

## 動的再構成可能計算機機構のためのプラスチックセルアーキテクチャ

伊藤 秀之 小栗 清 永見 康一 小西 隆介 塩澤 恒道

NTT 光ネットワークシステム研究所  
〒 239-0847 神奈川県横須賀市光の丘 1-1

{hi,oguri,nagami,ryusuke,shiozawa}@exa.onlab.ntt.co.jp

## 要旨

複雑な処理を実行する上で、汎用計算機にとってオブジェクトの動的な操作機構は不可欠である。現在のプログラム内蔵型計算機は、この機能をメモリ空間上にオブジェクトを配置することで実現している。この機構を布線論理の世界に導入することで、布線論理の持つ並列性とプログラム論理の持つ柔軟性を兼ね備えた汎用の再構成可能計算機アーキテクチャを実現できると考えられる。本稿では、布線論理オブジェクトの動的な操作機構を持つ再構成可能計算機アーキテクチャの実現方法を提案する。この計算機アーキテクチャは、組み込み論理部分と可変論理部分の二重構造を持つセルで構成される。このセルの構造と組み込み機能について述べる。

## The Plastic Cell Architecture for Dynamically Reconfigurable Computing

Hideyuki Ito Kiyoshi Oguri Kouichi Nagami Ryusuke Konishi Tsunemichi Shiozawa

NTT Optical Network System Laboratories  
1-1 Hikari-no-oka Yokosuka-shi Kanagawa 239-0847 Japan

{hi,oguri,nagami,ryusuke,shiozawa}@exa.onlab.ntt.co.jp

## Abstract

*Dynamic manipulation of the object is an essential function for the general purpose computer to perform complex procedures. Conventional stored-program computer realizes it by allocating objects in the memory space. It is considered that introducing this mechanism to the wired logic realizes an general purpose reconfigurable computer architecture which possesses both parallelism of the wired logic and flexibility of the program logic. In this paper, we propose a reconfigurable computer architecture which has functionality of dynamical reconfiguration of the hardware object. This architecture has a feature that it consists of the cell of duality of built-in functionality and programmable functionality. We mention the cell structure and its built-in functionality.*

## 1 イントロダクション

近年の再構成可能論理素子の発達に伴って、再構成可能計算機アーキテクチャに関する研究が数多く行なわれている [1][2][3][4]。この計算機アーキテクチャは、FPGA 等の再構成可能論理素子上にターゲットとするアプリケーションを実現する論理回路を構成し、布線論理で処理を行なうものである。その狙いは、布線論理の持つ並列性を利用した処理速度の向上にあるといえる。

アプリケーションを実現する手段として、布線論理とプログラム論理の二つの方法が考えられる。布線論理による実現では、アプリケーションに特化した論理回路で実行するので、論理演算レベルでの並列性を引き出すことで高速な処理が可能になる。しかし、専用のハードウェアが必要なため、プログラム論理のような柔軟性、汎用性に欠けている。そして、汎用計算機機構である CPU には技術革新も集中し、処理性能の向上が著しく、数年たつと CPU とメモリで構成される

プログラム内蔵型計算機の性能が専用の布線論理の性能に勝ってしまう場合が多い。

だが、再構成可能論理素子の登場により、布線論理の世界に柔軟性を採り入れることが可能になりつつある。ハードウェア記述言語と高位論理合成システムの技術を用いれば、アプリケーション毎に布線論理を構成することも可能である。CPU とメモリからなる従来の計算機アーキテクチャに再構成可能論理素子を組み合わせることで、高性能かつ柔軟な再構成可能計算機アーキテクチャを実現できる可能性がある。しかし、我々は、このような計算機アーキテクチャは、処理速度と柔軟性の点で従来の計算機アーキテクチャに勝るには不十分だと考えている。

その一つの理由は、メモリの利用こそがフォンノイマン型計算機の本質であり、ある意味では、CPU はメモリを最も効率的に使うように設計された専用ハードウェアだからである。従って、先に述べたような再構成可能計算機アーキテクチャで、布線論理がメモリを使用するならば、CPU の性能に勝ることはできな

い。これが、上記のような計算機アーキテクチャでは布線論理の処理能力を十分に発揮できないと考える理由である。再構成可能計算機アーキテクチャで布線論理の性能を活かすためには、従来のようなメモリアクセスを排除した並列分散処理を考えるべきである。

更に、プログラム論理と布線論理には本質的な違いがある。それは、実行時のオブジェクトの動的な操作が可能か否かという点である。プログラム論理では、実行時にメモリ上にあるデータ構造を動的に扱うことができる。これは例えば、C言語に於けるメモリの動的な確保、解放機能 (malloc, free) や C++ 言語に於けるオブジェクトの生成、削除 (new, delete) 機能である。汎用の計算機に於ける複雑な処理の記述と実行には、このようなオブジェクトの動的な操作が不可欠である。しかし、布線論理ではそのような機能は考えられないので、実行できる処理の複雑さには限界がある。従って、汎用計算機機構の実現にはプログラム論理の方が適していると考えられる。

しかし、オブジェクトの動的な操作機能を導入できれば、布線論理によるより複雑な処理の実行が可能になる。つまり、布線論理の並列性とプログラム論理の柔軟性を持つ汎用の再構成可能計算機アーキテクチャの実現が可能になる。ここで、布線論理でのオブジェクトの動的な操作機能とは、論理回路等の布線論理オブジェクトが、実行時に自律的に別のオブジェクトの生成、削除などを行なう機能である。しかし、この機能を実現するためには、再構成可能論理素子の実行時の並列分散的な書き換え機能が必要である。

以降では、我々の提案する、布線論理オブジェクトの動的な操作機能を有する再構成可能計算機アーキテクチャの概念と、その実現仕様について述べる。この計算機アーキテクチャは、布線論理を構成するための可変論理部分と、これらの構成等に必要固定機能を提供する組み込み論理部分の二つの部分で構成される。本稿では組み込み論理部分の機能の設計を中心に述べる。

## 2 プラスティック セル アーキテクチャ

我々が提案する、布線論理オブジェクトの動的な操作機能を有する再構成可能計算機機構の概念と、その実現に必要なデバイスの基本構造について述べる [5]。

### 2.1 計算モデル

ハードウェア構造を自律的、動的に再構成するため、布線論理による並列オブジェクト指向的な計算モデルを考える。このモデルは、ハードウェア上でのメッセージを介した複数のオブジェクトの協調的な動作で処理を実現するものである。ここで、オブジェクトとは、再構成可能論理素子上に構成される論理回路やメモリである。このオブジェクトを機能単位にして、処理を実現する布線論理を構築する。一方、メッセージとは、オブジェクト間でやり取りするデータや制御情報である。オブジェクトはメッセージを生成

し、これを授受することで他のオブジェクトと協調動作する。

このようなオブジェクトのカプセル化は、ハードウェアの動的再構成のためには不可欠である。実行時のハードウェアの変更を効率的に行なうには、部分的な再構成や、並列分散的な再構成を行なう必要があるからである。布線論理の再構成は、オブジェクトを単位にして、それらの生成、消滅によって行なう。この再構成手続きは、ホスト計算機による構成データ書き換えのように集中化せず、オブジェクト自身の動作によって分散的に行なう。これにより、再構成処理の集中化による新たなボトルネックの発生を防ぐことができる。そのために、オブジェクトの生成はメッセージを用いて行ない、オブジェクト自体がオブジェクト生成用メッセージを送出することで、他のオブジェクトを生成するものとする。

布線論理による複雑な処理は、オブジェクトの並列動作と、オブジェクト間のメッセージの授受を明示的にプログラムすることで実現する。プログラマーは、これらをオブジェクト指向ハードウェア記述言語で記述する。この言語は、オブジェクトの動的な再構成機能をサポートすることで、ソフトウェア言語と同様に、複雑な処理の記述を可能にする。

複数のオブジェクトはハードウェア上に直接実現され、並列に動作する。また、オブジェクトの生成、消滅は実行時に必要に応じて行なう。よって、並列に実行可能な処理に対して、それらを実行するオブジェクトを生成して並列処理を行なうことが可能である。不必要なオブジェクトは、その処理の終了後に消去し、他のオブジェクトのために資源を解放する。従って、この計算モデルは、ハードウェア資源を有効に使ったスケラブルな並列処理を可能にする。

### 2.2 デバイス

構成すべき布線論理の多くは、HDL記述から高位論理合成システムで合成することになるため、論理素子の構造はマッピングの自動化に適したものでなくてはならない [6]。従来のFPGA等では、論理ブロックの粒度が大きく、それらの間の結合も単純ではない。例えば、4入力1出力のLookup Table(LUT)を用いると $2^2$ の論理関数を表すことができるが、それらの間の結合は階層化がされる等複雑である。この粒度の粗さと構造の複雑さは、布線論理を自動的にマッピングする際の障害となると考えられる。

マッピングの自動化を容易にするために、デバイスは細粒度で均質な構造であるべきである。そのために、再構成可能論理素子を全く同種の部品の結合網として構成する。論理回路の構成部品は、論理ゲート、レジスタ、結線である。細粒度化のため、論理ゲートの入出力数、レジスタのビット幅は小さくする。また、部品間の結合の自由度を保ち、構造を均質化するために、結線も他と同様に部品化して考える。即ち、4つの隣接セルと1ビット幅の接続線のみを持ち、これらのうち3入力に対する1出力の論理ゲート、1ビットのレジスタ、更に4つの接続線間を自由に結ぶ

結線として機能するセルを部品とする(図1)。再構成可能論理素子は、このセルの二次元メッシュ結合で構成する。各セルは、4つの隣接セルとのみ接続され、バス結合やその他の階層的な構造は持たない。

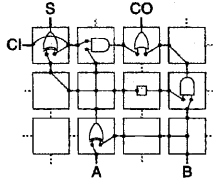


図1: フルアダー回路の部品化

次に、この再構成可能論理素子上に、他のオブジェクトの動作中に並列分散的にオブジェクトを構成する機構が必要になる。そのためには、メッセージを運ぶ仕組みと、そのメッセージを受けてオブジェクトを生成する仕組みが必要である。デバイスはマッピング効率を考慮して、グローバルな接続を排した均質なメッシュ構造になっているので、メッセージによる非局所的な通信もこのメッシュ構造を利用するのが望ましい。しかし、メッセージ伝達の機構がオブジェクトと同じセルの資源を使用すると、離れたオブジェクト間のメッセージ通信に対して他のオブジェクトが障害物となり、メッセージ伝達の自由度が低下する。そこで、このセル構造上にオブジェクトを実現する層とは別の機能を持つ層を設ける(図2)。この層は、メッセージを転送する経路としての機能と、オブジェクトを生成する機能を持つ。この機構は、セル中に幾つかの組み込み機能を提供する固定的な布線論理として実現する。

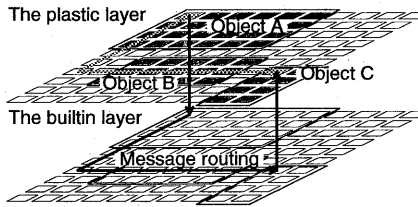


図2: 可変論理層と組み込み論理層

以上の機構を実現するため、次の2つの機能部分を持つセルをデバイスの構成単位にする。その一方は可変論理部分(plastic part)であり、これはオブジェクトを構成する部品となる。もう一方は組み込み論理部分(built-in part)で、メッセージを処理する。この部分が、本アーキテクチャに必要な組み込み機能を提供する。各部分は隣接セル間に独立した接続線を持ち、前述の二種類の機能層を提供する。また、セル内部には二つの機能部分間の通信路があり、これを用いて可変論理部層と組み込み論理層間での通信を行なう(図3)。

その構造から、以降、このアーキテクチャをプラスチックセルアーキテクチャ(Plastic Cell Architecture, PCA)と呼ぶ。このデバイス構造の導入により、2.1の計算モデルの実現に必要な、ハードウェアオブジェクトの動作、メッセージ伝達、そして、オブジェクトを動的に構成する機能の実現が可能になる。

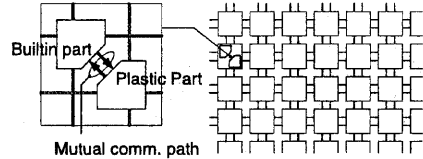


図3: 可変論理部分と組み込み論理部分

### 3 PCA の設計

2で述べたPCAの機能を実現するオブジェクト、メッセージ、セルの実現仕様について述べる。

#### 3.1 オブジェクト

オブジェクトは、可変論理部に構成情報を書き込んだセルの集合で構成する。オブジェクトは、特定の機能を実現するように設計した機能オブジェクトとしても、任意のデータを記憶するメモリオブジェクトとしても使用できる。

オブジェクトの各セルの可変論理部分に、論理回路の部品をマッピングすることで、任意の論理回路を実現する機能オブジェクトを作る。機能オブジェクトでは、内部の可変論理部分はマッピングされた論理回路に応じて結線で接続されるが、外部とはオブジェクトのカプセル化のために結線による接続は持たず、メッセージによる通信を行なう。メッセージの転送は組み込み論理層が行なうが、メッセージの生成と送出はオブジェクトが行なう。従って、メッセージを作成し組み込み論理層に送出する機構は、論理回路で実現し、機能オブジェクトの一部として埋め込む。また、オブジェクトの動作の制御のため、機能オブジェクト内にはタスクレジスタを設け、オブジェクト内のレジスタ値の更新などを制御する。この機構で、生成中のオブジェクトの不定な動作などを制御する。

また、可変論理部分は、その構成情報を保持するためのメモリを持つが、このメモリを任意のデータの格納にも利用する。1つのセルを1メモリセルとして、任意の容量のメモリオブジェクトを構成する。

#### 3.2 メッセージ

メッセージは、セルの組み込み論理部分により転送されるパケット列で構成される。パケットは固定長で、組み込み論理部分に与えられる命令か、可変論理部分に与えられるデータ片である。オブジェクト間のデータの交換や、オブジェクトの動作の制御は、メッ

セージの授受によって行なわれる。メッセージの一般的な形は図4のようになる。

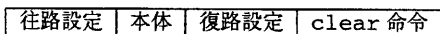


図4: メッセージの一般的な形式

オブジェクトが他のオブジェクトを一連のメッセージを用いて制御する場合に、メッセージの順序は送信時と受信時で保存されなくてはならないしかし、一連のメッセージに対する転送経路が異なる場合、これが保証されない場合がある。これを防ぐために、メッセージの経路は送信側で一意に指定する。経路設定は、送信点となるセルから目的地のセルまでの相対的な位置関係を指定することで行なう。

図4で、「往路設定」と「復路設定」の部分が、4.1で述べる routing 命令列で構成されるメッセージの経路情報である。前者は、送信オブジェクトが受信オブジェクトにメッセージの残り部分を送り届けるための情報である。後者は、送信オブジェクトが受信オブジェクトから何らかの応答を要する場合に付加する経路情報である。「本体」部分がメッセージの機能に応じた命令やデータの列である。「clear 命令」はメッセージのトレーラに相当する命令である。これらは、先頭のバケットから順にセルの組み込み論理部分で処理され、必要に応じて本体内のデータが受信側オブジェクトに引き渡される。

### 3.3 セル

2.2で述べたように、提案する再構成可能論理素子は、内部に組み込み論理部と可変論理部を持つセルが二次元のメッシュ構造に並んだ構造である。メッシュ構造のセル空間内では、各セルの絶対的なアドレス付けは行わず、セル間の経路設定がアドレスの代わりとなる。このようなセル構造は幾つかの利点を持つ。まず、均質なセルアレイ構造は、VLSIの製造技術を考えて場合に高集積化に有利である。また、チップ内で絶対的なアドレス付けを行わないので、セル空間の拡大が容易である。アドレス付け、論理の構成共に製造後に行なうので、製造過程での欠陥に対して製造後に対処可能である。そのため、大きなダイサイズのチップを作ることも可能になる。上記のことから、複雑な処理の実現に必要な大きなセル空間を得ることが可能である。

#### 3.3.1 組み込み論理部分

組み込み論理部分は、メッセージの転送、経路設定や、可変論理部分の構成等の組み込み機能を実行する。この部分を単純なプロセッサとして設計した例について述べる。つまり、これらの機能は組み込み論理部分に命令を与えることで実行させる。この組み込み論理部分への命令は、メッセージ中にバケットとして埋め込まれ、メッセージの転送機能により該当するセルの組み込み論理部分に渡される。命令長は5ビット

であり、このうち最上位ビットはメッセージのトレーラである clear 命令と他の命令、データとの区別のためだけに使用されるので、実質の命令長は4ビットである。

組み込み論理部分は、メッセージを受けるポートを持ち、ここに送られるバケットをデコードして制御を行なう。例えば経路設定を行なう場合ならば、メッセージ中の routing 命令を受けて、clear 命令で設定の解除が行なわれるまでの入力バケットの転送先を設定する。ただし、各セルが隣接セルに対応した4つのポートを持ち、全ての方向からのメッセージを受け、転送先が競合しない限り独立に転送を行なう。この機構で、セル上で複数のメッセージが交差できる。これによりメッセージのデッドロックが回避できる(図5)。また、これら以外に、可変論理部分とのインターフェースとなるポートを設け、オブジェクトとのメッセージの授受を行なう。

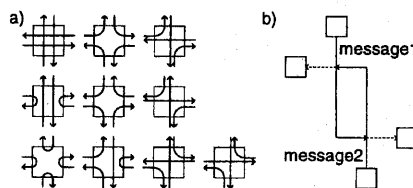


図5: メッセージの a) 交差 b) デッドロック

#### 3.3.2 可変論理部分

可変論理部分は回路部品に構成することのできるプログラムブル論理素子である。回路部品は、3入力1出力の組合せ論理か、1ビットのレジスタ、更に隣接セルの間を任意の組合せで接続する結線である。この機能を実現するため、可変論理部分は図6のような基本構造とし、論理部分にはLUTを用い、結線部分にはスイッチングマトリクスを用いる。可変論理部分は、これらの回路部品としての設定情報を保持する構成用メモリを持つ。組み込み論理部分はこの部分を read/write 可能なRAMとして扱う。構成用メモリは、回路部品の構成と切り離して任意のデータの格納にも利用する。

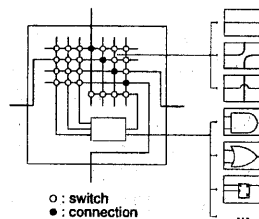


図6: 可変論理部分(回路部品)

## 4 組み込み論理部分の設計

PCA 上で、2.1 に述べたオブジェクトとメッセージを用いた処理の実現には、

- メッセージの伝達
- オブジェクトの操作
- セル空間の管理

の機能が必要である。このうち、最も基本的な機能を組み込み論理部分が提供する。組み込み論理部分を、プロセッサとして設計することで、これらの機能を実現する命令セットを定めることが組み込み論理部分の設計を行なうこととなる。

多くの機能を組み込み論理部分に任せると、組み込み論理部分が複雑になる。オブジェクトを構築するための可変論理部分の領域を大きくとるために、組み込み論理部分は極力小規模に抑えなくてはならない。また、制御に必要な命令数の増加はメッセージを長くし、メッセージ伝達のオーバーヘッドを大きくする。従って、基本機能の絞り込みと、それを実現するコンパクトな命令セットの決定が重要である。組み込み論理部分の複雑化を招く機能は、上位層である可変論理部分で構成する機能オブジェクトで実現させる。

以上の理由から、上述の機能のうちセル空間の管理を組み込み論理部分の機能から除外した。セル空間の管理機能は、並列分散的なオブジェクトの操作に必要である。しかし、この機能には複雑な処理が必要となるため、可変論理層でセル空間管理用のオブジェクトを複数個設けて実現することとした。本節では、これ以外の二つの機能の実現方法について述べる。

### 4.1 メッセージの伝達

メッセージの経路設定は routing 命令で行なう。組み込み論理部はメッセージ先頭の各 routing 命令を受け、後続するパケットを転送する方向を設定する。経路上のセルは routing 命令列を受理後、それを消費し、続くメッセージのみが目的とするセルに渡される。図 7 に、セル A からセル B への経路と、それを設定する routing 命令列を示した。設定された経路は clear 命令によって解除されるまでメッセージを転送する。更に、可変論理部分を転送先にする事でオブジェクトへのメッセージ伝達を実現する。従って routing 命令は 5 種類 (4 隣接セル + 可変論理部分方向) になる。

経路上のメッセージ転送には  $2\tau$  パイプライン機構を用いる (図 8)。経路上の組み込み論理部分は、転送先セルの入力ポートにパケットがないことを確認してメッセージの転送を行なう。この機構により、経路を共有するメッセージは先行するメッセージの通過を待ち、メッセージの衝突を回避することができる。

### 4.2 オブジェクトの操作

可変論理部分を構成する機構は、可変論理部分の構成を保持する構成用メモリに構成データを書き込むこ

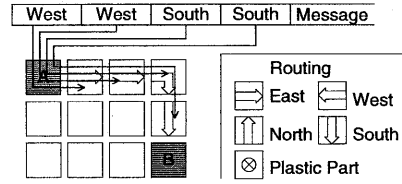


図 7: セル A からセル B への経路設定

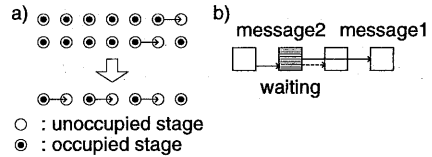


図 8: a)  $2\tau$  パイプライン b) メッセージの待ち

とで実現する。組み込み論理部分は configure-in 命令を受け、引続き与えられる一定個数のパケットをデータとして順に構成用メモリに書き込む。つまり、一続きの configure-in 命令と構成データのバケット列で構成するメッセージ片で一つのセルを構成する。更に、routing 命令とこのメッセージ片の組み合わせで、任意のオブジェクトを生成することができる (図 9-a)。この際に回路部品としての構成データを書き込めば、機能オブジェクトの構成ができる。任意のデータを書き込めば、メモリオブジェクトへのデータ書き込みとなる。

一方、組み込み論理部は configure-out 命令を受け、routing 命令を受けてメッセージの転送先を設定し、その後 clear 命令を受けると構成メモリ中のデータを設定された経路に送出する。つまり、一続きの configure-out、routing、clear 命令列で、セルの構成メモリ内のデータの読み出しを行なう。更に、routing 命令とこの命令列を組み合わせることで、任意の形状で配置されたメモリオブジェクトからデータを読み出すことができる (図 9-b)。

上述のオブジェクトの構成データ読み出しメッセージと構成データ書き込みメッセージを組み合わせ、オブジェクトのコピー、移動等の動作が実現できる。

構成後のオブジェクトの起動、停止は、カプセル化されたオブジェクトのインターフェースとなり、セルの組み込み論理部分に可変論理部分とのポートに各々 open、close 命令を送ることで実現する。構成したオブジェクトは、セル空間管理用のオブジェクトの管理下に置かれ、その消去はセル空間の管理機能で実現する。

## 5 まとめと考察

本稿では、動的に布線論理の構成を変えることのできる可能な計算機アーキテクチャである、PCA の実現仕様

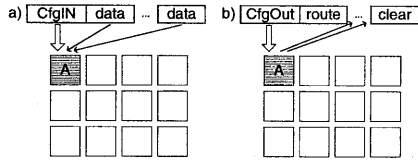


図9: 構成データの a) 書き込み b) 読み出し

を述べた。PCA と他の再構成可能計算機アーキテクチャとの違いは、プログラム内蔵型計算機におけるメモリ上の動的データをハードウェア上の動的布線論理オブジェクトとして実現することに着目している点である。この機能を並列オブジェクト指向的な方法で実現することにより、PCA では複雑な処理の布線論理によるプログラミングと、ハードウェア資源を効率的に使用した粒度可変な並列処理が可能である [7]。このアーキテクチャの本質は、均質な細粒度のセル構造とセルの機能の二重化である。そして、この機構の実現の鍵はセルの組み込み論理部分の設計にある。

組み込み機能の設計/評価は、PCA の設計のために作られた機能検証用のシミュレータを用いて行なっている [8]。これは、C 言語記述による組み込み論理部分と可変論理部分の機能をシミュレーションするものである。このシミュレータを用いて、メッセージ伝達、オブジェクトの生成等を並列分散的に行なう組み込み機能を設計した。シミュレーションにより、組み込み論理部分を小規模なプロセッサとして実現できるところまで機能を絞り込み、命令セットの単純化を行なった。

組み込み論理部分を小規模回路で実現することは、重要な課題である。現在、シミュレーションを用いた機能設計を元に、ハードウェア記述言語による組み込み論理部分の論理設計を行なっている。ハードウェア記述言語には SFL 言語を用い、論理シミュレータ、論理合成系には PARTHENON システムを使用している [9]。セルの組み込み論理部分は回路として 2000 ゲート程度を目標にしている。組み込み論理部分の回路規模に応じて、セル内の可変論理部分の設計は調整が必要である。セル内の組み込み論理部分と可変論理部分のバランスを考えると、組み込み論理部分は、可変論理部分と同等かそれより小規模であるべきである。そのため、組み込み論理部分の回路規模に応じて、セル内には図 6 の構造を単位に可変論理部分を複数個 ( $4 \times 4 = 16$  程度) とする構成についても検討している。

今後は、設計の詳細化と並行して、動的な再構成機能が有効な基本的なアプリケーションによる性能評価を行なう。このアプリケーションとして、動的に検索キーの更新を行うハードウェアパターンマッチエンジンを考えている [10]。これは、簡単な布線論理で実現できるパターンマッチモジュールを複数個持ち、与えられたストリームに対して、モジュール毎に異なるパターンの検索を並列に実行するものである。このアプリケーションを評価し、再構成のオーバーヘッド等、

本アーキテクチャ固有の問題の評価を行なう。

## 参考文献

- [1] Becker, J., Hartenstein, R., Herz, M. and Nageldinger, U.: Parallelization in Co-Compilation for Configurable Accelerators, in *Proc. of the Asia and South Pacific Design Automation Conference*, pp. 23–33 (1998).
- [2] Bittner, R. and Athanas, P.: Wormhole Run-time Reconfiguration, in *ACM/FPGA Conference*, pp. 79–85 (1997).
- [3] Hauser, J. and Wawrzynek, J.: Garp: A MIPS Processor with a Reconfigurable Coprocessor, in *Proc. of the IEEE Symposium on FPGAs for Custom Computing Machines*, pp. 98–105 (1997).
- [4] Schmit, H.: Incremental Reconfiguration for Pipelined Applications, in *Proc. of the IEEE Symposium on FPGAs for Custom Computing Machines*, pp. 47–55 (1998).
- [5] Nagami, K., Oguri, K., Shiozawa, T., Ito, H. and Konishi, R.: Plastic Cell Architecture: Towards Reconfigurable Computing for General-Purpose, in *The IEEE Symposium on FPGAs for Custom Computing Machines* (1998), To be published.
- [6] 中田広, 小栗清, Imlig, N., 塩澤恒道, 稲森稔: 動的再構成可能な VLSI プロセッサとそのレイアウト手法, コンピュータシステム研究会電子情報通信学会 (1998).
- [7] 塩澤恒道, 小栗清, 永見康一, 伊藤秀之, 小西隆介: 汎用計算機構を実現する再構成可能 LSI アーキテクチャ, 集積回路 & VLSI 設計技術合同研究会, pp. 9–15 電子情報通信処理学会 (1998).
- [8] 小西隆介, 伊藤秀之, Imlig, N., 小栗清: プラスチックセルアーキテクチャの一実現仕様と機能シミュレータについて, パルテノン研究会予稿集パルテノン研究会 (1998).
- [9] Nakamura, Y., Oguri, K. and Nagoya, A.: Synthesis From Pure Behavioral Descriptions, in Camposano, R. and Wolf, W. eds., *High-Level VLSI Synthesis*, pp. 205–229, Kluwer Academic Publishers (1991).
- [10] Oguri, K., Imlig, N., Ito, H., Nagami, K., Konishi, R. and Shiozawa, T.: General Purpose Computer Architecture Based on Fully Programmable Logic, in *International Conference on Evolvable Systems* (1998), Submitted.