超高集積 LSI 時代に向けたハードウェア設計法

正 員 中村 次男^{*}

非会員 鈴川 敦之^{**}

正 員 冬瓜 成人^{***}

正 員 笠原 宏^{***}

正 員 田中 照夫^{***}

A Hardware Design Technique for Next Generation

Tsugio Nakamura^{*}, Member, Nobuyuki Suzukawa^{**}, Non-member, Narito Fuyutsume^{***}, Member,
Hiroshi Kasahara^{***}, Member, Teruo Tanaka^{***}, Member

As for the next generation of ultra large-scale integrated circuit called system LSI and system in package, the difficulty in designing many large circuit blocks has been pointed out. Therefore, it is being searched for new ways to cope with designing them.

In this paper, we propose a method of hardware modularization, each of which has the function of judging necessary operation without any control from the outside, and without any limitation of the operational precision, i.e. an autonomous module. We also discuss how to standardize the module interface for ease-of-use.

キーワード：システム LSI, ハードウェアモジュール, IP コアの再利用, オブジェクト指向技術

Keywords : system LSI, hardware module, reuse of intellectual property core, object-oriented technique.

1. まえがき

集積回路技術の向上は著しく、数千万ゲート規模の超大規模集積回路(System LSI)が 1 チップ上に実現できるようになり、2010 年には 4,081M Tr (メガ・トランジスタ) に達するであろうと予測されている⁽¹⁾。これにより、従来のプリント基板上にシステムを構築していたシステムボードという形態から、1 チップ上にシステムを構築する SOC(System On Chip)や 1 パッケージ上にシステムを構築する SIP(System In Package)への移行が進み、現在では各種コンピュータ、FAX、携帯電話、携帯情報機器(PDA)をはじめとする多くの電化製品に用いられている⁽²⁾。しかし、数千万ゲート規模の集積化には様々な問題が指摘されている。半導体技術向上を目的とする米国政府と半導体メーカ

による官民プロジェクト(SEMATEC)の予測では「LSI の集積度は年率 58%で向上し続けるが、LSI 設計者の設計能力は年率 21%しか向上せず、それに伴う設計の困難さも年平均 58%で増加する」と言われている⁽³⁾⁽⁴⁾。この設計工数が増加する問題をコンピュータや EDA ツールの性能向上で改善することは難しい。それどころか、新しい EDA ツールに対応する各種専門知識の開発のために、膨大な開発リソースが必要となり、設計工数は増大する一方である⁽⁴⁾。この問題に対処する方法として、設計手法の改革と成果物の再利用法が議論されているが、その中でも特に、IP コアとして出まわっているものの再利用が有効な手段であると言われている。

例えば、四則演算器、DSP、MPU やメモリなど、すでに設計検証済みの IP コアを新規システム LSI の設計時に、そのまま利用できれば IP コア化されている回路部分の再設計が不要となり、設計工数は大幅に削減できる⁽⁵⁾。しかし、多様な開発環境下で設計された IP コアの再利用には各コア間のインタフェースや使用容易性などの大きな難問が存在する。

筆者らは IP コアの再利用を容易にするための新しいハードウェアモジュール (以後、モジュールという) 設計法の提案⁽⁶⁾⁽⁷⁾を行ってきた。モジュールとは以下の 2 項目を満足する。

^{*} 国際短期大学情報ネットワーク学科
〒165-0022 東京都中野区江古田 4-15-1
Department of Information Network, Kokusai Junior College
Egata 4-15-1, Nakano, Tokyo 165-0022

^{**} ローム株式会社
〒222-8575 横浜市港北区新横浜 2-4-8
ROHM Co.,LTD
Shin-Yokohama 2-4-8, Minato-Kita, Yokohama 222-8575

^{***} 東京電機大学情報環境学部 & 工学部
〒270-1382 千葉県印西市武西学園台 2-1200
School of Information Environment & Faculty of
Engineering, Tokyo Denki University
Muzai Gakuendai 2-1200, Inzai, Chiba 270-1382

1. モジュール間のデータ転送にはオブジェクト指向技術を導入し、メッセージの後にパラメータを付加した形式のみに限定する。
2. パラメータに指定した処理精度に自ら適応できる自立形のモジュール化

以上のようなモジュールの有効性を先の論文⁽⁶⁾では検証した。

本論文では再利用可能な IP コアを実現するために、ソフトウェアの再利用性や拡張性などオブジェクト指向技術がいかにこのハードウェア設計法に活かすことができるかを具体例を挙げてその手法を述べる。

モジュールの設計にオブジェクト指向的手法を用いるというのはモジュール内部を設計するために、現在、広く普及している様々な回路(機能)記述法(VHDL/ VerilogHDL, Spec C および System C など)とは根本的に異なる。本論文は回路の記述法についての提案ではなく、処理データの規模に左右されないソフトウェアのような柔軟性を持ったハードウェア回路モジュール、以下にこれをオブジェクトと呼び、その設計法について、

1. オブジェクト間のインタフェース
2. 再利用可能で拡張容易な自立形オブジェクト

の具体例について述べる。これにより、システム LSI の設計者は IP コアの内部を知らなくても容易に IP コアを使用することができるようになり、オブジェクト間のインタフェースの問題が解決できる。

2. 提案するハードウェア設計法の目的と意義

SOC での IP コアの再利用および SIP でのシステムモジュール化の傾向が強まっており、モジュール設計技術の開発が求められている⁽¹³⁾。

IP コアを再利用して新規に回路を設計する場合、IP コアの標準化が不可欠である。コアの標準化には製造プロセスやインタフェースなど大きな技術的問題が存在する。各ベンダ間でそれぞれ特定の技術を持って競合している現状を考慮すると、その標準化は大変な難問であり、更に、多くの IP コアを使いこなすには使用容易性の問題も考慮されなければならない。

IP コアを複数再利用して新規にシステム LSI を設計しようとした場合、チップ上に用いられる全ての IP コアに関する仕様や動作を設計者は全てを、また LSI の使用者は全コアの動作を全て理解している必要がある。もし、再利用する IP コアがひとつや二つならば理解することも可能であるが、多数の異なった開発元から提供された場合、その難度は極めて高くなることは容易に推察できる。

そこで、ソフトウェアの再利用性や拡張性などのメリットから普及してきているオブジェクト指向技術をハードウェアモジュール設計に適用する。モジュールをオブジェクト化することにより、処理データの規模に左右されないソフトウェアのような柔軟性を持ったコアが可能となるだけでなく、そのコアの処理を指示するメッセージと処理に

必要なデータをパラメータとしてバス上に送出することで結果を得ることができる。これにより、インタフェースは入出力/制御メッセージだけとなり、また、モジュールの内部動作を LSI の設計者や使用者は知る必要がなくなる。

モジュールの制御はひとつの制御部から全モジュールを制御する、いわゆる中央制御形が一般的に行われている。この方法ではモジュール拡張に際し、制御回路を変更する必要があるため、処理精度に影響されない柔軟性を持ったモジュールを設計することは困難である。

そこで本設計法ではモジュール自体にモジュール内の処理に必要なクロック数、処理精度判定および I/O 制御などの各種制御を判断する機能を持たせ、モジュール自身がデータを処理するために必要とされるモジュール数と全クロック数など実行に必要な項目を判断できる機能を持たせる。つまり、モジュールの持つ機能の実行は外部からの制御を受けることなく、モジュール内部で処理が行われ、結果を返すという、正に自立形のオブジェクトの実現である。こうすることで、処理精度に影響されない柔軟性を持ったモジュールを設計することが可能となる。データの処理に必要なモジュール数は各モジュール内で判断するため、モジュールを拡張するときは従来方式のように制御回路の変更は必要なく、単にカスケード接続するだけである。

図 1(a)に本設計法による複数のモジュールから成るオブジェクトの構成を示す。図に示すように、オブジェクト化されたモジュールのインタフェースは「入出力/制御メッセージ」だけである。本設計法におけるモジュールは全てこのインタフェースを使用し、メッセージ通信を行う。

本方式ではモジュールのインタフェースにソフトウェアにおけるオブジェクトと同様に、メッセージとパラメータ(パラメータの最後の 1 ワードは全て“1”とし、バス上にデータが無いことを示す)というフォーマット(図 1(b))を用いる。このフォーマットは全てのモジュールに共通で

図 1 オブジェクトの概要
Fig. 1. Outline of the object.

あり、モジュールの機能ごとにシステム LSI 内のアドレスコードのようなビットパターンを各 IP コアの実装時に設定する。LSI の使用者はコード変換表を介して、使用するモジュールに対応するメッセージとパラメータをモジュールに送るだけでモジュールの使用が可能となり、処理結果を得ることができる。

3. 基本オブジェクトの設計

モジュールとして構成される最小単位である基本オブジェクトを組み込んで、より規模の大きなモジュール (IP コア) を構成できるというのが提案する設計法でもある。四則演算は情報処理における基本演算であることから、加算と除算のオブジェクト化を、更に、四則演算以外の例としてソートオブジェクトの設計例を取り挙げる。

〈3・1〉 **加算オブジェクト** 四則演算の基本である加算器のオブジェクト化を取り挙げる。1 モジュールの仕様を以下に、ブロック図を図 2 に示す。

メッセージフォーマット：図 2(a) 参照。

演算精度：k ビット (k ビット並列加算器；モジュールのカスケード接続により k の整数倍に拡張可)。

クロック周波数：演算にクロックを用いる場合は精度情報から最大遅延を判断し、システムクロックを分周して最適なクロック周波数を選定し、出力用クロックに用いる。この場合は並列加算器であるため加算データの入力用にはシステムクロックを用いる。

モジュール間メッセージ入出力：精度情報と比較しなが

ら下位モジュールから上位モジュールへと加算データを取り込むデータラッチメッセージおよび加算中の桁上げ伝搬。

オブジェクト選択コードとパラメータの精度情報から成るメッセージはそれぞれオブジェクトとモジュール数の最大数により、ビット数が決まる。モジュールの 1 ワードを仮に 8 ビットとする。上位 4 ビットをオブジェクト選択コード、下位 4 ビットをパラメータの精度とすれば、システム LSI 内オブジェクトは最大 16 種類、加算精度は $k \times 16$ ビットまで可変長加算が可能となる。それで不足する場合はオブジェクト選択コードとパラメータの精度をそれぞれ 1 ワードとするかメッセージ内に拡張用ビットを設けることで可能である。

〈3・2〉 **除算オブジェクト** 最も単純で説明容易なことから加算オブジェクト 〈3・1〉 を例に、本設計法を説明した。次に除算オブジェクトを取り上げる。除算は乗算と異なり、除算データを分割し、別々に除算した結果から拡張した解を得るといった除算の分散処理は不可能である。しかし、本設計法によるモジュール化のカスケード接続により、可変長除算が可能になる。筆者らが先に開発した準並列形除算器⁽⁸⁾のオブジェクト化について述べる。

準並列形除算器は並列除算を規定回数繰り返すことにより、除算結果を得るもので、並列形的高速性と順序回路形の繰り返し演算を取り入れ、ゲート数と入出力数を抑制した機構である。1 モジュールの仕様を以下に、ブロック図を図 3 に示す。

メッセージフォーマット：図 3 (a) 参照。

図 2 加算オブジェクトのブロック図

Fig. 2. Block diagram of adder object.

除算精度: k ビット (m 個のモジュールをカスケード接続により, $k \times m$ ビット精度の除算に拡張可)

クロック周波数: k ビット並列減算器を k 段 1 ビット右にシフトした構成で, 1 モジュールの最長伝搬遅延が求まる。精度情報からモジュールのカスケード数を判断し, 総伝搬遅延より, システムクロックの分周を決め, 演算クロックとする。

部分剰余伝搬入力制御信号: 被除数の上位 k ビットづつ, 最下位モジュールの部分剰余伝搬入力に内部バスを介して転送, または, 商を各モジュールに格納する際の切り替え信号。

部分剰余のラッチ信号: m モジュールのカスケード接続で $m-1$ 回の繰り返し演算を行う際の部分剰余ラッチ信号。

被除数制御信号: モジュール内被除数の MSB は最下位モジュールだけ固定で, 他のモジュールは下位桁からの部分剰余とするための信号。

被減数の制御信号: 部分剰余を初め “0” とし, 1 回目の繰り返し演算以降の部分剰余を被減数として取り込むための信号。

モジュール間メッセージ;

除算データラッチ: 精度情報と比較しながら下位モジュールから上位モジュールへ除算データを取り込むデータラッチメッセージ (結果の出力用も兼ねる)。

被除数データの転送信号: 数モジュールをカスケード接続した場合, 各モジュールに取り込まれた被除数を最上位モジュールから順に, 最下位モジュールに転送するためのメッセージ (1 モジュール

(a) Message format

(b) Block diagram

図 3 準並列形方式の除算オブジェクトの概要

Fig. 3. Outline of divider object with quasi-parallel architecture.

分の演算終了後、次の下位モジュールが k ビットの被除数データを転送)。

桁借り伝搬：下位モジュールからの桁借り入力と上位モジュールへの桁借り出力がそれぞれ k ビット。

減算抑止信号：減算結果を更新して出力するかどうかを判断する k ビットの入力。

部分剰余伝搬入力： k ビットの被除数入力。最下位モジュールだけが被除数を k ビットずつ最上位モジュールから順に入力。2 段目以上のモジュールは

下位ジュールからの部分剰余を入力。

部分剰余出力： k ビットの部分剰余出力（モジュール内では $2k-1$ ビットの部分剰余が得られるが、その上位 k ビット）。

部分剰余のフィードバック信号： k ビットの部分剰余出力中、最下位ビットを入力。

商のラッチメッセージ：商は最上位モジュールから上位 k ビットずつ得られる。順次 k ビットの商を内部バスを介して下位モジュールに取り込む。

(a) Message format

(b) Block diagram

図4 ソートオブジェクトの概要

Fig. 4. Outline of sort object.

除算結果の精度判定デージーチェーン：結果の商と剰余はカスケード接続された各モジュール内に k ビットずつ格納されている。自モジュール内の結果が“0”で、かつ上位モジュールも“0”のとき、“0”を下位モジュールに伝搬するデージーチェーン。これにより結果の精度を判定し、不要な“0”の出力を避ける。

モジュール内部バス：被除数データと商の転送用 k ビット内部バス。

〈3・3〉 ソートオブジェクト 四則演算以外の適用例としてデータの並び替え機能を有するソートを取り挙げる。高速で効率的なアルゴリズムはいろいろ開発されているが、設計法の説明を優先し、単純でソートの基本となるバブルソートとする。1モジュールの仕様を以下に、ブロック図を図4に示す。また、昇順か降順かは結果のデータを最上位から最下位に向けて参照するかその逆かということなので、昇順あるいは降順の固定とする。

メッセージフォーマット：1ワードを k ビットとした場合の例を図4(a)に示す。

比較精度： k ビット $\times r$ (1モジュールで k ビット単位に r 個のレジスタ)

クロック周波数：1モジュール kr ビット分の遅延と精度情報からモジュールのカスケード数を判断し、最大遅延にあった最適なクロック周波数を、システムクロックを分周して選定する。

レジスタ：ソートデータ用レジスタ。それぞれ k ビット $\times r$ 個の格納用レジスタと比較用レジスタ A と B。

比較器： $k \times r$ ビット長の比較器

モジュール間メッセージ

ソートデータラッチ：精度情報とデータ数を比較しながら下位モジュールから上位モジュールへ格納用レジスタにソートデータを取り込む（結果の出力用も兼ねる）。

比較結果のデージーチェーン：上位モジュールの比較結果を下位モジュールに伝搬する。

4. 複数の基本オブジェクトから成る IP コアの構成

提案する設計法は処理データの規模に制約を受けないソ

フトウェアのような柔軟性を持ったモジュールから成る IP コアとその IP コアの再利用容易性の実現を目指したものである。先ず、3.で設計した加算、除算およびソートの基本オブジェクトから成るそれぞれの IP コアの構成について述べ、次にそれぞれの IP コアの再利用例について説明する。

〈4・1〉 加算コア 1モジュールで k ビットの加算〈3・1〉を行い、図5のように単純な同モジュールの複数個のカスケード接続により、 k の整数倍の加算が可能になる。図2(a)で示したメッセージフォーマットで、バスに送出されたメッセージ中のオブジェクト選択コードから、加算の指定が判断されると加算コアは続けて送られてくる被加数と加数のパラメータを精度情報に従って順次下位モジュールから上位モジュールへと k ビット単位に取り込む。どのモジュールまで取り込んだかは精度情報から判断できるので、加算結果はメッセージのオブジェクト選択コードを応答コードに替えて（全ビットを“0”とする）、精度情報の次に付加して最下位モジュール（あるいは最上位モジュール）から順次バス上に送出する。

〈4・2〉 除算コア 〈3・2〉の1モジュール k ビットの除算を実行する除算モジュールを図6で示すようにカスケード接続することにより、 k ビットの整数倍の除算に対応できる。図3(a)のメッセージフォーマットにより、先ずオブジェクト選択のメッセージがバス上に送出されると存在する複数のコアは自分宛てであるかを判断する。この場合、除算の指定であると、続けて送られてくるパラメータの精度情報より、除算コアは被除数と除数を最下位モジュールから順に上位モジュールへと取り込む。どのモジュールまで取り込むのか、また必要な演算クロック数などは精度情報から判断し、除算結果はオブジェクト選択コードを応答コードに替えて、精度情報を付加して、商と剰余をシステムバス上に送出する。

〈4・3〉 ソートコア k ビット $\sim k \times r$ ビットまでのソートが可能な〈3・3〉で設計したソートモジュールを複数個カスケード接続することにより、その整数倍のデータ数（長）のソートが可能になる（図7）。バスが使われていないシステムバスはすべて“1”になっており、1ワード（ k ビットとする）のオブジェクト選択メッセージがバス上に送出されるとシステムバスに接続された全コアは自分宛てであるかを判断する。ソートの指定であるとソートコアだ

図5 加算コアの構成

Fig. 5. Structure of adder core.

けが続けて送られてくる k ビットのデータ精度とデータ数を順次取り込み、必要なデータを最下位モジュールから上位モジュールへと取り込んでいく。精度情報とデータ数により必要なデータの並び替えを行い、結果は先ず、オブジ

ェクト選択コードをすべて“0”にして応答コードに替え、精度情報、データ数、の次に並べ替えたデータ列を順に、ワード単位でバス上に出送する。

〈4・4〉 IP コアの再利用 補数を用いることで全ての

図 6 除算コアの構成

Fig. 6. Structure of divider core.

図 7 ソートコアの構成

Fig. 7. Structure of sort core.

図 8 コアの使用例

Fig. 8. Example using each core.

算術演算は加算で行うことができる。従って、〈4・1〉の加算コアを必要に応じて選択し、任意精度の算術演算に用いることができる。あるコア（コア A とする）の処理時に加算演算が必要になった場合、コア A は図 2 (a) のフォーマットで先ず、加算コアを指定するオブジェクト選択コードをバス上に送出する。これにより加算コアだけが続けて送られてくる精度情報により、被加数と加数を取り込み、加算コアは加算結果を応答コードの次に精度情報を付加してバス上に送出し、コア A が加算結果を取り込む（図 8）。

集積回路技術が向上した現在では、同様に直接減算コアを無理なく用意できる環境下にあり、減算コアも用意し、これらの演算を繰り返すことにより乗除算結果を得ることができる。しかし、乗除算には頻繁に加算／減算コアの使用を繰り返すため、バスの使用効率や演算時間を考慮すれば、システム LSI のような大規模な集積回路においては〈4・2〉の除算コアと同様に構成した乗算コアも実装しておくことが望まれる。図 8 に四則演算の各コアを配置した構成例を示す。一般の四則演算を行いたい場合はメッセージでどのコアかを指定し、続けて精度情報とパラメータをバス上に送出して、指定したコアから処理結果を得る。

例えば、情報のセキュリティ技術として公開鍵暗号システムの重要性が増し、専用のチップが開発されている^{(9)~(12)}が、乗除算コアを用いて RSA 公開鍵暗号システムを実現することができる。RSA 公開鍵暗号システムでは暗号化／復号には

$$M^e \equiv C \pmod{n}$$

という剰余演算を行う。それぞれ 1,024 ビット前後の可変長精度が求められ、自乗の剰余乗算と一般の剰余乗算が多数回繰り返される。これをコア（C）が行う場合、乗算コアと除算コアを順次使うことになる（図 8）。

5. まとめ

本論文は外部からの制御を受けることなく、モジュール内部で処理実行に必要な操作を判断する自立形で、また、処理精度の制限を受けないハードウェアモジュールおよび IP の再利用の際に課題となっているインタフェースの標準化と使用容易性を考慮したオブジェクト指向設計法について提案するものである。従って、回路を記述、合成する SpecC、System C や HDL といった設計手法のようなオブジェクト指向を設計言語に適用したものとは根本的に異なるものである。設計過程で用いる記述法や言語は問題ではない。IP コアの標準化に大きく影響するインタフェースを「入出力／制御メッセージ」だけとし、メッセージ内のオブジェクト選択コードによって、複数の IP コアから成るシステム LSI 内の目的とするコアを起動させて、結果を得ることのできる IP コアのモジュール化をいかに設計するかというのが主題である。

本設計法を説明するため、基本的な四則演算器とソートへの適用例について述べた。システム LSI への試作は行っていないが、それぞれシミュレーションおよび FPGA での

動作確認を行った。それぞれゲート規模、演算精度や速度、回路方式および回路技術の向上による改良などにより、コアの内部機構は多様に変化する。しかし、オブジェクト化されたコアはその目的や環境に応じて、オブジェクト（部品）交換での置き換えが容易であり、提案する設計法はシステム LSI 時代に適応した技法である。

（平成 14 年 10 月 28 日受付、平成 15 年 11 月 25 日再受付）

文 献

- (1) Y.Hokari: "Review of the ITRS2001 Technology Roadmap", *IPSS SIG Technical Report on SLDM*, Vol.2002, No.46, pp.55-60 (2002-5) (in Japanese)
徳刈泰明:「半導体技術ロードマップの最新動向」, 情処研報, **2002**, 46, pp.55-60 (2002-5)
- (2) M.Bimbaum and H.Sacks: "How VSIA Answers the SOC Dilemma", *IEEE COMPUTER*, pp.42-50 (1999-6)
- (3) 木下常雄: システムレベル設計とは何か, *Design Wave Magazine*, pp.28-33 (2000-7)
- (4) H.Mukai: "Integrated Circuits Technology Trend, Stepping into System LSI Era", *Oki Technical Review*, OTR180, Vol.66, No.1, Oki Electric Industry Co., Ltd.
向井久和:「集積回路技術動向—システム LSI 時代を迎えて」, 沖電気研究開発, 第 180 号, Vol.66, No.1, pp.3-6 (1999-5)
- (5) 情報通信活用辞典編集委員会編:「情報通信活用辞典」, (株)産業調査会辞典出版センター (2000-6)
- (6) T.Nakamura and H.Kasahara: "A Proposal of Introducing the Object-Oriented Technique for Hardware Design and the Modularizing of VSI core", *IEEJ Trans.EIS*, Vol.121, No.3, pp.564-573 (2001-3) (in Japanese)
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, **121**-C, 3, pp.564-573 (2001-3)
- (7) N.Suzukawa, T. Nakamura, and H.Kasahara: "A New Hardware design method for ULSI", *IEICE GENERAL CONFERENCE*, A-3-14 (2001-3) (in Japanese)
鈴川敦之・中村次男・笠原 宏:「大規模集積回路技術に対応した新しいハードウェア設計法の提案」, 2001 年信学総大, A-3-14 (2001-3)
- (8) T. Nakamura and H.Kasahara: "An Architecture of a High-Speed Quasi-Parallel Divider for Numbers of Arbitrary Word Length", *IEEJ Trans. EIS*, Vol.120, No.1, pp.158-167 (2000-1) (in Japanese)
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, **120**-C, 1, pp.158-167 (2000-1)
- (9) T.Hasegawa and H.Yamamoto: "Division Method of Modular Exponentiation Using Redundant Binary Representation", *IPSS SIG Technical Report on ARC*, 129-8 (1998-5) (in Japanese)
長谷川俊夫・山元浩幸:「冗長 2 進数表現を利用したべき乗剰余演算回路の分割処理」, 情処研報, 計算機アーキテクチャ, 129-8 (1998-5)
- (10) S.Ishii, K.Ohyama, and K.Yamanaka: "High-Speed Public Key Encryption Processor", *TIEICE Japan*, D-1, Vol.J80- D-1, No.8, pp.725-735 (1997-8) (in Japanese)
石井晋司・大山勝一・山中喜義:「高速公開鍵暗号プロセッサ」, 信学論, D-I, **J80-D-I**, 8, pp.725-735 (1997-8)
- (11) 日本 IBM (株):「1024 ビット RSA 暗号アクセラレータ」, NIKKEI BYTE (1999-2)
- (12) T. Nakamura, H.Ohishi, and H.Kasahara: "A Bit Sliced System on RSA Public Key Cryptosystem implementation", *IEEJ Trans. EIS*, Vol.118, No.7/8, pp.1073-1081 (2000-7/8) (in Japanese)
中村次男・大石博朗・笠原 宏:「RSA 公開鍵暗号システム実装におけるビットスライス化の一方式」, 電学論 C, **118**, 7/8, pp.1073-1081 (1998-7/8)
- (13) 海野雅史:「～SiP&システム・モジュールの設計～SiP モジュールとは何か?」, *Design Wave Magazine*, pp.36-43 (2002-10)

中 村 次 男

(正員) 1971年3月東京電機大学工学部電気工学科卒。同年4月日本電気精器(株)入社。1990年3月同大学大学院修士了。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)。デジタル演算器、高速暗号、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「デジタル回路の基礎」(日本理工出版会)、「デジタル回路設計法—ワンチップ化の実例集—」(日本理工出版会)、「デジタル回路入門」(日本理工出版会)、「電子回路(2) デジタル編」(コロナ社)、「電気・電子—なぜなぜおもしろ読本」(山海堂)。IEEE 会員

鈴 川 敦 之

(非会員) 2002年3月東京電機大学大学院修士課程了。同年4月、ローム(株)入社。在学中、デジタル演算器、公開鍵暗号処理の高速化、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。

冬 爪 成 人

(正員) 1995年3月東京電機大学工学部電気工学科卒。2000年同大学大学院博士課程了。同大学工学部電気工学科助手を経て、現在、同大学情報環境学部情報環境工学科助手。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。電子情報通信学会会員。

笠 原 宏

(正員) 1970年3月東京電機大学大学院博士課程満期退学。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、情報環境学部情報環境工学科教授。工学博士。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。情報処理学会、電子情報通信学会、IEEE、ACM 各会員。

田 中 照 夫

(正員) 1943年9月19日生。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。主としてパワーエレクトロニクスに関する研究に従事。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。IEEE 会員