

## PCA-Chip2 におけるデータ駆動型回路設計手法

越智 直紀<sup>†</sup> 湯浅 隆史<sup>†</sup> 泉 知論<sup>†</sup> 尾上 孝雄<sup>††</sup> 中村 行宏<sup>†</sup>

<sup>†</sup> 京都大学 大学院 情報学研究科 通信情報システム専攻  
〒606-8501 京都市 左京区 吉田本町

<sup>††</sup> 大阪大学 大学院 情報科学研究科 情報システム工学専攻  
〒565-0871 大阪府 吹田市 山田丘 2-1

E-mail: reconf@easter.kuee.kyoto-u.ac.jp

**あらまし** 再構成可能論理デバイスを発展させたアーキテクチャとして、プラスチック・セル・アーキテクチャ(PCA)が提案されている。PCA では、オブジェクトと呼ばれる小規模回路群をパイプライン通信路により接続し、非同期的に協調動作させることにより所望の機能を実現する。本稿では、PCA の試作デバイスである PCA-Chip2 をターゲットとし、PCA-Chip2 上に所望の機能を実現するための回路構成法を提案する。PCA の非同期的動作に基づいたデータ駆動型の回路構成を提案し、それを実現するためのデータフロー制御や基本演算などの要素オブジェクトを設計する。これら要素オブジェクトを用いた設計事例により、本設計手法について説明する。

**キーワード** 再構成可能論理デバイス, プラスチック・セル・アーキテクチャ, 設計手法, データ駆動

## A Design Scheme for Data Driven Circuits on PCA-Chip2

Naoki OCHI<sup>†</sup>, Takafumi YUASA<sup>†</sup>, Tomonori IZUMI<sup>†</sup>, Takao ONOYE<sup>††</sup>,  
and Yukihiro NAKAMURA<sup>†</sup>

<sup>†</sup> Department of Communications and Computer Engineering,  
Graduate School of Informatics, Kyoto University  
Yoshida-hommachi, Sakyo-Division, Kyoto-City, 606-8501 Japan

<sup>††</sup> Department of Information Systems Engineering,  
Graduate School of Information Science and Technology, Osaka University  
2-1 Yamada-Oka, Suita, Osaka, 565-0871 Japan  
E-mail: reconf@easter.kuee.kyoto-u.ac.jp

**Abstract** Plastic Cell Architecture (PCA) is proposed as an extension of Programmable Logic Devices (PLDs) with unique characteristics. A target function is implemented on PCA by asynchronous cooperation of small circuits referred to as objects connected via pipelined communication paths configured on the device. In this paper, a design scheme to implement a target function on our prototype device PCA-Chip2 is presented. We propose a framework of data driven circuits based on asynchronous behavior of PCA and design primitive objects for the framework such as objects to control data flow and objects for arithmetic operations. A design of a function with the proposed primitive objects is presented as a case study.

**Key words** Reconfigurable Logic, Plastic Cell Architecture, Design Methodology, Data Driven

## 1. はじめに

現在半導体技術が大きく発展し、コンピュータ等の中枢部分であるLSI(Large Scale Integrated circuit)は急激な進歩を遂げている反面、それらの開発には多大な時間、費用、人手、技術力が必要とされる。また、多様な製品の開発が求められる反面、製品寿命は短くなっている。これらの問題を軽減する有用なデバイスとして、PLD(Programmable Logic Device)が様々な場面で用いられている。PLDはユーザが設計した機能を手でハードウェア化可能なデバイスであり、近年では回路の動作中に再構成(動的再構成)可能な製品も登場している。動的再構成可能なハードウェアの研究は、汎用コンピュータの柔軟性とASIC(Application Specific Integrated Circuit)のような高い並列性能の両立をめざし、リコンフィギュラブルコンピュータリングと呼ばれる分野で盛んに行われている。

PLDを用いることによって、開発期間や費用、試作検討などの問題が軽減されるが、従来のCPUやASICなどと同様に配線遅延やクロックスキューの問題は変わらず存在する。また、回路規模が増大する割合よりも配線領域が増大する割合の方が大きく、その結果、論理の実現に使用可能な領域の割合が減少し、論理密度の低下を招くことになる。

そこで、PLDを発展させたアーキテクチャの1つとしてPCA(Plastic Cell Architecture) [1] [2]が提案されている。PCAでは、従来のようにデバイス上に単一の同期回路を構成するのではなく、オブジェクトと呼ばれる小規模な回路を複数構成し、それらをパイプライン接続して協調動作させることで、所望の機能を実現する。この機構によってクリティカルパスを短くすることが可能となり、配線遅延やクロックスキューの問題を軽減可能である。各オブジェクトはそれぞれが独立して動作しており、デバイス動作中にオブジェクトを書き換える動的再構成が可能である。またPCAでは、外部からの制御信号だけでなく、内部のオブジェクトで生成された制御信号でも動的再構成可能である。

本研究室では、PCAを実現するデバイスとしてPCA-Chip2 [3]を試作しており、本稿ではこれを研究対象とする。PCA-Chip2に関しては、これまでに、デバイスについての研究 [3] [4] [5]、ツール類についての研究 [4] [6]、オブジェクト設計やライブラリ作成 [7] [8]、運用方法やアプリケーションについての研究 [9] [10] など、様々な成果が発表されている。

本稿では、PCA-Chip2における基本的な動作モデルの検討と提案、演算器 [7] や制御回路 [8] などの要素オブジェクトの設計及び、それらの提案モデルと要素オブジェクトを用いた機能回路の設計例について述べる。

以下、第2章でPCA-Chip2の概略について述べ、第3章で前提とする基本的な動作モデルについて述べる。第4章では基本機能オブジェクトについて述べ、それらを用いた設計例を第5章で述べる。そして、第6章でまとめと今後の課題について述べる。

## 2. PCA-Chip2

本章では、PCAおよび、本研究室の試作デバイスであるPCA-Chip2の概略を述べる。

### 2.1 PCAの概略

PCAは、概念的には組み込み機能層と可変論理層と呼ばれる2つの層から構成される。組み込み機能層は組み込み機能部と呼ばれる基本単位の集合であり、可変論理層は可変論理部と呼ばれる基本単位の集合である。実際は、1つの組み込み機能部と1つの可変論理部の対がPCAセルと呼ばれる基本単位を成し、これを2次元メッシュ状に配置した構造を採っている(図1)。PCAセルを単数または複数用いて、オブジェクトと呼ばれる任意の機能回路を実現する。PCAを実現するデバイスとしては、現在PCA-2 [11]、PCA-Chip2 [3]などが提案されている。

### 2.2 組み込み機能部

組み込み機能部は、対になる可変論理部及び隣接するPCAセルの組み込み機能部と接続されている。組み込み機能部は主に、PCAセル間の通信経路の設定、可変論理部の制御や構成、オブジェクト間のデータ伝送などを行う。

オブジェクト間では、経路を構成する組み込み機能部によるパイプライン通信が行われる。これにより、信号の伝搬においてレイテンシが増加するが、クロック周波数を高めることでスループットの向上が期待できる。また、各オブジェクトはそれぞれ独立動作しており、デバイス動作中もオブジェクトの消去や上書きが可能である。さらに、オブジェクトの制御は外部からの制御信号だけでなく、同一デバイス内の他のオブジェクトから行うことも可能である。

PCA-Chip2のデータの伝送は、ビジー信号とリクエスト信号を用いたハンドシェイク通信で行われる。PCA-Chip2の通信路は12ビット幅であり、最上位ビットから順に、リクエスト信号、ビジー信号、命令/データの判別信号、残る9ビットがデータになる。これら12ビットで1ワードとし、その下位10ビットを特にパーティクルと呼ぶ。

### 2.3 可変論理部

可変論理部では所望の論理を実現する。PCA-Chip2の可変論理部は、組み合わせ回路を実現するためのLUT(Look-Up Table)アレイ、ORアレイと、順序回路を実現するためのFlip Flop部によって構成される(図2)。組み合わせ回路を実現する部分は、PLA(Programmable Logic Array)のANDアレイをLUTアレイに置き換えた構造といえる。

LUTアレイ中の各LUTは3入力1出力を持ち、2つの入力

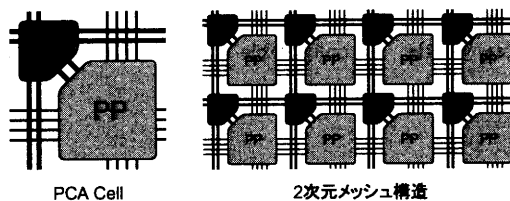


図1 PCAの構造

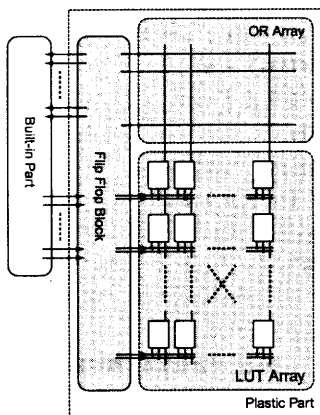


図 2 PCA-Chip2 可変論理部概略図

変数と前段 LUT からの入力により出力を生成する。LUT は直列多段接続されており、それにより生成された論理を項 (term) と呼ぶ。1 つの可変論理部の LUT アレイには 8 個の LUT による直列多段接続が 24 列含まれており、最大で 24 の項を生成できる。

OR アレイでは LUT アレイで生成された項の論理和をとる。1 つの可変論理部の OR アレイには、32 ビットの出力信号線がある。出力信号線は 2 ビットを 1 組にして使い、それぞれフリップフロップの制御とデータの出力に用いる。32 ビット中 24 ビットのデータ出力先は、組込み機能部またはフリップフロップのいずれかを選択可能であり、残る 8 ビットはすべてフリップフロップに出力される。

Flip Flop 部には 16 個のフリップフロップが含まれる。PCA-Chip2 では、これらを用いて順序回路を構成する。

可変論理部を単数または複数まとめて用いることで、1 つのオブジェクトを構成できる。

### 3. 基本動作モデル

#### 3.1 データ駆動型演算モデル

PCA-Chip2 の通信方式からデータ到着のタイミングは経路長に依存するので、異なる通信路により伝送されるデータの同期をとる必要がある。そこで、各オブジェクトにデータの待ち合わせ機構を設け、演算に必要なデータがすべてそろった時点でデータを通信路上に待機させる。このように、演算に用いるデータの到着をもとに演算を行うデータ駆動型モデルを採用する。

#### 3.2 データ形式

オブジェクトで処理するデータパーティクルを、演算の入出力データとなる演算パーティクルと、制御オブジェクトにおける真偽値などの制御信号となる制御パーティクルの 2 種類に分類する。

データの形式について議論する。一般的な計算機の多くでは 32 ビットや 64 ビットなど、8 の倍数のビット幅で演算される。よって、作成した各オブジェクトの演算データも 8 の倍数のビット数とする。1 ワード内のデータ形式は、下位ビットから

順に以下のように定める。負数は 2 の補数で表す。

- データ (下位 8 ビット)
- データ終了フラグ (下位から 9 ビット目)
- コマンドとデータの判別
- ビジー信号
- リクエスト信号

データパーティクルの転送について議論する。PCA-Chip2 ではデータ部が 9 ビットあるため、ビットシリアル転送は効率的な転送方式ではない。ワードパラレルではビット幅に応じて使用する通信路やオブジェクトの入出力ポートが増加する。オブジェクトに同時に入力するビット幅の増加は、LUT 多段接続の段数に直接関係するため、回路の動作速度低下も引き起こす。それに対してワードシリアルの場合、入力ワード数が多くなれば演算に要するサイクル数が増加してしまいが、入力データのビット幅が増加してもセル数は増加しない。また、小さなオブジェクト群のパイプライン化によるスループット向上を狙うことができる。

以上より、本モデルでは原則的にワードシリアル回路を用い、必要に応じてワードパラレル回路も用いることとする。データの入出力は、加算や乗算に用いられる桁上げを効率良く処理するために下位桁ワードから行う。よって、最後尾ワードとなる最上位ワードのデータ終了フラグが 1、それ以外は 0 となる。

### 4. 要素オブジェクト

オブジェクトは、演算オブジェクトと制御オブジェクトの 2 種類に分類する。本章では、PCA-Chip2 用の演算オブジェクトや制御オブジェクトの概要と、それらを用いて設計した機能オブジェクトについて検討し、さらに問題点などについて述べる。

#### 4.1 制御オブジェクト

制御機能としては、条件分岐の if や switch、ループ処理の while や for などがある。このうち、基本となるのは if 及び while である。switch は if を何段にも繰り返すことで作成でき、for は while に繰り返し判定をする回路を加えることで解決できる。そして、これらの機能は、分岐、合流、ループ制御付き合流、分岐、終端器、データ破棄型合流という 6 種類の要素オブジェクトの組み合わせで実現できる [8]。要素オブジェクトについて、オブジェクト数、項数、セル数は、表 1 のようになる。

表 1 要素オブジェクトの項数およびセル数

	オブジェクト数	総項数	セル数 (縦 × 横)
分岐	1	15	2 × 1
合流	1	28	3 × 2
ループ制御付き合流	1	37	4 × 2
分流	1	12	2 × 1
終端器	1	0	1 × 1
データ破棄型合流	1	21	3 × 1

## 4.2 演算オブジェクト

演算オブジェクトは、加算器と乗算器の2種類が作成されている [7]。それぞれの演算器についてシャッフル無し型、順序回路型、パイプライン処理型の3種類が作成されており、8bit 演算回路のオブジェクト数、総項数、セル数は、表 2,3,4 の通りである。シャッフルとは入力データをビット単位で並び替える機能である。入力データの並び替えに1ステップ必要だが、PCA-Chip2 の可変論理部の構造では入力データの順序に項数が大きく依存するため、セル数の削減に大きく寄与する。シャッフル無し型では1オブジェクトで処理を行う。シャッフルを行わないために項数が増大し、セル数が多すぎるので実際には使用しない。順序回路型では、シャッフルも含めた全ての処理を1つのオブジェクト内で状態遷移を行って処理する。パイプライン型では1ステップを1つのオブジェクトで処理するため、ステップの数と同数の小さなオブジェクトに分割される。

表 2 8bit ワードシリアル加算器

	オブジェクト数	総項数	セル数 (縦×横)
シャッフル無し型	1	1532	2×62
順序回路型	1	71	3×3
パイプライン処理型	2	36(*1)	4(*2)

\*1: 1 段目;18, 2 段目;18, 合計 36 項

\*2: 1 段目;2×1, 2 段目;2×1, 合計 4 セル

表 3 8bit ワードパラレル加算器

	オブジェクト数	総項数	セル数 (縦×横)
シャッフル無し型	1	1004	2×42
順序回路型	1	43	3×2
パイプライン処理型	2	28(*3)	4(*4)

\*3: 1 段目;18, 2 段目;10, 合計 28 項

\*4: 1 段目;2×1, 2 段目;2×1, 合計 4 セル

表 4 8bit ワードパラレル乗算器

	オブジェクト数	総項数	セル数 (縦×横)
シャッフル無し型	1	23791	2×992
順序回路型	1	213	4×9
パイプライン処理型	8	345(*5)	64(*6)

\*5: 1 段目から 63, 40, 41, 42, 43, 44, 29, 43, 合計 345 項

\*6: 1 段目;4×3, 2~7 段目;4×2, 8 段目;2×2, 合計 64 セル

## 5. 機能回路の設計例

提案モデル並びに設計した要素オブジェクトを用いた、機能回路の設計例について述べる。設計する回路は、入力  $a > 0$  に対して出力  $x = \sqrt{a}$  を整数の範囲で求める平方根演算器である。図 3 に入力  $a$  が 8bit の場合の動作記述を示す。2 の補数を用いているので  $0 < a \leq 127$  となる。よって、 $1 \leq x \leq 11$  の範囲で解が存在する。

```

for ( i = 0 ; i < 11 ; i++ ){
    n = (i+1)*(i+1) - a;
    if ( n == 0 ) {
        x = i+1;
        i = 11;
    } else if ( n > 0 ) {
        x = i;
        i = 11;
    }
}

```

図 3 平方根演算の動作記述

### 5.1 要素オブジェクトによる設計

設計した回路のブロック図を図 4 に、入力  $a$  が 8bit の場合のレイアウト例を図 5 に示す。使用したオブジェクト (群) は以下の表 6 の通りである。演算器はパイプライン処理型か順序回路型を使用することになるが、今回は加算器、乗算器とも 8 ビットパイプライン型を用いて設計する。要素オブジェクト (群) 以外には、待ち合わせ、インクリメント、デクリメント、0 出力、符号反転、条件判定を設計して用いた (表 5)。

平方根演算オブジェクト群全体では、合計で 115 個のセルを要する。また、このレイアウト例 (図 5) では、オブジェクト全体の最大縦セル数が 8、横セル数が 26 である。

### 5.2 オブジェクトの最適化

ここまでは要素オブジェクトをできるだけ使用して作成したが、一部の回路を変更することで面積や配線の改善が可能である。具体的には、分流と乗算を用いて入力の 2 乗を得ている部

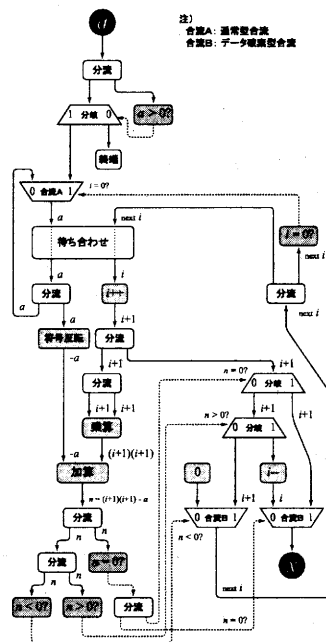


図 4 ブロック図

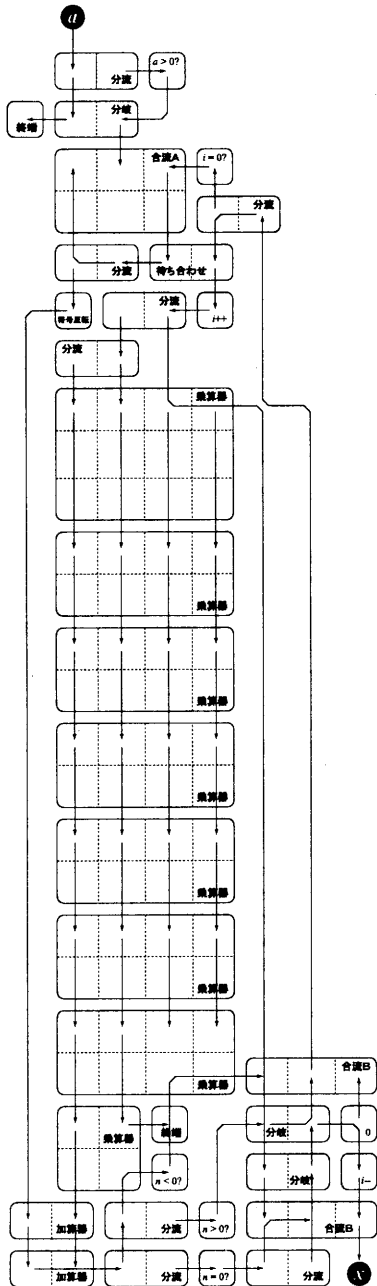


図5 レイアウト例

分を2乗演算器に変更し、0出力とデータ破棄型合流を用いている部分を2項演算器に変更する。今回は入力値  $a$  が8ビットなので、4ビットの2乗演算器があればよい。それらのオブジェクトの項数とセル数を表7に、ブロック図を図6に、レイアウト例を図7に示す。

この結果、改良後のオブジェクトは総セル数が50、縦セル数の最大が7、横セル数の最大が10となった。改良前と比べると、総セル数が56.5%減少した。

表5 追加オブジェクト

	オブジェクト数	総項数	セル数 (縦×横)
待ち合わせ	1	21	2×1
インクリメント	1	12	2×1
デクリメント	1	12	1×1
0出力	1	4	1×1
符号反転	1	12	1×1
条件判定	1	5	1×1

表6 利用した要素オブジェクト (群)

	図5で使用した数	占有セル数
分岐	3	6
合流 (合流A)	1	6
分流	8	16
終端器	2	2
データ破棄型合流 (合流B)	2	6
加算器 (パイプライン型)	1	4
乗算器 (パイプライン型)	1	64
待ち合わせ	1	2
インクリメント	1	1
デクリメント	1	1
符号反転	1	1
条件判定	5	5
0出力	1	1
合計	-	115

表7 変更箇所のオブジェクト

	オブジェクト数	総項数	セル数 (縦×横)
4ビット2乗演算器	1	12	1×1
2項演算器	1	12	2×1

### 5.3 評価と考察

設計例の平方根演算器について考察する。今回は専用の機能回路を1から設計するのではなく、既存の要素オブジェクトとわずかな追加オブジェクトを組み合わせることで設計した。追加オブジェクト自体も非常に簡単なものであり、この設計手法を用いることで専用の機能回路の設計よりも少ない労力で機能を実現できる。また、要素オブジェクトをさらに開発設計することで、他の機能も同様の手法で実現可能と考えられる。

このようにして設計された平方根演算器において、目立って占有セル数が多いのが乗算器である (表3,6)。平方根演算器の合計115セルのうち、乗算器が55.7%である64セルを占有している。それに加えて、配線が多いために配置配線の自由度が低くなってしまっている。乗算器は2乗演算をしているだけであるので、改良するにあたっては専用の2乗演算器を設計し、分流器と乗算器を置き換えた (図6)。この変更によって、セル数と配線を大幅に改善できた。また、0出力と合流Bを2項演算器にマージすることで、わずかながら面積と配線を改善できた。

要素オブジェクトで実現される機能は、様々な場面で用いられることが予想される。演算器以外のオブジェクトは、追加オブジェクト (表5) を含めてもセル数が1や2のものも多く、

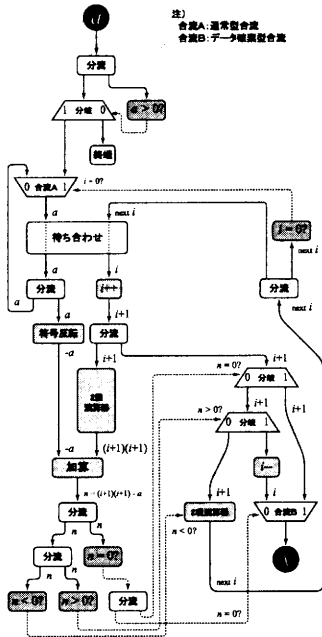


図6 ブロック図 (改良版)

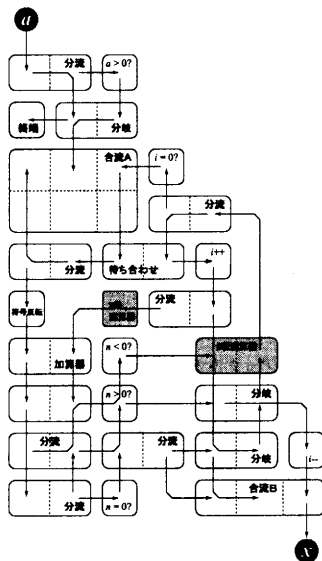


図7 レイアウト例 (改良版)

最大でも6である。それに対して乗算器のセル数は、8ビットのワードパラレルでも36や64であり、ビット幅の増加やワードシリアル化などを行うとさらにセル数が増加することが予想される。乗算器のセル数削減は、今後のオブジェクト開発における課題の1つである。

## 6. まとめと展望

本稿では、専用のオブジェクトを1から機能を設計して機能を実現するという方法でなく、要素オブジェクトの組み合わせ

で機能を実現するという回路構成方法と動作モデルを提案し、その設計例を示した。このことから、PCA-Chip2 向けの回路設計における多くの知見を得ることができた。

今後の課題としては、さらなる要素オブジェクトの設計やそれらを用いた機能の設計、演算オブジェクトの面積削減方法の検討が挙げられる。演算オブジェクトの面積削減に関しては、演算器自体の改良という方法も考えられるが、PCA-Chip2 の可変論理部の構造自体がセル数の増加に大きく影響していることも考えられるので、可変論理部の構造についても考察と改良をしていく必要がある。特に、PCA-Chip2 で設計した加算や乗算は桁上げ伝搬型であり、下位ビットで得られた値を上位ビットで何度も使用する。ある項で得られた値を他の項で何度も繰り返し求めていることになるので、たとえばこの値を途中で巧く引き出すことができれば、項数とセル数の削減が可能であると考えられる。このような可変論理部に関する検討を行うことも、今後の課題とする。

## 文 献

- [1] 永見, 塩澤, 伊藤, 小栗, 中村: “オブジェクト指向 HDL のための FPGA アーキテクチャ”, DA シンポジウム'97, pp. 209-214 (1997).
- [2] K. Nagami, K. Oguri, T. Shiozawa, H. Ito and R. Konishi: “Plastic Cell Architecture: Towards Reconfigurable Computing for General-Purpose”, Proc. of 6th Annual IEEE Symposium on FPGAs for Custom Computing Machines, pp. 68-77 (1998).
- [3] 冨田, 杉本, 筒井, 境, 檜田, 泉, 尾上, 中村: “LUT アレイ型 PLD の設計と試作”, 電子情報通信学会技術研究報告 VLD2000-98, ICD2000-155, FTS2000-63, 電子情報通信学会 (2000).
- [4] H. Tsutsui, A. Tomita, S. Sugimoto, K. Sakai, T. Izumi, T. Onoye and Y. Nakamura: “LUT-Array-Based PLD and Synthesis Approach Based on Sum of Generalized Complex Terms Expression”, IEICE Trans. on Fundamentals, E84-A, 11, pp. 2681-2689 (2001).
- [5] Y. Soga, T. Yuasa, T. Izumi, T. Onoye and Y. Nakamura: “An Improved Communication Channel in Dynamic Reconfigurable Device for Multimedia Applications”, Proc. of Euromedia Conference, pp. 152-157 (2003).
- [6] T. Okamoto, T. Yuasa, T. Izumi, T. Onoye and Y. Nakamura: “Design Tools and Trial Design for PCA-Chip2”, IEICE Trans. on Information and Systems, E86-D, 5, pp. 868-871 (2003).
- [7] 越智, 湯浅, 泉, 尾上, 中村: “LUT アレイ型可変論理デバイスのための基本演算器の構成法”, 第1回リコンフィギャブル研究会, 電子情報通信学会 (2003).
- [8] 湯浅, 曾我, 越智, 泉, 尾上, 中村: “プラスチックセルアーキテクチャにおけるハンドシェイク通信路を用いた制御論理モデルとその設計”, 電子情報通信学会技術研究報告 CAS2003-31, VLD2003-41, DSP2003-61, 電子情報通信学会.
- [9] 長川, 岡本, 湯浅, 泉, 尾上, 中村: “自己再構成型論理デバイスにおける時分割処理のための機能切替手法”, 第1回リコンフィギャブル研究会, 電子情報通信学会 (2003).
- [10] T. Izumi, K. Tada, T. Yuasa, T. Onoye and Y. Nakamura: “An Adaptive Load Distribution Model for Self-Reconfigurable Logic Device”, Proc. of the 1st NorthEast Workshop on Circuit and Systems, pp. 13-16 (2003).
- [11] 伊藤, 小西, 中田, 坪井, 名古屋: “動的再構成可能論理 LSI PCA-2”, 第1回リコンフィギャブル研究会, 電子情報通信学会 (2003).