完全自立形スケーラブルIPコアの設計法と楕円曲線暗号処理システムへの適用

佐藤正幸[†,☆] 中村次男[††] 畠中浩行[†]
冬爪成人[†] 笠原宏[†] 田中照夫[†††]

次世代集積回路の規模は数十から数百のIPコアからなると予測されており，IPコアの流通が不可欠となる．しかし，製造元の異なる多種多様なIPコアの仕様を理解して集積化および各IPコアを使用することは非常に困難である．また，ますます高精度化する処理データに対し，柔軟に対応しようとして，単にスケーラブルなアーキテクチャを用意しても，回路やソフトウェアの変更をともなうのでそのIPコアの詳細な仕様を理解することが不可欠となる．そのため，次世代の高集積回路においては多種多様なIPコア間インタフェースの標準化，再利用性，使用容易性および拡張性などが求められる．そこで，処理に必要なデータをIPコアに与えるとデータの長さに合った処理をIPコア内で判断し，結果を返すという，処理精度に対してスケーラブルなアーキテクチャを持ったIPコアの設計法を提案する．設計過程におけるスケーラビリティではなく，設計成果物としてのIPコアそのものがスケーラブルであり，外部からの制御をまったく必要とせず，かつ処理データの長さに制限されない，まさに完全自立形IPコアの実現である．これにより，以上に述べた諸問題がいかに解決できるかを，多くの多項式からなりモジュール化が困難であった楕円曲線暗号アクセラレータに，可変長鍵に対応可能なIPコアとして適用できることを示し，提案する設計法の有効性を検証している．

A Design Method of Self-supporting Scalable IP Core and Application to Elliptic Curve Cryptosystem Design

Masayuki Sato,[†,☆] Tsugio Nakamura,[††] Hiroyuki Hatakenaka,[†] Narito Fuyutsume,[†] Hiroshi Kasahara[†] and Teruo Tanaka[†††]

The next generation large scale IC is expected to be the collection of various IP cores, where, standardization of interface between each core, reusability, ease of use, and scalability, are unavoidable. In this paper, we propose a design method of IP core which has the architecture of scalability on arithmetic processing. For the input with any digit number into the core, it scales the least necessary procedure and returns the results with maximum efficiency of time and power expenditure. The *scalability* in this paper, does not means that in design procedure, but means a feature of design product (IP core), without external control circuit, and limitlessly adaptable to any digit numbers. We call the feature as *completely self-supportable*. As the proof of feasibility of the method, we show the result applied for designing an elliptic curve cryptosystem which is adaptable to any key-length.

1. はじめに

集積回路技術の向上は目覚ましく進歩し続けており，近い将来，数十から数百のIPコア（Intellectual Property Core）が集積されるというSoC（System on Chip），SiP（System in Package）およびシステムLSIが現実のものとなりつつある[1)]．このような超高集積回路においては1つの機関ですべての回路ブロックを初めから設計することは困難であることからIPコアの流通が不可欠となる．しかし，いろいろな機関から提供される多種多様なIPコアを1つのチップ上に集積化するとなるとIPコア間のインタフェー

† 東京電機大学情報環境学部情報環境工学科
Department of Information Environment Engineering School of Information Environment, Tokyo Denki University
†† 国際短期大学情報ネットワーク学科
Department of Information and Network, Kokusai Junior College
††† 東京電機大学工学部電気工学科
Department of Electrical Engineering School of Engineering, Tokyo Denki University
☆ 現在，株式会社日立超LSIシステムズ
Presently with Hitachi ULSI Systems

ス，再利用性および拡張性など多くの課題が山積している．また，多種多様な IP コアの仕様を理解して使いこなすことは非常に困難であり，特に使用容易性が重要性を増してくる．

以上のような諸問題を解決するため，次のような設計法を提案する．提案する方法を分かりやすくイメージするのに永久磁石を例にとる．基本となる同一の小さな永久磁石が数個あるとして，それぞれの磁石は単体でも磁石の機能を有している（自立している）．これを 2 個，3 個，と磁石を接触（続）すると数に比例した大きな磁石として機能する．使用者側から見るとどのような原理（仕様）であるか理解している必要はない．このようなことが IP コアで実現できれば次世代の超高集積回路における諸問題が解決できると考える．ここで基本となる小さな磁石に相当するのが，外部からの制御をまったく必要としないで単体でも機能する完全自立形のモジュールである．モジュール単体でも IP コアとして機能するが，通常，複数の同一モジュールを接続して IP コアは構成される．IP コアの処理に必要なデータが与えられると IP コア内でデータ長に合った必要なモジュール数，クロック周波数，クロック数および不必要なモジュールのスタンバイモード（消費電力の抑制）などを判断し，必要な精度で処理結果を返す（完全自立形）という，処理精度に対してスケーラブルなアーキテクチャを有する IP コアが実現できる[2)~6)]．設計過程におけるスケーラビリティや単なる機能の分割とは根本的に異なる[7)~9)]．

スケーラブルなアーキテクチャの要件と機能および制約の概要を以下に述べる．

(1) 基本となるモジュールの実現（基本モジュールの構成）

処理精度に柔軟に対応できるアーキテクチャとしての要件には基本モジュール間のインタフェースを考慮した設計が求められる（参照例：2.1 節）．

(2) 基本モジュールの再利用が可能なアーキテクチャ（再利用性）

流通する IP コアに求められる要件としては用途が特定されない VC（Virtual Component）として設計されていなければならない（参照例：図 3，図 7 および図 8）．

(3) 処理精度に対し，単純な同一モジュールのカスケード接続により対応可能（必要なモジュール数の判断機能，拡張性，使用容易性）

精度情報による動的な必要とするモジュール数の判断機能を持ったアーキテクチャの実現により拡張性と使用容易性が向上する（参照例：2.2 節）．

(4) 処理精度に合ったクロック周波数とクロック数の判断機能（プロセス遅延の考慮と効率的な処理）

必要とするモジュール数から遅延を考慮したクロック周波数および必要最小限のクロック数の判断機能を持ったアーキテクチャによりプロセス遅延を考慮した効率的な処理を実現（参照例：2.3 節）．

(5) 必要な精度に対応したモジュール構成には，回路の変更や外部回路からの制御を必要としてはならない（自立形機能の実現）

メモリボードにおけるメモリの拡張のように，基本モジュールの単純な追加だけで対応可能なモジュールの自立形機能を実現することにより，モジュール制御における回路やソフトウェアの変更といった制約から解放される（参照例：4 章）．

提案する設計法を楕円曲線暗号アクセラレータに適用し，その有効性を検証した．情報セキュリティ分野で短い鍵長で強い暗号強度を有することから楕円曲線暗号が注目されてきている[10)]．しかし，公開鍵暗号は多くの多項式を多用することから処理速度の高速化が求められている．本方式を用いて，完全ハードウェア化にもかかわらず，鍵長に柔軟に対応可能なアーキテクチャを持った暗号処理システムが，容易に設計でき，その実動作が確認できたことを述べる．

2. 自立形 IP コア

自立形 IP コアは複数の同一基本モジュールからなり，モジュール単体も自立して動作可能で，同一のモジュールのカスケード接続により任意の演算精度へ拡張が可能である．要求精度に必要なモジュール数やクロック周波数およびクロック数などは，IP コア内の各モジュール機構内で自動的に決定され，外部からの制御をまったく必要としない（完全自立）．**図 1** に 4 つの自立形 IP コアを組み合わせた集積化の例を示す．ここで，各々の自立形演算機能（SSBOF）は単体でも機能する IP コアとして設計されており，共通のシステムバスに接続させて利用する．中規模の自立形 IP コアは内部にローカルバスを持ち，基本となる自立形 IP コアを複数接続した構成となる．たとえば，図 1 の SSBOF-II のように，中規模の自立形 IP コア内部で使用した自立形モジュールを直接システムバスに接続し，独立した IP コアとして用いることができる．こ

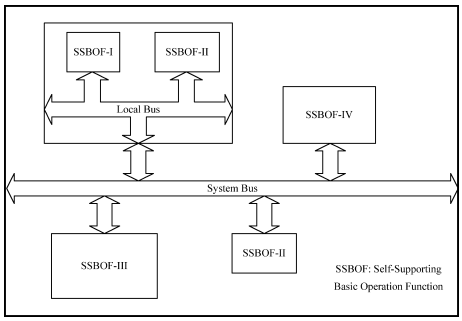


図 1 自立形基本演算機能を組み合わせて高度な演算機能を実現
Fig. 1 Realization of advanced operations combined self-supporting IP cores.

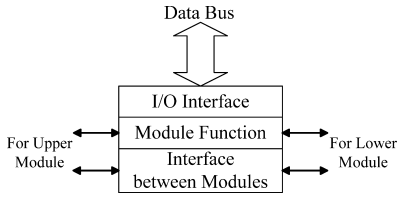


図 2 基本自立形 IP コアの構成
Fig. 2 Structure of a basic self-supporting IP core.

のため，各演算機能を持った自立形回路モジュールを組み合わせて，より高度な演算を行うことができ，再利用性に優れ，任意鍵長対応可能な楕円曲線暗号処理システムへの適用も比較的容易に構築できる．

2.1 自立形回路モジュールの構成

図 2 に基本自立形回路モジュールの構成を示す．その構成は，1) I/O インタフェース，2) モジュール機能本体，3) モジュール間インタフェースからなる．モジュール機能本体に，たとえば加減乗除算や比較などの機能を持たせ，必要な処理を自立的に判断させる．さらに，IP コア間のインタフェースの標準化を考慮した通信フォーマット用 I/O インタフェースを有する．自立して動作を実行するためには伝搬信号が必要であり，モジュール間インタフェースによってそのやりとりを行う．

図 3 に基本自立形モジュールを複数個カスケード接続して IP コアとした例を示す．IP コア内部では，そのときに必要であると判断した基本モジュールをあらかじめ用意する．このとき，自立形モジュール全体をカスケード接続することで任意処理精度となり，拡張性に優れる．さらに，処理精度に対する要求が高まった場合は図 3 の外部拡張に内部で使用しているのと同じ基本モジュールを追加すればよい．また，要求処理精度が大きければ大きいほど，モジュール全体の遅延が増大するが，自立形モジュールでは内部で適切なクロック周波数を設定することができる．このようにモジュール機構内で必要なクロックを決定するため，演算精度が小さい場合でも必要最小限のクロックで高速に処理することができる．使用するモジュールも要求処理精度によって選択し，それ以外の未使用モジュールはスタンバイモードとすることによって消費電力を抑える．

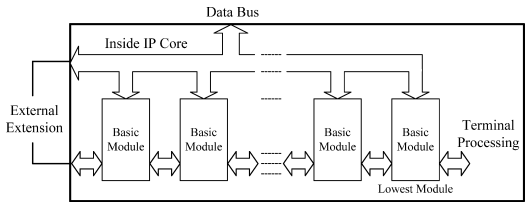


図 3 同一基本モジュールを複数用意した IP コア
Fig. 3 An IP core with several identical basic module.

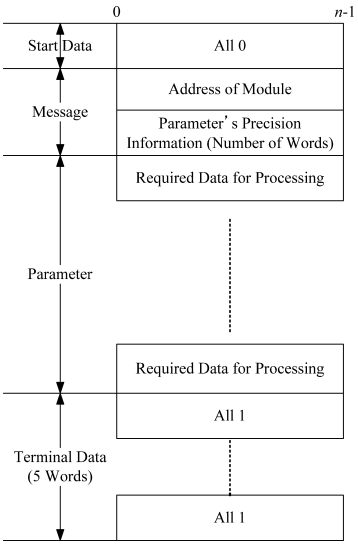


図 4 1 フレームの自立形 IP コア間通信フォーマット
Fig. 4 A frame of communication format between self-supporting IP cores.

2.2 自立形 IP コア間のデータ通信方式

自立形 IP コア間のデータ通信方法はオブジェクト指向の概念であるメッセージとパラメータに限定する．これ以外の制御信号を設けることなく，必要な処理は回路内部で自立して行う．このため，使用者は IP コア内部の詳しい構造を知る必要がなく，データ通信のためにモジュール構造を一部変更する必要もない．使用者は IP コアがどのような機能を持っているかが分かれば，必要なデータを定められたフォーマットで IP コアに与えるだけで動作させることができる．

2.2.1 自立形 IP コア間の通信フォーマット

自立形 IP コア間の通信フォーマットを**図 4** に示す．データ通信開始/終了パターンを設けることで送受信

表 1 加減乗除算の制御コード
Table 1 Control code of addition, subtraction, multiplication and division.

Control Code	Addition	Subtraction	Multiplication	Division
00	augend only	minuend only	multiplicand only	dividend only
01	addend only	subtrahend only	multiplier only	divisor only
10	addend and augend	subtrahend and minuend	multiplier and multiplicand	divisor and dividend
11	addition of one word for appointment of IP core			

データの 1 フレームを判断できるフォーマットとなる．1 ワードを n ビットとした場合，最初にデータ通信開始パターン（Start Data）として 1 ワードのオール 0 を付加し，続けてメッセージとパラメータ，最後にデータ通信終了パターン（Terminal Data）として 5 ワードのオール 1 を付加する．パラメータ中にオール 1 が 4 ワード連続したデータがある場合は，必ず 1 ワードのオール 0 を挿入し，受信する場合はそのオール 0 を取り除くことによって通信終了の誤判定を防ぐ．メッセージには自立形 IP コアのアドレスと制御コードおよび処理に必要なデータ（パラメータ）の精度情報が含まれている．パラメータは最下位ワードから順に送信する．図 3 で示す複数の同一基本モジュールからなる IP コア内では，右端の最下位モジュール内レジスタから，順次，上位モジュール内レジスタにと，精度情報に従ってデータバスからワード単位にパラメータを取り込む．

アドレスに含まれている上位 2 ビットは制御コードとして使用する．ここで，加減乗除算を例に説明する☆．加減乗除算では処理に必要なデータとして，オペランドが 2 種類必要となる（たとえば加算ならば，加数と被加数が必要となる）．このため，オペランドをそれぞれ個別に入力する場合と 2 つのオペランドの両方を一度に入力する場合とを制御コードによって区別する．制御コードのフォーマットを**表 1** に示す．

加算の場合，“00” で被加数のみの入力，“01” で加数のみの入力，“10” で被加数と加数の両方を続けて入力する．“11” はアドレスをもう 1 ワード追加する場合に使用する．たとえば 1 ワードを 8 ビットとした場合，8 ビットのうち 6 ビットがアドレスとなるのでモジュールを 64 個まで指定できることになる．それ以上のアドレスが必要な場合は，もう 1 ワードアドレスを拡張すればよい．

精度情報は続けて送られるパラメータのワード数となる．精度情報もアドレスと同じように拡張できるようにする．**図 5** に 1 ワードが 8 ビットの場合のメッセージ（精度情報）のフォーマットを示す．最上位ビットは精度情報の拡張用ビットとなる．“0” であればそのワードで精度情報が終了となり，次のワードからがパラメータとなる．“1” であればもう 1 ワード精度情報を追加する．8 ビットの場合，7 ビットが実際の精度情報となるので，0 から 127 バイトまで指定することができる．必要なデータが 127 バイト以上の場合はさらに 2 バイトに拡張し，14 ビットが精度情報として有効となる．これで，128 バイトから 16,383 バイトまでのデータを入出力することができる．

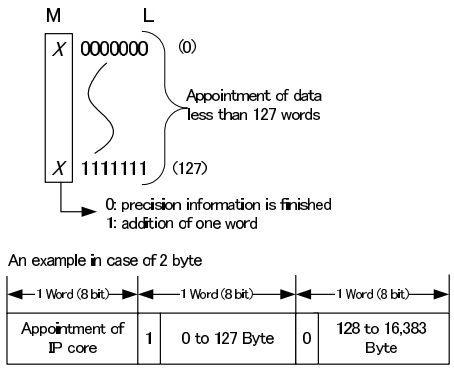


図 5 1 ワード 8 ビットとした場合のメッセージ（精度情報）のフォーマット
Fig. 5 Format of message (precision information) by a word as 8 bit.

2.2.2 自立形 IP コア間の通信例

自立形 IP コア間の通信の一例を**図 6** に示す．コア A，B，C が同じデータバス上に接続されており，コア A からコア C へデータを送信する場合について述べる．コア A は図 4 のフォーマットに従って，データをバスに流し，コア C と通信を行う．このときの通信の順序を以下に示す．

(1) コア A がデータバスにオール 0 のスタート・データを送出する．
(2) 続けて，コア A がメッセージであるアドレスを送出する．
コア B とコア C は，スタート・データがバスに流れたのを確認して，続けて送られるメッセージ（アドレス）を受け取る．アドレスが一致し

☆ 本方式は演算器に限定されるものではなく，一般的な例として四則演算器を取り上げる．

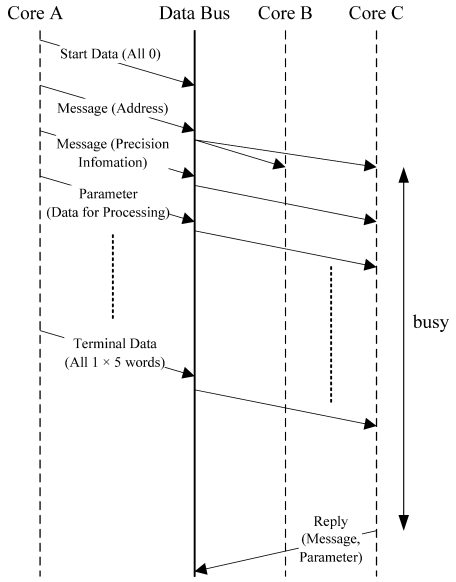


図 6 自立形 IP コア間の通信シーケンス
Fig. 6 Communication sequence between self-supporting IP cores.

たコアのみが続けて送られるデータを受け取る（ここでは，コア C のアドレスと一致）．

(3) コア C は，コア A がデータバスに送出するメッセージ（精度情報）とパラメータ（処理に必要なデータ）を順次受け取り，自立的に処理を行う（このとき，コア C はビジー状態となり，コア B はアイドル状態となっている）．

(4) コア A は，オール 1 を 5 ワードデータバスに送出することで，データの終了を知らせる．

(5) 処理に必要なデータをコア A から受け取り，処理が終了したコア C は，コア A がデータバスに送出したのと同様のフォーマットによって，データバスに返答する．

このような簡単な通信フォーマットによってデータの送受信を行い，自立形 IP コアはデータバスからのデータのやりとり以外に，制御信号を必要とせず，自立的に処理を行うことができる．

SoC における 1 つの課題としてバス調停機構があげられる．これに関してはバス調停機構を必要とせず，IP コア間のインタフェースの標準化を考慮した通信方式を筆者らはこれまで開発してきた[18),19)]．IP コアとバス間に通信制御機構（Access Control Unit: ACU）を各 IP コア間に配置し，多種多様な IP コア間のインタフェースをとる．ACU 間の連携にトークン方式を用いることで，バスの使用権を制御するもので，従来のようなバスアービタを構成するといった機構は不要となる．また，ACU はどの IP コアに対してもすべてまったく同じアーキテクチャであり，接続する IP コアに合わせた設定をする必要はない．単純に IP コアと接続するだけで，図 4 と図 6 で示す通信フォーマットとシーケンスで IP コア間の通信を行うことができる．

2.3 自立形剰余乗算コア

自立形コアで構成した剰余乗算器の例を紹介する．剰余乗算器については数多く提案されているが，本研究では基本となる乗算器と除算器のモジュールをそれぞれ自立形モジュールとして構成し，それを組み合わせて実現している．ここでは，並列形の高速性を活かし，順序形の繰返し演算を採り入れることにより，回路規模を抑えた準並列形演算器[11),12)]を使用する．自立形剰余乗算コアの基本となるモジュール構成を**図 7** に示す．図 7 は図 2 で示した基本モジュールに対応しており，ここでは同一基本モジュールが複数カスケード接続した場合の $N-1$，N，$N+1$ 番目のモジュールを示している．I/O インタフェースがデータバスと接続されており，ここで，メッセージとパラメータを解釈する．モジュール機能は準並列形乗除算器であり，伝播信号によりビットスライス化が可能となっている．モジュール間インタフェースはそれぞれの制御回路であり，入力ラッチ信号やクロック周波数などを制御する．メッセージにはモジュールのアドレスとパラメータの種類の特定および精度情報が含まれる．剰余乗算の場合，パラメータが除数，乗数，被乗数の 3 つが必要となる．本剰余乗算コアは図 7 に示すように，乗算と除算コアからなる．コア内では剰余を算出するために除数，乗数，被乗数の順にセットする．連続した剰余乗算を行う場合を考慮し，表 1 と同様にメッセージのアドレスの上位 2 ビットをオペランドの制御コードとして用い，“00” で乗数のみ，“01” で被乗数のみ，“10” で全オペランドとして設計した（他の IP コアも同様）．すべてのパラメータが揃った段階で，モジュール間インタフェースによって任意精度の乗算を行う．その結果である積のデータは内部バスによって，除算モジュールへ送られる．除算モジュールのモジュール間インタフェースによって，任意精度の除算を行い，最終的な結果として，乗算の剰余を得ることができる．この結果は I/O インタフェースへと送られ，決められたフォーマットに従って，データバスへ送出される．

準並列形乗除算器は乗算（除算）の基本モジュールのビット数が n ビットの場合，kn ビットの乗算（除算）結果を得るには k 個のモジュールのカスケードで，k クロックサイクルと $k-1$ 回の部分積（剰余）のラッチを行う．この乗除算器からなる剰余乗算モジュール

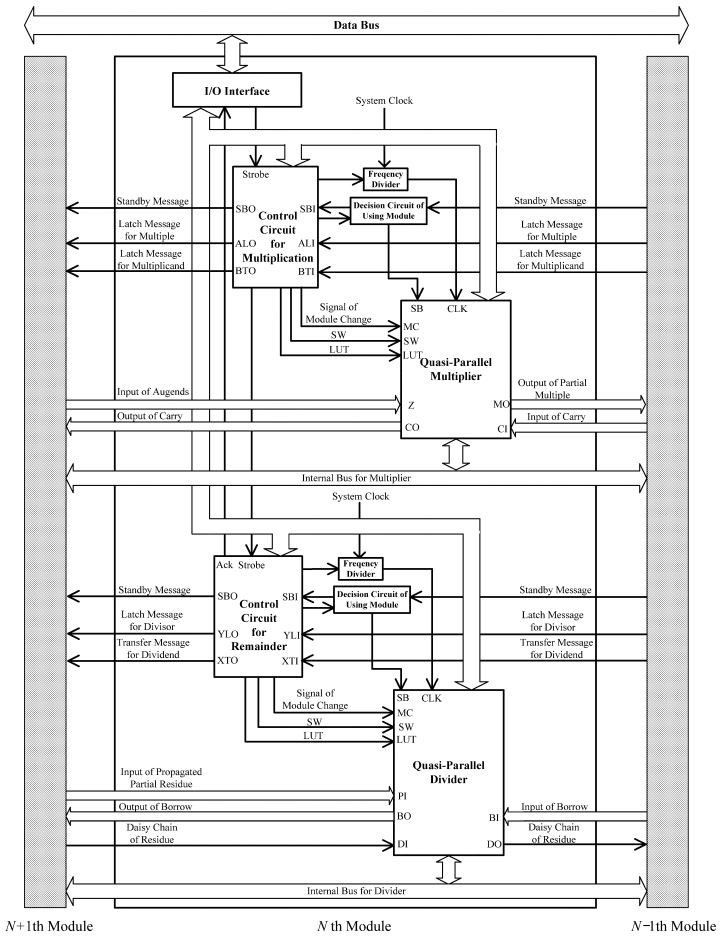


図 7 自立形基本剰余乗算モジュールの構成
Fig. 7 Structure of basic self-supporting modular multiplication module.

の 1 クロックサイクルは以下の式で示される．

$$2^a\left(\frac{n}{w}+S\right)+I+\frac{2n}{w}+2^b\left(\frac{2n}{w}+S\right) \qquad (1)$$

n は演算ビット数，w はデータバス幅，a は乗算器の分周段数，b は除算器の分周段数，I は精度情報となる．精度情報は 1 ワードで演算ビット数が入りきらなくなると 2 ワードに拡張する．このため，I は 1（$2n/w < 2^{w-1}$）または 2（$2n/w \geq 2^{w-1}$）クロックサイクルとなる．また，S はシステムクロックと分周クロックを切り替える際の同期をとるのに必要なクロックサイクルの余裕度で，4〜6 クロックサイクルを要する．

乗算器と除算器の分周段数は，そのときのプロセス固有の遅延などによって変化する．ここでの分周器は 2 のべき乗ずつクロックを分周できる．このため実際の演算サイクルに 2^a と 2^b を掛けている．

$n/w+S$ が乗算の実行サイクルとなり，その結果の出力および，除算モジュールでのレジスタの格納のためのクロックサイクルが $I+2n/w$ となる．除算モジュールでは 1 度レジスタに格納した後，最上位ワードから順に除算器に入力を行う．このときの除算の実行サイクルが $2n/w+S$ となる．

表 2 にクロックサイクルの概算結果を示す．バス幅 w が 8 ビット，16 ビットおよび 32 ビットのとき，演算ビット数 n を 128 ビット，256 ビット，512 ビットおよび 1024 ビットの場合を算出した．クロックサイクルはこれらのパラメータのほかに分周段数によっても変化するため，乗算の分周段数と除算の分周段数をそれぞれ 0 段から 1 段まで変化させた．また，乗算と除算の分周回路を別々の分周段数にさせている理由は，除算の方が最大遅延が大きいためである．

以上のように，クロック周波数とクロック数の判断はメッセージ内の精度情報によって，IP コア内で必要なモジュール数を判断し，そのモジュール数からク

表 2 自立形基本剰余乗算モジュールのクロックサイクル
Table 2 Clock cycle of basic self-supporting modular multiplication module.

Frequency Divde Stage ($a = 0$, $b = 0$)			
	Bus Width		
Bit Length [bit]	8 bit	16 bit	32 bit
128	93	53	33
256	173	93	53
512	334	173	93
1024	654	333	173
Freq. Divde Stage ($a = 0$, $b = 1$)			
	Bus Width		
Bit Length [bit]	8 bit	16 bit	32 bit
128	131	75	47
256	243	131	75
512	468	243	131
1024	916	467	243
Freq. Divde Stage ($a = 1$, $b = 1$)			
	Bus Width		
Bit Length [bit]	8 bit	16 bit	32 bit
128	153	89	57
256	281	153	89
512	538	281	153
1024	1050	537	281

ロックの分周段数を選定してクロック周波数とクロック数が決定される（自立的なクロック周波数とクロック数の判断機能）．

近年，高精度演算化が進み，1 ワード以上の演算を要求されることが多く，また，大規模な演算器を用意したとしてもワード単位に演算器にデータを与えることになる．本論文では乗除算器などの演算器も CPU の負荷を軽減することなども考慮して IP コアとした例を示したものである．1 ワードを 1 クロックでワード単位にデータをセットするパラメータ設定の時間を除き，演算部分のクロックサイクル概算結果として表 2 に示した．

今回は自立形乗算モジュールと自立形剰余演算モジュールを再利用して，自立形剰余乗算モジュールを設計した．このため，乗算モジュールからの出力を，除算モジュールで 1 度レジスタに格納してから剰余演算を行っている．最初から乗算器の出力を直接除算器の入力とする最適化を行うことで，このクロックサイクルは削減可能である．具体的には，式 (1) の中で，$I + 2n/w$ サイクル削減できる．こうすることで，表 2 に示した条件のとき，クロックサイクル数を約 15〜39%削減☆できる．よって，さらに高速化を実現することができるといえる．

☆ 分周段数が乗除算ともに 1 段，バス幅 32 ビット，128 ビットの演算を行う場合，15.8%の削減率となり，分周段数が乗除算ともに 0 段，バス幅 8 ビット，1024 ビットの演算を行う場合，39.4%の削減率となる．

以上に述べたような自立形回路モジュール機能により，再利用性や使用容易性に優れる IP コアを設計することができる．したがって，以上の特徴を持つ IP コアを使った本研究の楕円曲線暗号処理システムはそのまま自立形 IP コアとなることができる．

3. 楕円曲線暗号アクセラレータへの適用

楕円曲線暗号に必要な基本となる自立形回路モジュールを用意して，それを組み合わせて処理をすることで，より多機能な IP コアとしての楕円曲線暗号アクセラレータを構成することができる．これによりハードウェア本来の高速性を維持し，再利用性や使用容易性を実現し，さらに，バス幅は演算精度に依存しないため，スケーラブルな構造となる．**図 8** に楕円曲線暗号アクセラレータの構成を示す．基本的な構成は 2 章で述べてきたように，I/O インタフェース，モジュール機能，モジュール間インタフェースからなる．モジュール機能は各演算モジュールと制御回路からなる．楕円曲線暗号処理アクセラレータは，付録 A.1 に述べる Menezes-Vanstone 暗号系がすべて実行できる．楕円曲線暗号処理アクセラレータに備わっている機能を以下に列挙する．

- 楕円曲線上の点をすべて算出する機能（平方剰余の算出：Compute EC Points）
- 楕円曲線上の点加算・点倍算機能（EC A/D）
- 楕円スカラー倍算機能（ECSM）
- Menezes-Vanstone 暗号系による暗号化・復号処理機能（EC enc/dec）

I/O インタフェースでシステムバスからの入出力を行う．主制御回路には楕円曲線暗号に必要なレジスタファイルが含まれており，個々の制御回路と演算コアを制御する．データを格納するレジスタの本数は楕円曲線 E のパラメータである a，b，p，自分で生成した公開鍵 α，β と相手の公開鍵 α，β，暗号文（$y0$，$y1$，$y2$），復号文（$m1$，$m2$），秘密鍵，乱数用および計算用の（$c1$，$c2$）などの 7 本を加えて 26 本必要となる．演算コアは剰余乗算，除算，加算，減算，乗法の逆元コア（Inverse），楕円加算に必要な x，y 座標の計算コア（Calculate X，Y Coordinate）および式 (4) における λ の計算用に 2 倍算の分子式の計算コア（Calculate Lambda）を用意した．

図 8 において楕円曲線暗号アクセラレータの基本モジュールのシステムバスや内部バスは 8 ビットとし，除算を除く演算コアはすべて 32 ビットの演算が可能である（除算は 64 ビット）．バス幅は設計する際，システムに応じて任意に選択可能である．さらに，図 8

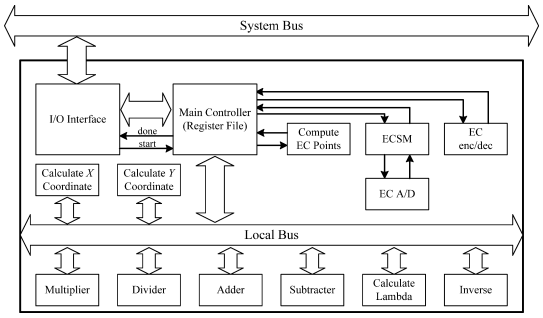


図 8 楕円曲線暗号アクセラレータの構成
Fig. 8 Structure of ECC accelerator.

を 1 つのモジュールとして図 3 のようにカスケード接続することで任意の鍵長に，スケーラブルに対応可能な IP コアとなる．それぞれの演算コアは自立形回路モジュールとなっており，制御回路部を変更することなく直接システムバスに接続することで，楕円曲線暗号以外の用途にも容易に再利用可能である．

剰余乗算は基本となるコアとして設計した乗算器と除算器モジュールを再利用し，また，楕円曲線暗号アクセラレータはその剰余乗算コア，除算，加算，減算，乗法の逆元など多くの演算コアを再利用し組み合わせて構成した．複数の IP コアを組み合わせて，より高機能な IP コアを構成するには，組み合わされるそれぞれの I/O インタフェース部分を取り外して 1 つにし，他はほとんど再利用することができる．このように楕円曲線暗号のような複雑なシステムも IP コアの再利用により，容易に設計することが可能である．

4. 実装評価

試作した自立形剰余乗算モジュールや楕円曲線暗号アクセラレータなどは Xilinx 社の FPGA Virtex-4（xc4vlx60-10，26,624 slices）をターゲットデバイスとして動作確認を行った．設計ツールに Xilinx 社 ISE Foundation 7.1i を使用し，シミュレーションには Mentor Graphics 社 ModelSim SE 6.1a を使用した．

4.1 合成結果

自立形剰余乗算モジュールおよび楕円曲線暗号アクセラレータの回路規模とターゲットデバイス Virtex-4 に配置配線後の最大動作周波数をそれぞれ**表 3** および**表 4** に示す．表 3 では 16 モジュール（32 ビット×16）までカスケード接続させた結果を示している．512 ビットまでの演算を行えるとき，使用した FPGA のスライス数が 26,622 スライスであるため，99%のリソースを消費していることが分かる．表 4 では 1 基本モジュールを 4 モジュールまでカスケード接続させた結果を示している．ここで，128 ビットの場合は FPGA のリソースが 169%となり実装することができなかった．

表 3 自立形剰余乗算モジュールの評価結果
Table 3 Evaluation result of a self-supporting modular multiple module.

Bit Length	Area [gates] (slices)	Maximum Frequency [MHz]
32	33,430 (2,093)	41.813
64	65,535 (4,110)	41.249
128	120,379 (3,412)	39.268
256	238,245 (14,742)	39.866
512	451,162 (26,622)	36.792

表 4 楕円曲線暗号アクセラレータの評価結果
Table 4 Evaluation result of a ECC accelerator.

Bit Length	Area [gates] (slices)	Maximum Frequency [MHz]
32	281,363 (16,203)	39.320
64	505,918 (26,622)	22.421
128	1,010,261 (45,039)	16.470

回路規模は等価ゲート数と FPGA のリソースであるスライス数を示した．制御回路などをすべて含んだ規模であり，カスケード接続すると，演算処理能力におよそ比例した規模となる．最大動作周波数に関しては，最も遅延の大きい演算器である剰余演算器の最大周波数で算出した．しかし，演算器は内部のクロック分周器によって制御部に独立した周波数での動作が可能である．このため実際にはさらに高速に動作☆できる．表 3 は楕円曲線暗号の主な演算機構となる剰余乗算演算の 1 回の最大動作周波数である．さらに，暗号化・復号に必要な演算を行う楕円曲線暗号アクセラレータは演算ビット長増に対して剰余乗算演算を含めた多数回の演算を行うことになる（算出する楕円曲線上の点が多くなる）ため，動作周波数は表 4 に示すように大きく低下する．なお，現在システムの開発中であるため動作確認を行うことが主目的であり，十分な最適化はなされていない．このため，回路規模や動作周波数に関してはさらに改善が望める．

4.2 比較

提案する楕円曲線暗号アクセラレータの構成要素のうち，最も処理時間がかかる剰余乗算について類似研

☆ 制御回路の最大動作周波数である 100 MHz 程度までの動作が見込める．

表 5 剰余乗算 256 ビットでの類似研究との比較結果
Table 5 Comparison result of modular multiplication (256 bit).

	Area [gates] (slices)	Freqency [MHz]	Clock Cycle	Compute Time [μs]	Platform	Remarks
This Work	238,245 (14,742)	39.86	173	4.34	Virtex-4-10	Self-Supporting 8 bit × 8 modules
Tenca, et al. [7]	5,719	192.0	558*	2.90*	ASIC 0.5 μm	8 bit PE × 16
Tenca, et al. [7]	20,893	166.6	519*	3.11*	ASIC 0.5 μm	32 bit PE × 8
Satoh, et al. [8]	3,228	363.6	2,278	6.27*	ASIC 0.13 μm	8 bit bus width
Satoh, et al. [8]	96,224	137.7	66	0.48*	ASIC 0.13 μm	64 bit bus width
Crowe, et al. [9]	(5267)	44.91		5.75	Virtex-II-6	
Örs, et al. [13]	(1548)	100.4	772*	7.69	Virtex-E-8	

*参考文献から筆者らが推定

究との比較を行った．比較対照としては，プロセッサによる方式ではゲート数を抑制できるメリットはあるが，ソフトウェアによるため処理速度の点で不利である．SoC のような超大規模集積回路ではゲート数は大きな設計要因とはならない．文献 20)，21) のソフトウェアによる方式では制御機構を内蔵しない従来方式の IP コアに比べて数十倍以上という大きな速度低下となることから，高速な IP コアとの比較とした．**表 5** にその結果を示す．文献 7)，8) ではモンゴメリ乗算において回路規模と速度のトレードオフが選択できるため，ここでは 2 種類の設定での結果と比較を行った．文献 7) ではバス幅とパイプラインで使用する PE（Processing Element）の個数を選択できる．このため，32 ビットのバス幅で 8 個の PE の場合と 8 ビットのバス幅で 16 個の PE の場合を比較した．文献 8) では 8 ビットバス幅と 64 ビットバス幅の乗算器と比較を行った．この研究がこの中では最も速い結果を示している．しかし，ASIC によって実装されているので筆者らが試作した FPGA とは速度や面積が根本的に異なる．また，文献 9)，13) でもモンゴメリ乗算を使用しているが，バス幅は演算ビット以上となる．デバイスは FPGA を使用しており，等価ゲート数は公表されていない．

本研究では，1 モジュールを 8 ビットのバス幅で 32 ビットまで演算できるように設計したため，8 モジュールカスケード接続した場合を示している．ここで式 (1) のクロックを同期するためのサイクル S を 6 クロックサイクル，分周段数は乗算器，除算器ともに 0 段としている．今回は 8 ビット幅でしか実験していないが，たとえば，32 ビット幅にするとクロックサイクルは 53 となり，類似研究と比べても最も少なくなる．また，回路規模では自立形回路モジュールの構造となるのは本研究のみであり類似研究では演算器のみの回路規模となるのでゲート数を単純には比較できない．しかし，参考のために本研究での演算器のみのゲート数は約 80 k ゲートとなることが分かっている．最適化によってどれだけ小さくできるかは今後の課題となる．

IP コア自身に自立性を持たせる機能を内蔵することにより，使用容易性および拡張性が向上し，IP コア間のインタフェースの標準化が容易になる．そのためには図 2 で示すように，モジュール機能本体に I/O インタフェースとモジュール間インタフェースを組み込み，単体でも IP コアとして機能しなければならない．自立的でない従来方式に比べてインタフェース部分がゲート数増となるが，剰余乗算モジュールの場合，従来方式では I/O インタフェース部で 1,000 ゲート，モジュール間インタフェース部で 5,700 ゲートに対して，提案する自立形では I/O インタフェース部で 2,100 ゲート，モジュール間インタフェース部は I/O インタフェース部で共通にモジュールの選定制御を行っているのでほとんど変わらない．試作では 1 モジュールを 8 ビットのバス幅で 32 ビット構成としたが，バス幅によってデータの取り込む回数とレジスタ長が変化する．したがって，バス幅が同じなら 32 ビット構成以外（たとえば 64 ビットや 128 ビット構成）でもインタフェースの制御部分の割合は変わらない．外部から制御する回路やプロセッサで制御する従来方式では，モジュールの制御法を理解しなければならないが，自立性を持った IP コアにより，モジュールの内部構造を理解することなく容易に使用することができる．数千万ゲート以上の規模となる SoC においては，自立形とすることによるゲート数のオーバヘッドよりも SoC のような超高集積回路において課題となっている IP コアの再利用性，拡張性，使用容易性などを考慮した生産性の向上が期待できることの意義が大きい．

5. ま と め

次世代の高集積回路においては多種多様な IP コア間インタフェースの標準化，再利用性，使用容易性お

よび拡張性などが求められる．そこで，処理精度に対してスケーラブルなアーキテクチャを持った完全自立形 IP コアの設計法を提案し，その基本モジュールの構成法とモジュール間インタフェースの標準化の容易性を考慮した通信方式について述べ，以上に述べた諸問題がいかに解決できるかを示した．提案する設計法の有効性を検証するため，多くの多項式からなりモジュール化が困難であった楕円曲線暗号アクセラレータに適用し，可変長鍵に対応可能な IP コアを FPGA で試作した．

完全自立形 IP コアは他に存在せず，類似研究と回路構成が異なるため，単純には比較できないが，参考のため回路規模と演算時間などについて比較検討を行った．試作した楕円曲線暗号アクセラレータは FPGA の回路規模の関係でバス幅を 8 ビットで行い，多精度に対応した場合の諸特性とバス幅を大きくした場合の検討を行った．現段階では十分な回路の最適化は行っていないため，使用ゲート数に関しては改善が必要であるが，処理速度は 32 ビットのバス幅で，類似研究中，最も高速化が期待できる．その主な要因としては完全ハードウェアによるモジュール化にあるが，処理精度に対してソフトウェアのように柔軟に対応できるアーキテクチャを実現したところによる．その実現手段が 1 章であげた 5 項目による「完全自立形スケーラブル IP コアの設計法」である．複数の基本モジュールからなる IP コアは処理に必要なモジュール数を判断し（拡張性），適切なクロック周波数とクロック数で効率的な処理を行い結果を返す．IP コア間は簡素なメッセージ通信であるため使用容易性に優れ，モジュールの再利用により単純なカスケード接続だけで楕円曲線暗号アクセラレータの任意の鍵長にも容易に対応できるという結果が得られた．

近い将来，構成要素がヘテロジニアスな SoC の実現が予測されている．SoC 内の多種多様な IP コア間の通信では，従来のバス方式では通信効率が低下することからチップ内でネットワークを構築する NoC（Network on Chip）に関するいろいろな研究報告がなされてきている[14]．筆者らが提案した完全自立形 IP コア間を NoC によって効率的に通信する機構についても，研究を進めている．

謝辞 本研究は東京電機大学総合研究所研究 Q04J-12 の研究費を受けて行っているものである．

参 考 文 献

1) Bimbaum, M. and Sachs, H.: How VSIA Answers the SOC Dilemma, *IEEE Computer*, Vol.32, No.6, pp.42–50 (1999).
2) 中村次男，笠原 宏：オブジェクト指向手法をハードウェア設計に導入する提案と VSI 向きコアのモジュール化，電学論（C），Vol.121-C, No.3, pp.564–573 (2001).
3) 中村次男，鈴川敦之，冬爪成人，笠原 宏，田中照夫：超高集積 LSI 時代に向けたハードウェア設計法，電学論（C），Vol.124, No.4, pp.995–1003 (2004).
4) 佐藤正幸，中村次男，冬爪成人，笠原 宏，畠中浩行，田中照夫：自立形回路モジュールの実現法，電子情報通信学会第 4 回リコンフィギャラブルシステム研究会論文集，pp.79–86 (2004).
5) Sato, M., Nakamura, T., Hatakenaka, H., Hayakawa, M., Fuyutsume, N. and Kasahara, H.: Hardware Implementation of Elliptic Curve Cryptosystem Adaptive to Infinite Key Length, 電子情報通信学会ソサイエティ大会講演論文集, C-12-10, p.90 (2005).
6) 畠中浩行，中村次男，佐藤正幸，早川雅文，冬爪成人，笠原 宏，田中照夫：SoC 内 IP コアの設計–楕円曲線暗号システムモジュール化への適用例–，電気学会電子・情報・システム部門講演論文集，GS2-1, pp.741–745 (2005).
7) Tenca, A. and Koç, Ç.K.: A Scalable Architecture for Modular Multiplication Based on Montgomery's Algorithm, *IEEE Trans. Computers*, Vol.52, No.9, pp.1215–1221 (2003).
8) Satoh, A. and Takano, K.: A Scalable Dual-Field Elliptic Curve Cryptographic Processor, *IEEE Trans. on Comput.*, Vol.52, No.11, pp.449–460 (2003).
9) Crowe, F., Daly, A. and Marnane, W.: A Scalable Dual Mode Arithmetic Unit for Public Key Cryptosystems, *Proc. IEEE Int. Conf. on Information Technology: Coding and Computing*, pp.568–573 (2005).
10) 辻井重夫，趙 晋輝：楕円暗号へのガイダンス，信学論（A），Vol.J82-A, No.8, pp.1200–1211 (1999).
11) 中村次男，笠原 宏：超高精度整数乗算器の高速化とモジュール化，電学論（C），Vol.121-C, No.7, pp.1212–1219 (2001).
12) 中村次男，笠原 宏：任意精度向き準並列形高速除算機構，電学論（C），Vol.120-C, No.1, pp.158–167 (2000).
13) Örs, S.B., Batina, L., Preneel, B. and Vandewalle, J.: Hardware Implementation of a Montgomery Modular Multiplier in a Systolic Array, *Proc. 10th Reconfigurable Architectures Workshop* (2003).
14) Benini, L. and Micheli, G.D.: Networks on Chips: A New SoC Paradigm, *IEEE Computer*, Vol.35, No.1, pp.70–77 (2002).

15) IEEE P1363, Standard Specifications for Public-Key Cryptography (2000).
16) Gordon, D.: A Survey of Fast Exponentiation Methods, *Journal of Algorithms*, Vol.27, pp.129–146 (1998).
17) Blake, I.F., Seroussi, G. and Smart, N.P.: *Elliptic Curve in Cryptography*, London Mathematical Society Lecture Note Series 265, Cambridge University Press (1999). 鈴木治朗(訳)：楕円曲線暗号，ピアソン・エデュケーション (2001).
18) 早川雅文，中村次男，佐藤正幸，畠中浩行，冬爪成人，笠原　宏，田中照夫：SoC 内 IP コア間の通信方式，電気学会電子・情報・システム部門大会，GS2-2 (2005).
19) 古屋憲吾，中村次男，冬爪成人，笠原　宏：オブジェクト指向技術を導入した IP コアの設計とその連携方式，電気関係学会関西支部連大，G10-15 (2003).
20) Hasegawa, T., Nakajima, J. and Matsui, M.: A Small and Fast Software Implementation of Elliptic Curve Cryptosystems over GF(p) on a 16-Bit Microcomputer, *IEICE Trans. Fundamentals*, Vol.E82-A, No.1, pp.98–106 (1999).
21) Miyaji, A., Ono, T. and Cohen, H.: Efficient elliptic curve exponentiation, *Proc. 1st International conference on Infomation and Communication Security*, Vol.1334 of Lecture Notes In Computer Science, pp.282–291, Springer-Verlag (1997).

付　　録

A.1 楕円曲線暗号

ここでは 3 章以降で実装する際に用いた楕円曲線の定義と楕円曲線上の計算方法について述べる[15]．

A.1.1 暗号に利用される楕円曲線の概要

素数 $p > 3$ であるガロア体 $GF(p)$ 上の Weierstrass 標準形楕円曲線 E は，

$$E : y^2 \equiv x^3 + ax + b \pmod{p} \tag{2}$$

で示される．ここで，$4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ となる a，b を満たさなければならない．このとき無限遠点（point at infinity）と呼ばれる特別な点 $\mathcal{O}$ を一緒に考える．楕円曲線上の点 $P = (x_1, y_1)$，$Q = (x_2, y_2)$ があるとき，$x_2 = x_1$ かつ，$y_2 = -y_1$ ならば，$P + Q = \mathcal{O}$ とするが，そうでない場合は $P + Q = (x_3, y_3)$ とする．点 P と Q を通る直線の傾きを λ としたとき (x_3, y_3) は，

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned} \tag{3}$$

となり，傾き λ は，

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \quad (P \neq Q) \tag{4}$$

$$\lambda = \frac{3{x_1}^2 + a}{2y_1} \quad (P = Q) \tag{5}$$

となる．

楕円曲線暗号の安全性はスカラー倍算 $Q = kP$ が楕円曲線上の離散対数問題に基づいていることによる．これは，点 P を k 倍して点 Q を求める．スカラー倍算を求めるには様々な方法[16),17)] があるが，本研究では最も基本的なアルゴリズムである 2 進展開法を用いる．

A.1.2 Menezes-Vanstone 暗号系

本研究での楕円曲線暗号は平文や暗号文が楕円曲線上に限定されない Menezes-Vanstone 暗号系を使用する楕円曲線とする．

そのアルゴリズムはまず，楕円曲線上の全点 (x, y) を求める計算を行う（平方剰余の算出）．平方剰余の算出により求めた点の中から任意の 1 点を選定し，原始元 α とする．この原始元 α はベースポイントとして公表する．次に，自由に選出した整数 γ を秘密鍵とする．ベースポイント α と γ の倍数を求め，その結果である $\beta = \gamma\alpha$ の β を公開鍵として公表する．

暗号文を作成するには，通信したい相手の公開鍵 α，β，秘密の乱数 k および楕円曲線上に限定されない平文 M (m_1, m_2) を用意する．平文に対する暗号文を y (y_0, y_1, y_2) とすると，

$$\begin{aligned} (c_1, c_2) &= k\beta \\ y_0 &= k\alpha \\ y_1 &= c_1 m_1 \mod p \\ y_2 &= c_2 m_2 \mod p \end{aligned} \tag{6}$$

で表される．(c_1, c_2) は乱数 k と公開鍵である β によって生成し，これを用いて暗号化を行う．

次に，暗号文を平文に復号する式を以下に示す．

$$\begin{aligned} (c_1, c_2) &= \gamma y_0 \\ m_1 &= y_1 c_1^{-1} \mod p \\ m_2 &= y_2 c_2^{-1} \mod p \end{aligned} \tag{7}$$

暗号文を元の平文に復号するには，暗号文として公表されている y_0 と秘密鍵 γ を用いて，暗号文を作成したときと同じ (c_1, c_2) を生成する．これの乗法の逆元と暗号文の積をとることで元の平文を得る．

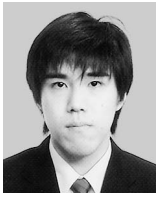
(平成 18 年 6 月 13 日受付)
(平成 18 年 11 月 2 日採録)

佐藤　正幸

2004 年 3 月東京電機大学工学部電気工学科卒業．2006 年 3 月同大学大学院情報環境学研究科修士課程修了．同年 4 月日立超 LSI システムズ入社．在学中，IP コアの設計法（オブジェクト指向ハードウェア設計法），高速暗号化に関する研究に従事，暗号処理回路，チップ内ネットワーク，高速・高精度演算器に興味を持つ．

中村　次男

1971 年 3 月東京電機大学工学部電気工学科卒業．同年 4 月日本電気精器（株）入社．1990 年 3 月東京電機大学大学院修士課程修了．現在，国際短期大学情報ネットワーク学科助教授．博士（工学）．ディジタル演算器，高速暗号化，集積回路の盗用防止機構，オブジェクト指向ハードウェア設計法に関する研究に従事．著書『ディジタル回路の基礎』（日本理工出版会），『ディジタル回路設計法—ワンチップ化の実例集』（日本理工出版会），『ディジタル回路入門』（日本理工出版会），『電子回路（2）ディジタル編』（コロナ社），『電気・電子—なぜなぜおもしろ読本』（山海堂）．電気学会，IEEE 各会員．

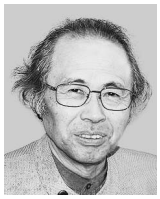


畠中　浩行

2005 年 3 月東京電機大学工学部電気工学科卒業．現在，同大学大学院情報環境学研究科修士課程在学中．IP コアの設計法（オブジェクト指向ハードウェア設計法），高速暗号化に関する研究に従事．

冬爪　成人（正会員）

1995 年 3 月東京電機大学工学部電気工学科卒業．2000 年 3 月同大学大学院博士課程修了．同大学工学部電気工学科助手，同大学情報環境学部情報環境工学科助手を経て，現在，同学部講師．博士（工学）．コンピュータネットワークに関する研究に従事．電気学会，電子情報通信学会各会員．



笠原　　宏（正会員）

1970 年 3 月東京電機大学大学院工学研究科博士課程満期退学．同大学助手，講師，助教授を経て工学部電気工学科教授．現在，同大学情報環境学部情報環境工学科教授．パワーエレクトロニクス，計算機システム，制御用分散処理システム，オブジェクト指向システム設計法，インタラクティブにとらえた脈診情報処理方式，高速暗号化に関する研究に従事．電気学会，電子情報通信学会，IEEE 各会員．

田中　照夫

1972 年 3 月東京電機大学大学院工学研究科博士課程満期退学．1969 年同大学助手，専任講師，助教授を経て，現在，同大学工学部電気工学科教授．主としてパワーエレクトロニクスに関する研究に従事．電気学会，IEEE 各会員．