＰＵマルチプライヤを、多項式演算専用に新しく定義されたインストラクションに応答するように修正することによって向上する。

【００１２】

ＧＦ（2^{163}）（ＩＥＥＥ１３６３－２０００基準によって推奨されるような）上で定義された楕円曲線を使用する時、必要とされる主要な演算は、フィールドＧＦ（2^{163}）上での乗算である。2^{163}のエレメントの各々は、０又は１に等しい係数を備えた多くて１６３項の多項式として表現されることができる。この表現では、２つのエレメントが単純なビットＸＯＲ（排他的論理和）を使用して加算され、また、２つの多項式ａ（Ｘ）及びｂ（Ｘ）が、ａ（Ｘ）ｂ（Ｘ）ｍｏｄ Ｐ（Ｘ）を計算することによって乗算されるが、ただし、積ａ（Ｘ）ｂ（Ｘ）は３２６項の多項式であり、Ｐ（Ｘ）はＩＥＥＥ１３６３－２０００基準によって特定されるような既約多項式である。

【００１３】

多項式乗算は、（１）通常の加算がＸＯＲに置き換えられること、及び（２）通常の３２ビット乗算が３２ビットの桁上げフリー乗算に置き換えられることを除いては、整数上で、ａｂｍｏｄ ｐというモジュール乗算と同じ形式を持つ。従って、多項式モジュール乗算は、シフト及び加算の代わりに、シフト及びＸＯＲを使用して実行される。

【００１４】

図１を参照して、多項式乗算を実現するために使用される典型的なマイクロプロセッサのアーキテクチャは、インストラクションがクロック・サイクル毎に発行され、かつ例えば４クロック・サイクルのような固定時間の中で実行される５段パイプラインを含む。各インストラクションの実行は、インストラクションフェッチ（ＩＦ）ステージ１００１、レジスタ読み取り（ＲＤ）ステージ１００２、算術／論理ユニット（ＡＬＵ）ステージ１００３、メモリ（ＭＥＭ）ステージ１００４、及びライトバック（書き戻し）（ＷＢ）ステージ１００５の５ステージに分割される。ＩＦステージ１００１では、指定されたインストラクションが、インストラクションキャッシュから取り出される。取り出されたインストラクションの一部が、インストラクションを実行するのに使用されるソースレジスタを指定するために使用される。読み取りレジスタ（ＲＤ）ステージ１００２では、システムが、指定されたソース・レジスタの内容を取り出す。これらの取り出された値は、ＡＬＵステージ１００３内の算術演算あるいは論理演算を実行するために使用される。ＭＥＭステージ１００４では、実行インストラクションが、データキャッシュ内のメモリを読み出し／書き込む。最後に、ＷＢステージ１００５では、インストラクションの実行によって得られた値が、レジスタにライトバックされる。

【００１５】

浮動小数点計算及び整数の乗算／除算のようないくつかの演算は、必ずしも単一クロック・サイクルで実行できるとは限らないので、いくつかのインストラクションはただインストラクションの実行を始めるためだけのものである。十分なクロック・サイクルが経過した後、別のインストラクションが結果を取り出すために使用される。例えば、整数乗算インストラクションが５つのクロック・サイクルを取る場合、１つのインストラクションが乗算計算を開始し、乗算が完成した後、別のインストラクションがその乗算の結果をレジスタに取り込む。結果が要求される時までに乗算が完了していない場合、結果が利用可能となるまで、パイプラインは時間を引き延ばす。

【００１６】

図２は、典型的なＲＩＳＣアーキテクチャを例示している。プロセッサ・コア２０００（又は「マイクロプロセッサ・コア」とも呼ばれる）は、実行ユニット２０１０、乗算／除算ユニット（ＭＤＵ）２０２０、システム制御コプロセッサ（ＣＰＯ）２０３０、メモリ管理ユニット２０４０、キャッシュ・コントローラ２０５０、及びバス・インターフェース・ユニット（ＢＩＵ）２０６０を含む。

【００１７】
実行ユニット２０１０は、プロセッサ・コア２０００内のインストラクションを実行するための主要なメカニズムである。実行ユニット２０１０は、レジスタ・ファイル２０１１及び算術論理ユニット（ＡＬＵ）２０１２を含む。１つの実施例では、レジスタ・ファイル２０１１が、例えば、スカラ整数演算及びアドレス計算に使用することができる３２個の３２ビット汎用レジスタを含む。２つの読み取りポート及び１つの書き込みポートを含むレジスタ・ファイル２０１１は、パイプライン内の演算待ち時間を最小限にするために完全にバイパスされる。ＡＬＵ２０１２は、加算、減算及びシフトのような論理と算術の両方の演算をサポートする。

【００１８】
ＭＤＵ２０２０は、乗算及び除算の演算（オペレーション）を実行する。一実施例においては、ＭＤＵ２０２０は、３２ビット×１６ビットのブース符号化マルチプライヤ（不図示）、結果累積レジスタ（ＨＩレジスタ２０２１及びＬＯレジスタ２０２２）、除算状態マシーン、並びに、これらの機能を実行するに必要な全てのマルチプレクサ及び制御論理ユニットを含んでいる。パイプライン化された実施例では、３２×１６乗算演算が、クロックサイクル毎にＭＤＵ２０２０に対して発生され、これにより、３２ビットの数がクロックサイクル毎に１６ビットの数と乗算される。しかしながら、その演算結果は、乗算が終了するまで、ＨＩ／ＬＯレジスタ（２０２１及び２０２２）において利用可能ではない。その演算結果は、インストラクションＭＦＨＩ及びＭＦＬＯによってアクセスされる。これらのインストラクションは、ＨＩレジスタ２０２１及びＬＯレジスタ２０２２から、結果を、指定されたレジスタに移動させる。例えば、インストラクション「ＭＦＨＩ＄７」は、ＨＩレジスタ２０２１の内容を、汎用レジスタ＄７に移動させる。

【００１９】
２つのインストラクション、すなわち乗算－加算（ＭＡＤＤ／ＭＡＤＤＵ）のインストラクションと乗算－減算（ＭＳＵＢ／ＭＳＵＢＵ）のインストラクションとは、乗算及び加算の演算を実行するため、並びに乗算及び減算の演算を実行するために使用される。ＭＡＤＤインストラクションは、２つの数を乗算した後に、その積をＨＩレジスタ２０２１及びＬＯレジスタ２０２２の現在の内容に加算する。得られた演算結果は、ＨＩ／ＬＯレジスタ（２０２１及び２０２２）に格納される。同様に、ＭＳＵＢインストラクションは、２つのオペランドを乗算した後に、その積をＨＩレジスタ２０２１及びＬＯレジスタ２０２２の現在の内容から減算し、その演算結果は、ＨＩ／ＬＯレジスタ（２０２１及び２０２２）に格納される。ＭＡＤＤ及びＭＳＵＢは、符号付きの値上で演算を行い、ＭＡＤＤＵ及びＭＳＵＢＵは、符号なしの値上でアナログ演算を実行する。

【００２０】
図３Ａは、乗算多項式（ＭＵＬＴＰ）インストラクション３０１０のインストラクション符号化を例示している。ＭＵＬＴＰインストラクション３０１０は、２つのレジスタ・フィールド（領域）ｒｓ３０１１及びｒｔ３０１２を備え、乗算されるべき多項式を格納しているソース・レジスタを特定する。乗算が完了すると、その結果がＨＩレジスタ２０２１及びＬＯレジスタ２０２２に格納される。ＭＵＬＴＰインストラクション３０１０はまた、実行すべき演算を識別するための１又は複数のオペレーション・コードを備えている。いくつかの実施例においては、インストラクション・フィールドの部分、例えばフィールド３０１４が使用されない。

【００２１】
一実施例においては、ｒｓ３０１１及びｒｔ３０１２によって特定されたレジスタは、２進多項式（すなわち、多項式の係数がモジュロ２で減じられる）を含んでいる。各係数は、「１」又は「０」である。多項式は、３２ビットのレジスタ内にコード化され、このとき、各ビットは多項式の係数を表す。例えば、多項式「x^4+x+1」は、x^3及びx^2の係数が「０」で、残りの係数が「１」であるので、「１００１１」としてコード化される。

【００２２】

ＭＵＬＴＰインストラクション３０１０は、２つの多項式の乗算を実行する。例えば、（ｘ^{4}＋ｘ＋１）（ｘ＋１）＝ｘ^{5}＋ｘ^{4}＋ｘ^{2}＋２ｘ＋１
である。多項式をモジュロ２で減じると、ｘ^{5}＋Ｘ^{4}＋ｘ^{2}＋１を生じる。多項式が上記の２進の表現でコード化される場合、同じ乗算が、（１００１１）（１１）＝１１０１０１として表わされる。

【００２３】

インストラクション及びオペランドのサイズは、任意に変えることができ、上記した３２ビットの例は単なる例示のためだけである。３２ビットの実施例において、ｒｓ３０１１に記憶された３２ビットのワード値は、ｒｔ３０１２に記憶された３２ビットのワード値によって、多項式ベースで乗算され、両方のオペランドが２進多項式の値として取り扱われ、６４ビットの演算結果が得られる。下位の３２ビット・ワード結果は、ＬＯレジスタ２０２２に記憶され、上位の３２ビット・ワード結果は、ＨＩレジスタ２０２１に記憶される。幾つかの実施例においては、算術的例外は生じない。ｒｓ３０１１及びｒｔ３０１２によって特定されたレジスタが３２ビットの正負符号拡張された値を含んでいない場合、演算結果は予測できない。

【００２４】

図３Ｂは、乗算－加算多項式（ＭＡＤＤＰ）インストラクション３０２０のインストラクション符号化を例示している。ＭＡＤＤＰインストラクション３０２０は、２つのパラメータ・フィールド（領域）ｒｓ３０２１及びｒｔ３０２２を備え、これにより、乗算されかつ多項式ベースでＨＩレジスタ２０２１及びＬＯレジスタ２０２２の内容に加算（ＸＯＲ）されるべき多項式を格納しているソース・レジスタを特定する。乗算及び加算が完了すると、その結果がＨＩレジスタ２０２１及びＬＯレジスタ２０２２に格納される。ＭＡＤＤＰインストラクション３０２０はまた、実行すべき演算を識別するための１又は複数のオペレーション・コードを備えている。いくつかの実施例においては、インストラクション・フィールドの部分、例えばフィールド３０２４が使用されない。

【００２５】

ＭＡＤＤＰインストラクション３０２０は、上記したように乗算を実行する。２進多項式加算は、ビットＸＯＲに類似している。例えば、２進の多項式加算（ｘ^{4}＋ｘ＋１）＋（Ｘ＋１）は、ｘ^{4}＋２ｘ＋２となる。係数をモジュロ２で減じると、「１００００」として表わされるｘ^{4}となる。

【００２６】

同様に、インストラクション及びオペランドのサイズは、任意に変えることができる。一実施例において、ｒｓ３０２１に記憶された３２ビットのワード値は、ｒｔ３０２２に記憶された３２ビットのワード値によって、多項式ベースで乗算され、両方のオペランドが２進多項式の値として取り扱われ、６４ビットの演算結果が得られる。そして、得られた結果がＨＩレジスタ２０２１及びＬＯレジスタ２０２２の内容と多項式ベースで加算される。下位の３２ビット・ワードは、ＬＯレジスタ２０２２に記憶され、上位の３２ビット・ワード結果は、ＨＩレジスタ２０２１に記憶される。ｒｓ３０２１及びｒｔ３０２２によって特定されたレジスタが３２ビットの正負符号拡張された値を含んでいない場合、演算結果は予測できない。

【００２７】

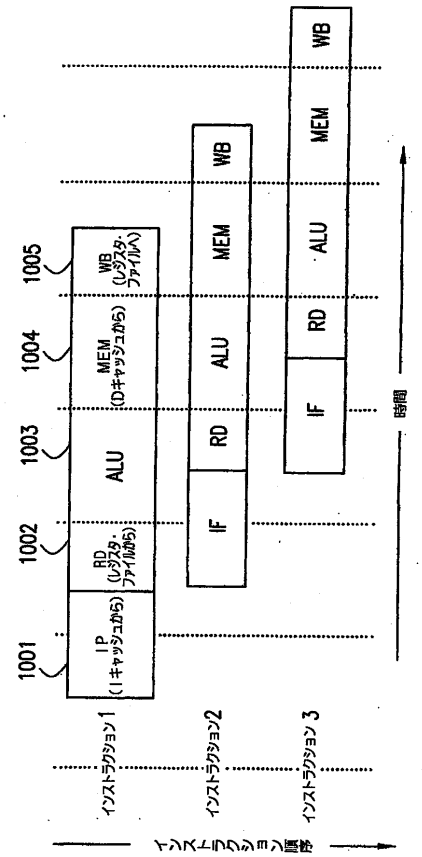
ハードウェア（例えば、マイクロプロセッサあるいはマイクロコントローラ内の）を使用するマルチプライヤの実施例に加えて、ソフトウェア（つまりコンピュータ読み取り可能なプログラムコード）を格納するように設定された、例えば、コンピュータが使用可能な（例えば、読み取り可能な）記憶媒体内に配列されたソフトウェアにおいても、マルチプライヤが具現化される。そのプログラム・コードは、ここに開示されたシステム及び技術の機能又は構成を、あるいはその両方を可能にする。例えば、これは、汎用プログラミング言語（例えばＣ、Ｃ＋＋）、Ｖｅｒｉｌｏｇ－ＨＤＬ、ＶＨＤＬ、ＡＨＤＬ（ＡｌｔｅｒａＨＤＬ）などを含むハードウェア記述言語（ＨＤＬ）、又は他の利用可能なプログラミング及び／又は回路（つまり回路図）キャプチャのツールの使用を通して達成される

ことができる。プログラム・コードは、半導体、磁気ディスク、光ディスク（例えばＣＤ－ＲＯＭ、ＤＶＤ－ＲＯＷ）、及びコンピュータ使用可能な（例えば、読み取り可能な）伝送記憶媒体（例えば、搬送波、あるいはデジタル、オプティカル又はアナログ・ベースの記憶媒体を含む他の記憶媒体）において具現化されるコンピュータ・データ信号のようなものを含む任意のよく知られたコンピュータ使用可能な記憶媒体内に配列されることができる。従って、コードは、インターネット及びイントラネットを含む通信ネットワーク上で伝送されることができる。

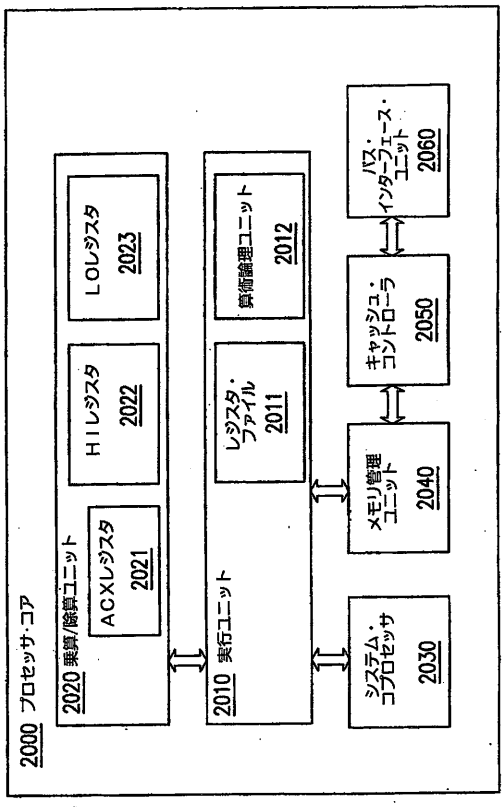
【００２８】

上に説明したシステム及び技術によって、達成される機能、及び／又は提供される構造が、プログラム・コードで実現されるコア（例えばマイクロプロセッサ・コア）内で表されることができ、またＩＣの製品の一部としてハードウェアに変換されることが、理解されるべきである。また、そのシステム及び技術は、ハードウェア及びソフトウェアの組合せとして実現されてもよい。従って、他の実施例も、特許請求の範囲内である。

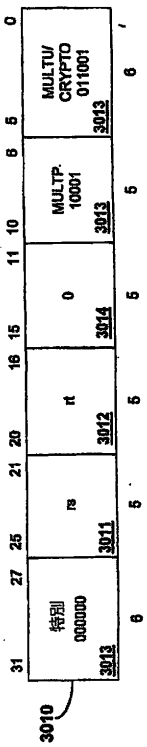
【図１】



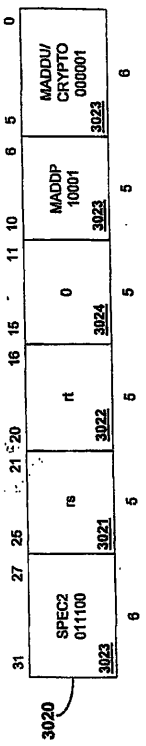
【図２】



【図３Ａ】



【図３Ｂ】



【手続補正書】
【提出日】平成21年8月7日(2009.8.7)
【手続補正１】
【補正対象書類名】特許請求の範囲
【補正対象項目名】全文
【補正方法】変更
【補正の内容】
【特許請求の範囲】
【請求項１】
インストラクション・セット・アーキテクチャの一部分であるインストラクションであって、多項式演算を実行するためのインストラクションにおいて、
インストラクションを、多項式演算オペレーションを実行するためのインストラクションであるとして認識するための１又は複数のオペレーション・コードと、
１又は複数のレジスタ識別子と
を含み、１又は複数のレジスタ識別子を用いて多項式演算オペレーションを実行することによって処理されることを特徴とするインストラクション。

フロントページの続き

(74)代理人 100096013
弁理士 富田 博行
(74)代理人 100153028
弁理士 上田 忠
(72)発明者 ストリベーク，モルテン
デンマーク国デーコー－２０００ フレデリクスベルイ，グスタフ・ヨハンセンズ・ヴェイ ３４
(72)発明者 キセル，ケヴィン・ディー
フランス国エフ－０６６２０ ル・バール・スュール・ルー，シュマン・デ・マルテル ３９
(72)発明者 パイリエ，パスカル
フランス国エフ－７５０２０ パリ，クール・ド・ヴァンセンヌ ３７
Ｆターム(参考) 5B056 AA05 BB74 FF01 FF02
5J104 AA20 AA21 JA25 NA16 NA17 NA39