設 計 自 動 化 87－3
（1998. 1. 30）

自律再構成可能アーキテクチャ

永見 康一　　塩澤 恒道　　小栗 清
NTT 光ネットワークシステム研究所
〒 239 神奈川県横須賀市光の丘 1-1

概要

ハードウェア記述言語 (HDL) とプログラマブル論理デバイス (PLD) の技術を応用して, 非ノイマン型の汎用計算環境の実現を目指す. そのために, 1) 動作記述型の HDL を, モジュールの動的インスタンス化セマンティクスにより拡張し, 2) PLD に, 自律再構成能力と呼ぶ新たな再構成能力を付加して, このセマンティクスの直接的な実行を実現する, というアプローチをとる. 本稿では, このアプローチについて触れ, 自律再構成能力について考察する. さらに, 自律再構成可能なデバイスのアーキテクチャ規範として, プラスティック・セル アーキテクチャ (PCA) を提示する. PCA は PLD とセルラ・アーキテクチャを融合したアーキテクチャであり, 汎用計算が求める不均質要素の活動を, 均質な物理構造によって実現することを狙っている.

キーワード: 再構成可能計算, FPGA, オブジェクト指向, ハードウェア記述言語, セルラ・オートマトン

Autonomously Reconfigurable Architecture

Kouichi NAGAMI　　Tsunemichi SHIOZAWA　　Kiyoshi OGURI
NTT Optical Network Systems Laboratories
1-1 Hikarinooka, Yokosuka-shi, Kanagawa 239, Japan
Email {nagami, shiozawa, oguri}@exa.onlab.ntt.co.jp

abstract

We have addressed a research direction towards purely-wired but general-purpose computing environment, which will be realized with progresses of the two technologies: HDL and PLD. We approach to the goal by 1) enhancing an existing behavioral HDL with the semantics of dynamic module instantiation, 2) and inventing a new PLD with *autonomous reconfigurability*, as a platform for direct execution of the semantics. In this paper we discuss the direction and approach, followed by discussions on the autonomous reconfigurability. Then we propose a new architectural reference called *Plastic Cell Architecture* (PCA), which simply models devices with the reconfigurability. PCA is a fusion of PLD and cellular architecture, which aims at providing a physically uniform stage for the activities of variable-grained computing elements, which are indispensable in general-purpose computing.

keywords: reconfigurable computing, FPGA, Object-Oriented, Hardware Description Language, cellular automata

1 はじめに

マイクロプロセッサと RAM (Randam Access Memory) によって構成されるノイマン型計算機は，その機能実現の柔軟性によって計算機アーキテクチャ進展の中心となってきた．しかしこの柔軟性は，逐次実行およびメモリ・ボトルネックという制約によって成り立っているため，ノイマン型計算機は専用布線論理と比較して速度性能が劣る．マイクロプロセッサの MMX (Multi Media eXtension) 技術や DSP (Digital Signal Processing) コプロセッサに見られるように，この問題に対するブレークスルーが，汎用目的の計算機システムにさえ求められている．

一方で，布線論理による機能実装が汎用ではなく専用目的に限定されてしまうのは，1) 機能の記述が困難，2) 機能の実現に半導体実装が伴う，3) 構造が静的であるため動的な要素を実現するのが困難，という三つの理由による．

PARTHENON [5] は，論理設計の分野にハードウェア記述言語 (Hardware Description Language; HDL) を導入した．これは，それまで構造記述しか手段のなかった布線論理の記述に，より人間の直観に近い動作記述という手段をもたらした．一方，半導体装置の機能変更可能性を，製造工場から設計現場に移すことができる技術として，FPGA (Field Programmable Gate Array) などのプログラマブル論理デバイス (Programmable Logic Device; PLD) が登場している．これら二つの技術は，上記 1), 2) を打開し，「布線論理による汎用計算環境」という共通の方向を指向しているとみなせる．

そこで我々は，HDL と PLD の技術を発展させることにより，OS (Operating System) からアプリケーションに渡る機能を，全て布線論理としてプログラミングでき，かつ布線論理として実行できる汎用計算機環境の実現を目指す．しかし，布線論理による機能実装にとっては上記 3) が以前課題として残っている．端的にいうならば，標準 C 言語のライブラリ関数 `malloc/free` や C++, Java 言語の `new/delete` 演算子が持つ，要素の動的生成・消滅セマンティクスは，記述手段も実行手段も与えられていない．

動的セマンティクスの記述手段に関しては，上記 PARTHENON が導入した HDL である SFL [1] のオブジェクト指向拡張により対応する．SFL は，モジュール化 HDL が一般に提供する構造化プログラミングパラダイムに加え，オブジェクト指向が提唱するカプセル化およびメッセージ・パッシングという概念を自然に備えている．これは主に制御信号という文法要素によって実現されている．この言語を自然に拡張して，動的セマンティクスの記述を，モジュールの動的インスタンス化として実現する．すなわち，部分回路のインスタンスをオブジェクトみなし，従来静的にしか存在し得なかったこのインスタンスを `new/delete` できるような言語を設計する．

一方，このモジュール・インスタンスの動的な生成・消滅セマンティクスの実行手段として PLD を用いることを考えると，1) 動作時の再構成能力 2) 部分的な再構成能力 3) デバイスによる自己再構成能力，の三つの能力を同時に実現することが必須である．現在の技術動向として，1), 2) については，SRAM (Static RAM) ベースの FPGA として製品レベルに達している [7]．そこで，この技術を踏襲し 3) の実現を目指す．この 3) の能力，すなわちデバイス上にプログラムされた論理回路 (オブジェクト) が，その論理回路的動作によって直接的にデバイス上に別の論理回路をプログラムできる能力を，本稿では自律再構成能力とよぶ．

以下，2 節で自律再構成能力の実現について考察する．その考察に基き，3 節で自律再構成可能なデバイスのアーキテクチャ規範となるプラスティック・セル アーキテクチャというモデルを導入する．最後にまとめと今後の研究方針について 4 節で述べる．

2 自律再構成能力の実現

2.1 デバイスに要求される性質

前述したように，動作時再構成と部分的再構成の能力を実現することを考え，SRAM ベースの FPGA を前提にする．すなわち，デバイスの物理構造は均質なセルのアレイ構造とし，各セルは然るべきビット数の SRAM からなる構成情報記憶を持つ．そして，各セルは，構成情報記憶の値を設定することにより，あらかじめ定められた論理プリミティブ・セットのうちの一つの機能を果たすように構成できる．

自律再構成を考える時，セル・アレイが実行時資源であること，すなわちセルの領域の使用状況をデバイス自身が動作時に管理しなければならないことを考慮に入れなければならない．したがってセルの物理的構造は一様であることが望ましく，かつなるべくシンプルであることが要求される．これを考慮し，デバイスはセルをまたがる配線のための専用資源を持

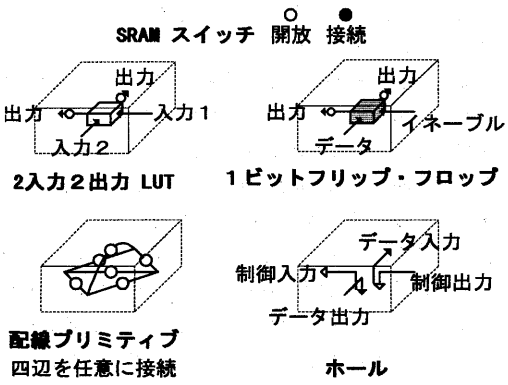


図 1: 論理プリミティブ・セットの例

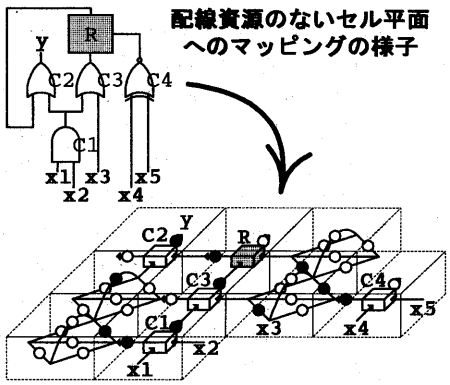


図 2: 配線資源を用いないオブジェクトの構成

たない，という制限を置く．その代わりに，論理プリミティブの中に配線要素を含める．すなわち，セルをまたがる配線も，セルの構成によって実現する．図 1 に配線要素を備えた論理プリミティブ・セットの一例を示し，図 2 に，その論理プリミティブ・セットを用いて，配線資源を持たないデバイス上に構成されたオブジェクトの例を示す．

図 1 は，配線要素，LUT (Look-Up Table)，フリップフロップの他に，ホールとよぶ論理プリミティブを示している．このホールは，オブジェクトの外部出力信号と，以降で述べる組み込みの物理機構とを結合するためのものである．すなわち，ホールを持つオブジェクトは，そのホールを介した信号交信によって，デバイスの組み込み機能と相互作用することができる．自律再構成可能なデバイスを実現するためには，このホールに相当する結合要素を，論理プリミティブの中に必ず含めなければならない．次節では，この組み込み機能の概念について述べる．

2.2 組み込み機能

一様なセル・アレイは，オブジェクトを動的に構成するための二次元的な空間となる．RAM が動的データを配置するための一次元的な空間であるのと同様である．しかし，RAM が CPU 無しにはデータ構造を構築できないのと同様に，セル・アレイのみでは，オブジェクトの能動動作によるオブジェクトの構成や，オブジェクト間のメッセージ交換などの動的セマンティクスの実行は不可能である．すなわち自律再構成可能デバイスは，再構成可能なセル・アレイに加えて，動的オブジェクトの活動を支えるための種々の組み込み機能を備えなければならない．

2.2.1 セルラー・オートマトンによる実装

一様平面上に特定のアルゴリズムを実装するのに適しているアーキテクチャとして，セルラー・オートマトン (Cellular Automata; 以下 CA) がある．したがって，一様なセル・アレイの構造を自然に利用して組み込み機能も実現するために，以下のようにしてこのアーキテクチャを応用する．

- 各セルは，論理プリミティブを実現するリソースに加えて，オートマトンを含む．
- オートマトンは隣接間が接続され，一様な結合網をなす．
- ホール・プリミティブは，論理プリミティブのアレイと CA とを接続し得る．

この実現方法は，図 3 に示すように，再構成論理のための平面と CA 平面という，概念的な二つの平面によって表すことができる．

デバイスに，どのような組み込み機能集合を持たせるかという判断は，一概には定められない設計問題であるが，以下に述べるメッセージ伝達機能，オブジェクトの生成および消滅機能は，どのような設計においても必須である．

メッセージ伝達組み込み機能 動的セマンティクスにおいては，オブジェクトの配置は動作時に決定される．すなわち信号伝達というメッセージ・パッシングによって互いに通信し合う複数のオブジェクトについて，それらの物理的距離や位置関係がコンパイル (合成) 時に決定できないため，メッセージの伝達

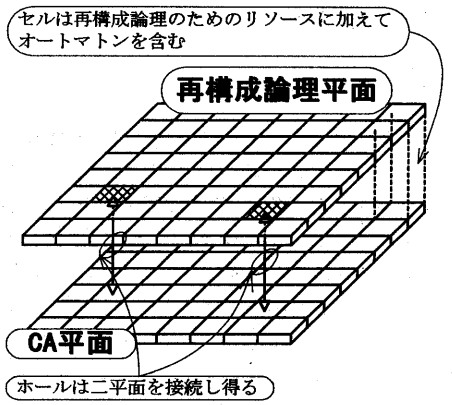


図 3: 自律再構成能力のための概念的な二平面

経路となる配線経路および遅延時間が静的に解決できない．つまり，動的オブジェクト間のメッセージ・パッシングを，固定的な配線によって実現することには無理がある．メッセージ伝達組み込み機能は，この問題を解決するためのアルゴリズムである (図 4).

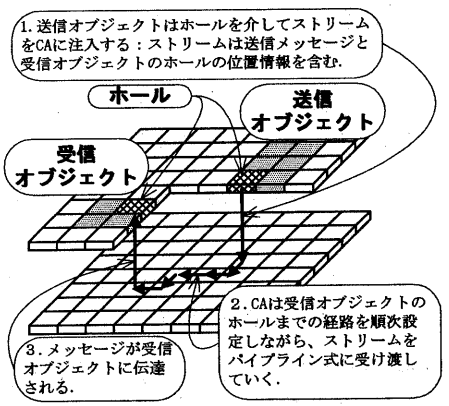


図 4: メッセージ伝達組み込み機能

この機能は，ストリーム (信号系列) のパイプライン式自己ルーティングを実現する．メッセージ送信オブジェクトは，ホールを通じて送信ストリームを CA にビットシリアル的に印加していく (注入). ただしストリームは，送信メッセージに加え，受信オブジェクトのホールの位置に関する何らかの情報を含んでいるとする．ストリームはセルからセルへ順次パイプライン式に受け渡されていき，その結果 CA は，送信ホールから受信ホールへとストリームを経路付ける．経路判断は受信ホールの位置情報による．これにより，送信オブジェクトから受信オブジェクトへメッセージが伝達される．

各セルが一意なアドレスを持っていることを前提にとした場合の，この機能を果たす CA についての詳細な検討が報告されている [4].

生成・消滅組み込み機能 オブジェクトを生成する動作も，セル領域の各所で並列に実現できる方が好ましい．しかしながら，オブジェクトを配置するためのセル領域は共有資源であるため，生成動作には排他制御が必要である．しかし，生成動作を，領域確保動作と構成情報記憶書き込み動作とに分けて考えれば，排他制御が必要なのは領域確保動作だけである．

消滅は，領域解放動作だけか，あるいは領域解放と構成情報記憶のクリアとの組合せによって実現される．いずれしろ，生成・消滅組み込み機能の実現は，次の二つの独立した組み込み機能に整理できる．

領域確保・解放組み込み機能 各セル内のオートマトンは，セルが使用されているかどうかを内部状態によって常に保持している．オブジェクトを生成しようとするオブジェクトは，新しく生成するオブジェクトが占める領域の形状情報を，ホールを通じて CA に挿入する．CA は指定された形状の領域を未使用領域の中から排他的に探索する．指定形状の領域が見つかったら，その領域内の全セルが全て使用状態に遷移した後に，その領域の位置を表す情報がホールを通じてオブジェクトに返される．また，使用状態にあるセルに対して解放を表す信号が挿入されると，そのセルを含む使用中の領域内の全セルが未使用状態に遷移する (図 5).

オブジェクトの形状は必ず矩形であるという前提をおいても，オブジェクトのコンパイルの観点からも領域使用効率の観点からも，さしたる制約にはならない反面，CA の複雑度は大幅に軽減できることが予測できる．

構成情報充填組み込み機能 いったんセル領域が確保された後に，その領域内の各セルの構成情報記憶の値を設定することによりオブジェクトが生成される．つまり構成情報充填組み込み機能は，オブジェクトを表現する構成情報の集合を，平面的に配置するための機能である．

この機能に対する直観的なアプローチとして，Wormhole Run-Time 再構成 [6] と呼ばれる技術が応用できる．すなわち，オブジェクトを生成する側のオブジェクトは，ホールを介して CA にストリームを注入する．ストリームは，生成されるオブジェク

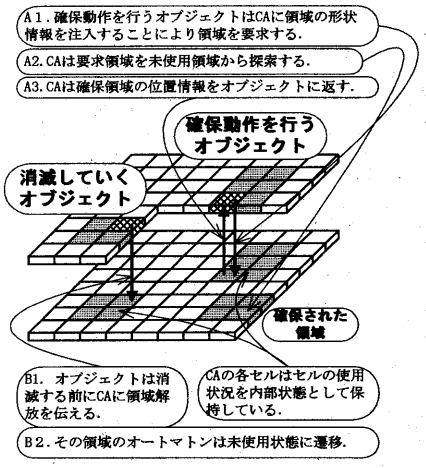


図 5: 領域確保・解放組み込み機能

トの各セルの構成情報と, 命令語との混合である. ここでいう命令とは, ストリームの移動方向の制御と, セルへの構成情報書き込み契機を表現するものである. CA は, 命令にしたがいストリームの経路制御と, SRAM への値設定を行なう.

3 プラスティック・セル アーキテクチャ

前節の議論により, 自律再構成可能デバイスが備える以下の性質が導かれる. 1) 空間的に一様な物理構造を持つ. 2) オブジェクトを構成するための資源に加えて, オブジェクトの活動を支えるための組込み機能を持つ. 3) 組込み機能は, CA により実現される. 4) オブジェクトと組み込み機能は, 論理回路的な動作によって相互に作用できる.

これらの諸性質は, 次に述べるプラスティック・セル アーキテクチャ (Plastic Cell Architecture; PCA) としてまとめることができる.

3.1 PCA の構造

PCA は, 図 6 が示すような構造を持っている.

まず, 物理構造は同一のセルを平面的に敷き詰めた一様なアレイ構造である. そして各々のセルは, 組み込み部およびプラスティック部と呼ぶ二つの部分によって構成される.

組み込み部およびプラスティック部は共に, 隣接間が接続されており, それぞれ組み込み結合網およびプラスティック結合網と呼ぶ二つの結合網を構成する. 組み込み結合網は, CA として動作し, あらかじめ定められた組み込み機能を提供する. 各々のプラスティック部は, 構成情報記憶素子を持ちその値によってあらかじめ定められた論理素子集合のうちの一つとしてプログラムすることができる. またプラスティック結合網は動作時に部分的に再構成可能であり, オブジェクトを配置するための空間となる.

単一セル内の組み込み部とプラスティック部とは, 相互に作用を及ぼすことができる. すなわち, 組み込み部はプラスティック部の構成情報記憶にアクセスすることができ, 一方プラスティック部は組み込み機能部に対して信号を印加できる.

このモデルは, 動作時部分書き換え可能な SRAM ベース FPGA とセルラオートマトンとの融合と捉えることができる.

3.2 既存アーキテクチャとの比較

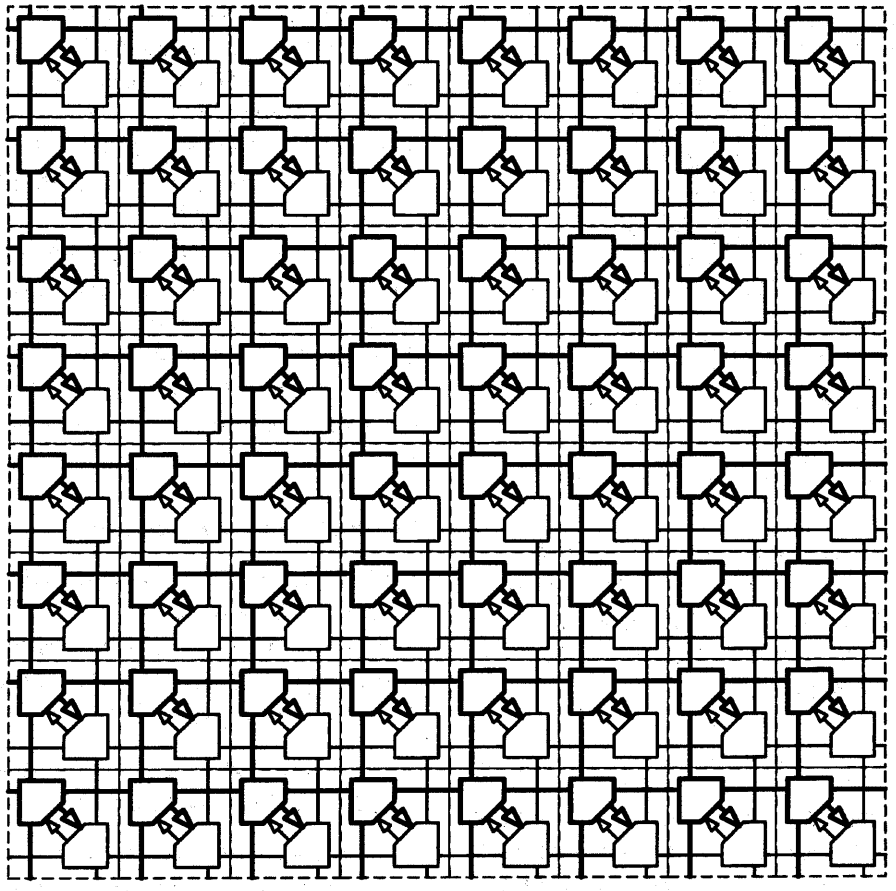
PCA の特徴を把握するために, 種々の既存アーキテクチャとの比較を述べる.

ノイマン アーキテクチャ ノイマン計算機は, CPU と RAM によって構成される. これは, PCA を次のように変形したものと見ることができる (図 7).

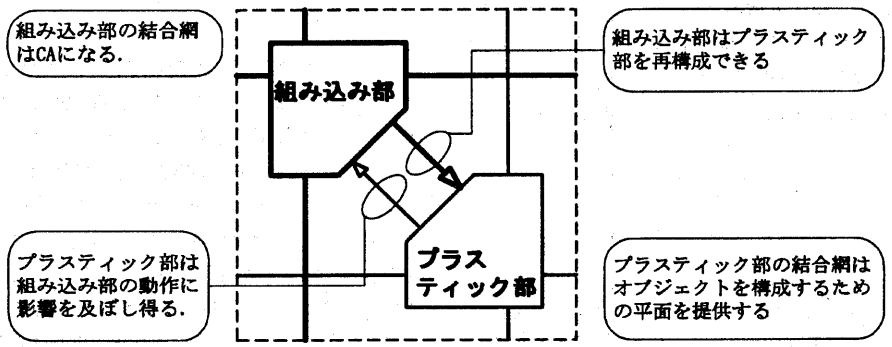
- 組み込み結合網が, 唯一かつ単一の能動計算要素である CPU に凝集されている.
- プラスティック結合網は RAM となり, 能動的な計算要素を組織する能力を失い, CPU が用いるワード・データを受動的に記録する機能に専念している.

これにより, PCA と比較してノイマン型は, 1) システム内の能動的な計算要素が, 固定的な粗粒度であり, 2) 計算実行中の可変サイズ要素は, RAM 上の受動的データ構造により間接的に実現される, という性質を持つ. マルチプロセッサ・システムによる粒度調整は, 1) への対策と見ることができ, 一方で機能メモリ技術は 2) への対策と見ることができる.

セルラ・アーキテクチャ セルラ・アーキテクチャとしては, CA およびシストリック・アレイが代表的であるが, これらは PCA からプラスティック結合網を取り去ったものである. この結合の欠落により, このアーキテクチャは可変サイズの能動計算要素を実現することができない. このことにより, セルラ・アーキテクチャは, 汎用計算ではなく特定のアルゴリズムの実現に適している.



(1) PCA はセルの一様な配列構造によって構成される.



(2) 各セルは, 組み込み部およびプラスティック部の二部からなる.

図 6: 自律再構成可能デバイスの設計規範: プラスティック・セル アーキテクチャ

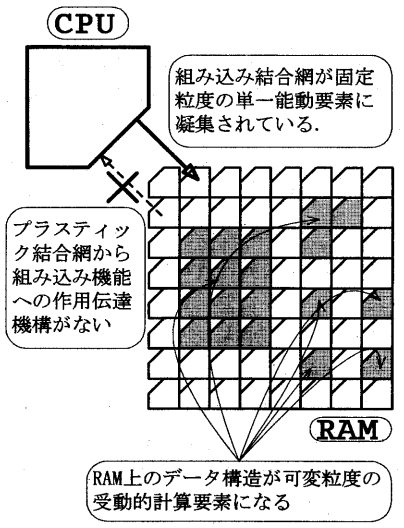


図 7: PCA とノイマン型との対比

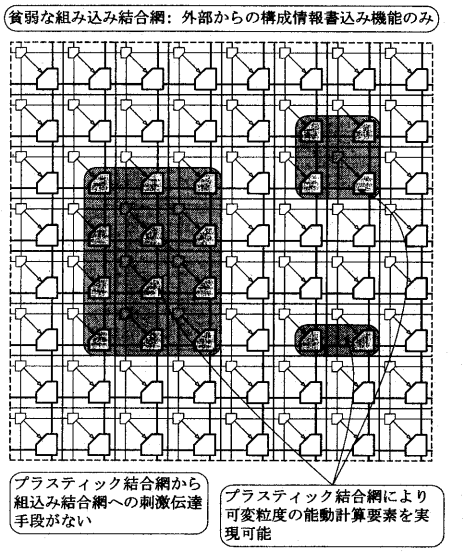


図 8: SRAM ベース FPGA と PCA との対比

SRAM ベース FPGA 一方で, 既存の SRAM ベース FPGA を PCA の図式で説明すると, 図 8 の様になる. この図が示すように, 動作時部分書き換え可能な FPGA は, 動作時に粒度が可変な能動計算要素を実現できる. ノイマン型およびセルラ・アーキテクチャでは, 能動計算要素は固定粒度であったことを考えると, この要素は FPGA がもたらした重要な新規性である.

しかし現状の FPGA は, 構成情報の書き込み機能に限定された貧弱な組み込み結合網しか持たず, またプラスティック部から組み込み部への作用を実現できない. したがって再構成を制御するためには, デバイス外部の能動計算要素が必要になる.

以上の比較により, PCA の特長は次のようにまとめられる.

1. 布線論理によって汎用計算を直接的に実現するために必要な, 可変粒度の能動回路 (オブジェクト) を実現することができる.

2. 構成情報書き込み機能に限定されない種々の組み込み機能と, プラスティック結合網からそれら組み込み機能へ作用できる能力によって, 上述の粒度可変性をプログラムすることができる.

3. 加えて, 汎用計算のプラットフォームを物理的にスケーラブルな構造によって実現できる.

3 については, 次に述べるような効果が期待できる.

3.3 一様構造による利点

半導体技術の進歩により, デバイス上に実装できる論理の規模は増加し続けている. この進歩により, 複雑なディジタルシステム全体を単一チップ上に実現することが可能になってきているが, 一方ではそのような複雑なシステムをどのようにして設計するかという新たな問題が発生しつつある.

この問題に対して, ノイマン型計算機に基づいた詳細なアプローチが報告されている [3]. 一方で PCA は, この問題に対するもう一つのアプローチを示している. すなわち, PCA が持つ物理的に一様な構造は, 利用可能なトランジスタの増加に対し, セル・アレイのサイズを単純に拡張することによって対応することを可能にする. このときデバイスの設計は単一セルが持つ複雑度しか要求せず, これはデバイス全体に複雑なシステムを設計することの複雑度に比べてはるかに小さい. 言い替えると, システム設計の複雑さは, アプリケーション・プログラミングという形で, 容易に反復修正可能な作業に委ねることができる.

3.4 実装の方針

PCA に基づくデバイスの設計にあたっては, 大きく分けて次の二つの作業が必要になる.

プラスティック部の設計 は, オブジェクトを構成するのに必要十分な論理プリミティブの集合を決定す

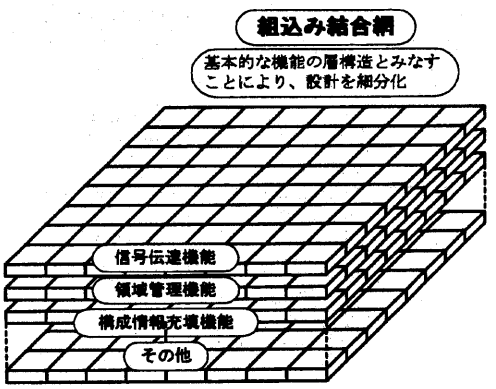


図 9: 積層指向戦略

ることから始まる. この決定は, マイクロ・プロセッサの命令セット設計に相当している. すなわち, オブジェクトの合成処理系の複雑度や, そのデバイスがどのようなアプリケーション分野を指向するかなどに影響する. また, 組み込み部との論理量的なバランスを考慮して決定しなければならない.

組み込み部の設計 は, オブジェクトの活動を支えるのに必要十分な機能集合を, セルラ・プログラミングによって実装する作業である. 組み込み部の設計方針として, 次に示す二つの戦略が考えられる.

プロセッサ指向戦略 この戦略では, 組み込み部のオートマトンを, プロセッサとみなして設計する. すなわち, 各組み込み機能を命令として表現し, それらの命令を解釈実行するプロセッサを設計する.

積層指向戦略 PCA を実装していく場合, 必要な基本的組み込み機能を数え挙げていくことになる. この戦略は, これらの各基本機能をそれぞれ単一のセルラ・アルゴリズムとして設計していく方針をとる. 結果として組み込み結合網は, 概念上複数の CA 平面が積層された構造として設計される (図 9).

この戦略では, 各セルラ・アルゴリズムを独立に設計・検証することが可能である. さらに, 将来的に三次元 VLSI 実装 [2] が現実的になった場合でも, 概念的な積層構造を直接的に物理構造とすることによって, 自然にその技術を利用できる. また, 単純な基本機能には速いクロックを印加し, 逆に複雑な機能には遅いクロックを印加するというような, 複数クロックによる性能の向上の可能性がある. このような理由により, この戦略はプロセッサ戦略に比べてより複雑な組み込み機能の設計に向いている.

4 おわりに

本稿では, 非ノイマン型の汎用計算環境として, プログラミングおよび計算の実行の双方に布線論理を直接取り込むアプローチを示した. さらに, その実現のために PLD に加えなければならない自律再構成能力について考察した. この能力は, オブジェクトを構成するための SRAM ベース FPGA と, オブジェクトの活動のための組み込み機能を実現する CA とを融合したデバイスによって実現される. そこで, この融合を表現する PCA というデバイスのアーキテクチャ規範を提案した.

現在, PCA に基づくデバイスのプロトタイプ設計が進行中である. 今後はこれと並行して, 動的セマンティクスを持つ布線論理の記述例の例示, およびデバイス設計支援のための CA 設計手法確立に取り組んでいくつもりである.

参考文献

[1] 小栗清. ハードウェア記述言語と高位論理合成システムに関する研究. 学位論文, 九州大学, 1997.

[2] 堀口進, 大木孝之, V.Jain. ウェーハスタック構造型・超並列コンピュータネットワーク:TESH. 研究報告 Vol.97, No.61 (97-ARC-124), pp. 37–42, 情報処理学会, 1997 年 6 月.

[3] Kazuaki Murakami, Koji Inoue, and Hiroshi Miyajima. Parallel Processing RAM (PPRAM). In *Proc. of Japan-Germany Forum on Information Technology*. November 1997.

[4] 中根良樹, 永見康一, 中村行宏. FPGA 上での動的信号伝達アルゴリズム. 第 11 回パルテノン研究会 資料集. パルテノン研究会, 1997 年 12 月.

[5] PARTHENON Home Page. http://www.kecl.ntt.co.jp/car/parthe/.

[6] Jr. Ray A. Bittner and Peter M. Athanas. Computing kernels implemented with a wormhole RTR CCM. In *Proc. of the Fifth IEEE Symposium on FPGAs for Custom Computing Machines*, pp. 98–105. IEEE Computer Society Press, April 1997.

[7] XILINX. XC6200 Field Programmable Gate Arrays, April 1997.