論　文

”圧力 ”の概念を利用した新しいPCAのための領域管理法

永本　太一[†a)]　矢野　智史[†]　内田　満[†]　高野　智明[†]
柴田裕一郎[†]　小栗　清[†]

New Area Management Method Based on “Pressure” for Plastic Cell Architecture

Taichi NAGAMOTO[†a)], Satoshi YANO[†], Mitsuru UCHIDA[†], Tomoaki TAKANO[†], Yuichiro SHIBATA[†], and Kiyoshi OGURI[†]

あらまし　「圧力」の概念を基礎とした動的再構成ハードウェア用の新しい領域管理法を提案した．PCA(Plastic Cell Architecture) は処理の本質と汎用性に着目して提案された動的再構成可能なデバイスである．PCA 上で動的再構成の特徴を用いて C 言語の malloc のような機能を実現するには領域を管理する機構が必要である．しかし PCA でそのような領域管理を実現する場合，管理と並列性は両立しないため「管理者」を用いて領域全体を管理することは難しい．問題はどのように空いている領域を確保するかである．本論文では，圧力によって隣のオブジェクトを移動させることで空き領域を動的に作る領域管理法を提案し，実装に必要な機構を述べ，３種類のコマンドセットを提案し，実行面積・有効利用率・オブジェクト間の距離について評価した．

キーワード　PCA, 動的再構成デバイス, 領域管理, 圧力, コマンドセット

1. まえがき

ユビキタス時代を迎え，フォンノイマン型コンピュータは生活に多大な影響を与え社会の隅々まで浸透している．フォンノイマン型コンピュータの発展の鍵はその汎用性と柔軟性にあり，ハードウェアとソフトウェアからなる構造が汎用性の，メモリ上のデータ表現やオブジェクトインスタンスを malloc や new 操作などによって動的に生成できるヒープ管理が柔軟性の源となっていると考えられる．

PCA（Plastic Cell Architecture）は，この汎用性と柔軟性に着目して提案された新しいコンピューティングアーキテクチャである [1], [2]．フォンノイマン型コンピュータがプログラム論理をベースとするのに対し，PCA は布線論理をベースとし，動作中の回路が新たな回路を生成して協調動作することができるという特徴がある（以降このことを動的再構成機能と呼ぶ）．PCA の最初の LSI （PCA-1）は NTT の研究グループにより製造され特徴的な機能のすべてが正しく動作することが確認された．同グループは性能向上を目指した二つ目の LSI である PCA-2 を製造している [3], [4]．我々は非同期ビットシリアル演算に特化することにより性能の向上をねらう Bit Serial PCA を提案している [5].

しかし，PCA の最大の特徴である動的再構成機能を積極的に使ったアプリケーションの研究 [6] はあまり活発に行われていない．その理由は，動的再構成機能を使うための malloc または new 操作のような領域管理法が未だ提供されていないためであると考えられる．

この問題に対する一つの解決法として PCA にフォンノイマンコンピュータの OS のような管理者をおいて領域を一元管理するという方法が考えられた [7]．これは PCA 上に領域管理回路を作り，それが空き領域の管理を行うという方法である．しかし多くの回路が同時に領域割り当て要求をこの領域管理回路に求めた場合に全体の性能がこの領域管理回路の性能で制限されてしまうという問題がある．あるいは，領域管理回路へのアクセスがデッドロックを起こしてしまう．すなわち，管理と並列性は両立しない．

そこで本論文では領域管理回路を使わない ”圧力 ” の概念を用いた新しい領域管理方法を提案する．これ

† 長崎大学，長崎市
Graduate School of Science and Technology, Nagasaki University Bunkyo 1–14, Nagasaki-shi, Nagasaki, 852–8521 Japan
a) E-mail: taichi@pca.cis.nagasaki-u.ac.jp

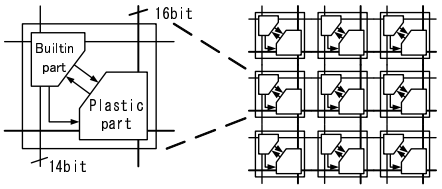


図 1　PCA セル
Fig. 1　PCA cell

は細胞分裂のように隣の細胞を押しのけて空き場所を作り，そこに新たな細胞（インスタンス）を作るようなイメージである．この動作は複数箇所で独立に行うことができ，領域を管理するという必要がまったくない．さらに，必要なインスタンスを隣に作るため関係するインスタンス間の距離が短くなると期待できる．このイメージを具体化するためにディジタルシステムとしての PCA の中に圧力や移動に対応する機能を作りこまなければならないが，PCA では PCA の特徴である動的再構成機能を組み込み部（2. 章を参照のこと）のコマンドセットとして実現しているのでこの圧力や移動もコマンドセットの拡張（圧力コマンドセット）として実現することとした．

本論文では，2. で PCA と Bit Serial PCA の紹介を行い，3. で圧力によって領域を確保する新しい領域管理法を提案し，4. で実現するために必要な機構を述べ，5. で作成した 3 種類のコマンドセットを紹介し，最後に 6. でそれぞれのコマンドセットの評価を行う．

2. PCA (Plastic Cell Architecture)

PCA はルーティングネットワークの中に小規模の真理値表である LUT（Look Up Table）が敷き詰められた構造を採っており，処理の本質である「データの加工・転送・保持」をメモリとしての LUT または回路としての LUT と，ルーティングネットワークにより直接かつ並列に行うことができる [1] [2].

PCA は基本要素となる PCA セル（図 1 左）を二次元メッシュ状（図 1 右）に並べたものである．PCA セルは，記憶機能や演算などを行う再構成可能な回路としての可変部（Plastic part）と，その可変部の制御と通信を行う組み込み部（Builtin part）からなっている．

図 2 はオブジェクト間通信の様子を表している．一つの回路として動作している PCA セル群をオブジェクトと呼び，オブジェクト間の通信は組み込み部を介して行われる．複数のオブジェクトが互いにメッセージを送りあい並列に動作するだけでなく，組み込み部に対して構成メッセージを送ることにより新しいオブジェクトを作り出すことができる．

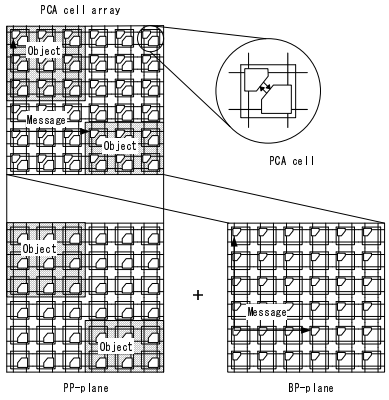


図 2　オブジェクト
Fig. 2　Object

PCA では新しいオブジェクトを接続する際のタイミング制御が容易な非同期回路方式が採用されている．

2.1 Bit Serial PCA

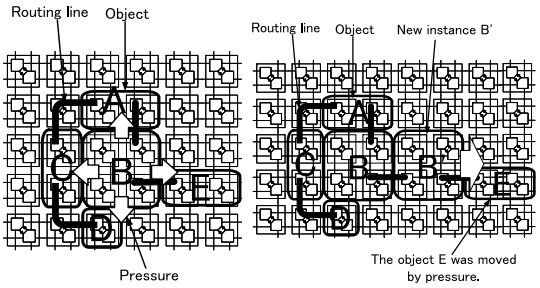
2.1.1 ビットシリアル

データ処理方式にはビットパラレル方式とビットシリアル方式がある．現在広く使われているのはビットパラレル方式であるが，信号間のスキューや配線遅延の割合が増えるとビットシリアル方式の方が高速になる可能性がある．加えて再構成可能ハードウェアでは，配線部分に再構成のためのスイッチが必ず存在し，配線遅延の割合がより大きいため，ビットシリアル方式の優位性が増すものと考えられる．また，PCA ではデータフローに沿って多くの演算回路を配置して任意の処理を行わせるため，機能あたりの回路が小さいビットシリアル方式がより適していると考えられる [5]. これらの理由により Bit Serial PCA が提案された．

2.1.2 シフトレジスタ＋ステートマシン構造

従来の PCA の構造（LUT 敷き詰め型）は自由度が高いことが利点であるが，配線を構成する場合にも LUT を使うため配線が長くなると遅延が大きくなり，集積度が低くなるという欠点がある．これまでの研究からビットシリアル方式の設計において，ビットシリアルデータパスはシフトレジスタとステートマシンに含めてしまえるような簡単な回路で構成できることがわかっている [8]. 記憶機能がシフトレジスタで良いならば，

$$[\text{可変部}] = [\text{記憶機能}] + [\text{加工機能}]$$



(a) When the object B needs a new object　(b) When the object B makes new object B'

図 3　提案手法のイメージ
Fig. 3　Idea image

$$= [\text{シフトレジスタ}] + [[\text{シフトレジスタ}] + [\text{ステートマシン}]] = [\text{シフトレジスタ}] + [\text{ステートマシン}]$$

となり，可変部をシフトレジスタとステートマシンで構成できると考えられる．この構成を用いることにより従来の LUT 敷き詰め方式よりも粒度をあげることができ，よりコンパクトかつ高速な可変部を作れることが期待できる．これも Bit Serial PCA が研究された理由の一つである．

3. 提 案 方 法

図 3 に提案方法の PCA におけるイメージを示す．図中の四角はオブジェクト（Object）を表し，太い線はデータ経路（Routing line）を表す．オブジェクトは 1 個以上の PCA セルで構成され，オブジェクト間の通信は組み込み部を介して行われる．

このイメージでは，オブジェクト B が新しいオブジェクト B' を必要な時，圧力コマンド（Pressuer）をまわりのオブジェクトへ送り，圧力コマンドは各オブジェクトの組み込み部で処理され，まわりのオブジェクトは隣が空いていれば移動し（図 3 の場合，オブジェクト E が移動した），オブジェクト B の隣に空き領域ができるとそこにオブジェクト B ’を作るという動作を示している．

最終的には PCA に組み込むとしても，圧力による領域管理という全く新しい手法の特徴や効果を明らかとするために本論文では以下のような単純化を行うこととした．

（1） オブジェクト間の経路や経路を設定する上での問題は考えない．

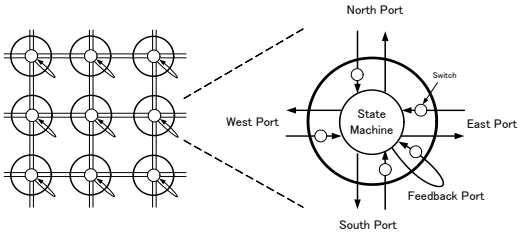


図 4　圧力コマンドセットのための機構
Fig. 4　The Mechanism for the pressure command set

（2） オブジェクトは 1 つの PCA セルから構成される．

圧力コマンドセットを実現するに当たって，様々なコマンドセットを考えることができるが，正しいコマンドセットを作り出すのは思いのほか難しい．そこで本論文では多くの試行錯誤の結果得られた正しく動作する特性の異なる 3 種類のコマンドセットについて述べる．

4. 圧力コマンドセットの実現のために必要な機構

圧力コマンドセットを実現するために PC 上でシミュレータを実装し試行錯誤を重ねた．その結果，図 4 に示す機構を用意することで圧力コマンドセットの実装が可能であることがわかった．図は，圧力コマンドを処理するセル（図 4 右）が二次元メッシュ状に敷き詰められた構造（図 4 左）となっており PCA を模した構造となっている．このセルはステートマシン（State machine）で構成され，東西南北と自分自身への計 5 つのポートを持ち，スイッチ（Switch）によってコマンドの受け付ける方向を限定できるようになっている．以下にその詳細を説明する．

4.1 ステートマシン

オブジェクトの動作状態と圧力コマンドに対処するための状態（空き状態，回路として動作している状態，圧力を受けている状態，移動を決心した状態など）を識別するために，ステートマシンをおく．圧力コマンドセットを実装するためにはコマンド群とその状態遷移を考えなければならない．

4.2 圧力コマンドの通信路

オブジェクト間の通信において，デッドロックは重要な問題である．従来の PCA のルーティング方式は Wormhole 方式で実現されているため，圧力コマンドが大量に発行された場合高い確率でデッドロックが起

きる．デッドロックが起こらない通信の方法として，構造化バッファ，e-cube ルーティングなどが挙げられるが，これらの方法は複雑であり実装面積が大きくなることが予想される．そこで，隣接間のみで通信を行う場合のデッドロックは送信側の要求に対して受信側の応答を待つことが原因であることに着目し，応答を待たない通信路とした．

すなわち受信側に未処理のコマンドがあっても送信側はコマンドを出力できる．従ってデッドロックは起こらないが，受信側のコマンドはいつ上書きされてしまうかわからないこととなる．

4.3 コマンド受付方向の設定機能

通信路で直接の応答をしないため，つじつまの合う通信を行うためには別の工夫を行う必要が発生した．例えば圧力を受けてオブジェクトの移動，複製を行う場合，この情報を確実に要求元に伝えなければ矛盾が発生することになる．そこで確実に通信をすべき相手以外からのコマンドは処理しないようにし，確実に通信すべき相手との通信は応答をコマンドで行うこととした．これを実現するのがコマンドの受付方向の限定機能である．

4.4 フィードバックポート

複製を作りたいオブジェクトは圧力コマンドを一回出したからといって必ず周りが空くわけではないため，周りが空くまで何度も圧力コマンドを出し続けなければならなかった．ステートマシンを連続して動作させるためには連続してコマンドが入力されなければならない．そこで，入出力ポートを一組追加し自分自身へフィードバックさせることで連続動作を可能とした．

5. コマンドセット

本章では多くの試行錯誤の結果得られた 3 種類のコマンドセットについて紹介する．これらのコマンドセットは，オブジェクトを 1 個から 1024 個 (1,2,4,...,512,1024) まで正確に増やせることが確認されている．正しく増えたかどうかは，個々のオブジェクトに ID をつけ照合することによって確かめた．

5.1 Command set 1

5.1.1 概　　要

このコマンドセットは圧力コマンドを十字方向に出力し，十字方向にあるオブジェクトを外に押し出すことで空き領域を作り複製を作る．従って十字方向がすべてオブジェクトで埋まっていると，たとえ斜め方向が空いていてもそれ以上増えることができない．

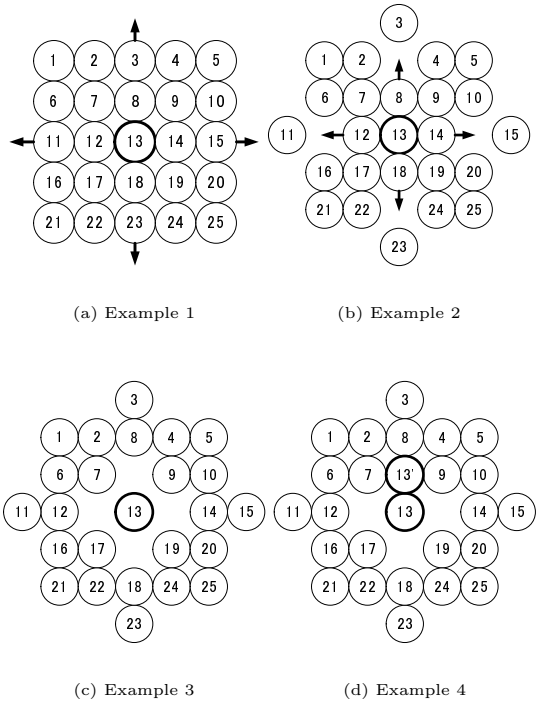


図 5　Command set 1 の動作例
Fig. 5　An example of operation for command set 1

5.1.2 動　作　例

図 5(a) において，13 番のオブジェクトが自分の複製を作りたい場合，十字方向に圧力コマンドを送る．すると，十字方向で一番外側の 3，15，23，11 番のオブジェクトが外に押し出される (図 5(b))．13 番の隣はまだ空いていないので続けて圧力コマンドを送り，8，14，18，11 番のオブジェクトが外に押し出される (図 5(c))．これで 13 番の隣が空き複製を作る．(図 5(d))

5.2 Command Set 2

5.2.1 概　　要

このコマンドセットは 4 方向に加えて斜め方向にあるオブジェクトにも圧力コマンドを伝えることができる．図 6 は，圧力コマンドが伝わる様子を表しており，圧力コマンドは，十字方向に進むメイン圧力コマンド（Main pressuer command）と，その進行方向に対して直角に進むサブ圧力コマンド（Sub pressuer command）からなる．そのため斜め方向にも圧力コマンドを送ることができ空き領域を作りやすい．

5.2.2 動　作　例

図 7(a) において，13 番のオブジェクトが自分の複

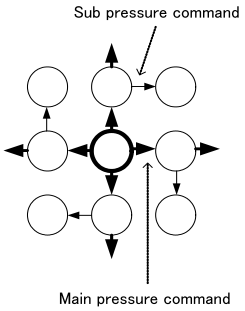
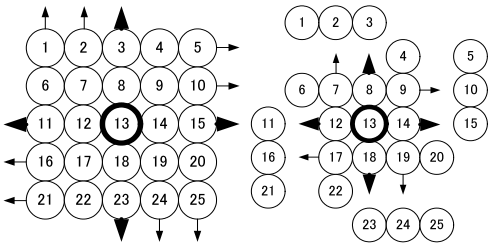
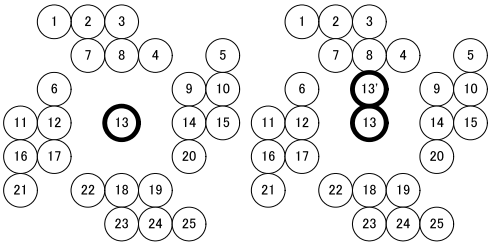


図 6　圧力コマンドが転送される様子
Fig. 6　Signs that a pressure command is transmitted



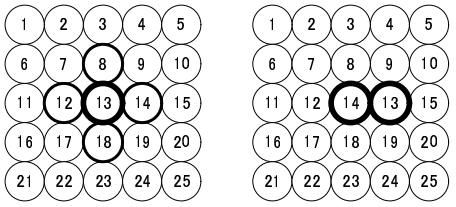
(a) Example 1　(b) Example 2



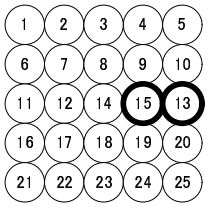
(c) Example 3　(d) Example 4

図 7　Command set 2 の動作例
Fig. 7　An example of operation for command set 2

製を作りたい場合，4 方向にメイン圧力コマンドを送り，十字方向にある 3,15,23,11 番のオブジェクトを外に押し出すとともに，サブ圧力コマンドによって 1,2,5,10,25,24,21,16 番も外に押し出す（図 7(b)）．13 番の隣は空いていないので続けてメイン圧力コマンドを 4 方向に送る．すると，8,14,18,12 番が外に押し出され，サブ圧力コマンドによって 7,9,19,17 番も外に押し出される（図 7(c)）．これで隣が空き状態となったので複製（13’ 番）を作る（図 7(d)）．



(a) Example 1　(b) Example 2



(c) Example 3　(d) Example 4

図 8　Command set 3 の動作例
Fig. 8　An example of operation for command set 3

5.3　Command Set 3

5.3.1　概　　要

このコマンドセットは泡が水圧を受け水面に向かって上昇するように，自らが外へ移動し空いている場所で自分の複製を作る．複製を作りたいオブジェクトは初め隣接する 4 方向を調べ，空き状態ならば複製を作り，4 方向ともオブジェクトならば入れ替えを繰り返し空いている場所まで移動しそこで複製を作る．

5.3.2　動　作　例

図 8(a) において，13 番のオブジェクトが複製を作りたい場合，8, 14, 18, 12 番のオブジェクトを調べる．1 番応答の早いオブジェクト（図 8(a) の場合 14 番のオブジェクト）と交換を行ない (図 8(b))，さらにその方向の 15 番と交換を行なう (図 8(c))．端まで移動した 13 番はそこで複製（13’ 番）を作る．(図 8(d))．

6.　評　　価

Bit Serial PCA への実装を意識し以下の観点から評価を行った．

- 圧力コマンドセットの実装面積
- 複製を行った後のオブジェクト間の距離
- 無駄なく PCA セルを使用できるかどうか

以下にその詳細を述べる．

表 1　コマンド数と状態数
Table 1　Number of commands and number of states

Command set	Number of commands	Number of states
Command set 1	6	5
Command set 2	8	5
Command set 3	11	4

6.1 実装面積

各コマンドセットの実装面積を比較するために，コマンド数と状態数について比較した．表 1 にその結果を示す．

コマンド数と状態数の値が少ないほど少ない面積で実装できることを表す．コマンド数の一番少ないコマンドセットは command set 1 で 6 個，状態数が一番少ないコマンドセットは command set 3 で 4 個であった．ステートマシンの規模はどのコマンドセットも 6 入力 6 出力のステートマシンで実装でき，ステートマシンの構造は command set 1 と command set 2 はともに入出力が 3bit で状態変数フィードバックが 3bit, command set 3 は入出力が 4bit で状態変数フィードバックが 2bit となる．従って command set 3 はデータバスのための面積がほかのコマンドセットに比べ不利である．command set 1 と command set 2 を比べると，状態数は 5 個で等しいが，コマンド数は 2 個 command set 1 が少ない．従って，ステートマシン内の組み合わせ回路は command set 1 が少なくなることが予想できる．これらの結果より，command set 1 が最も少ない面積で実装できると考えられる．

6.2 オブジェクト間の距離

オブジェクトを 1 個から 100 個になるまでまで 1 個ずつ増やした際のオブジェクト間の距離について調べた．測定方法はオブジェクトを 1 個用意し，複製の契機を示すコマンド (以降，複製コマンドと呼ぶ) を与え 2 個になるまで待ち，一番新しくできたオブジェクト（今回複製されたオブジェクト）に複製コマンドを与え，オブジェクト数が 100 個になるまでこの操作を繰り返した．そして，i 番目 ($1 \leqq i \leqq 99$) にできたオブジェクト（以降，親と呼ぶ）と $i+1$ 番目にできたオブジェクト（以降，子と呼ぶ）間の平均と最大のマンハッタン距離を測定した．図 9 にその結果を示す．

縦軸は距離を示し，横軸は平均か最大かあるいはどのコマンドセットかを示す．平均距離が一番小さいのは command set 2 の 2.04 で，一番大きいのは command set 1 の 3.13 であった．

平均距離がどのコマンドセットも 1 前後でない理由

図 9　オブジェクト間の距離
Fig. 9　Distance among the objects

は以下のことがあげられる．command set 1 と command set 2 は，圧力コマンドによって親のオブジェクトを移動させてしまうため親子間の距離が離れていくためであると考えられる．command set 3 に関しては，オブジェクトが 1 列に並んでいる場合，複製を作りたいオブジェクトは端から端まで移動してしまうので親との距離が離れてしまうためであると考えられる．

最大距離が一番小さいのは command set 1 の 8 で，一番大きいのは command set 2 の 11 であった．command set 2 は圧力コマンドを全方向へ出力するので周りのオブジェクトを移動させやすいため最大距離がほかのコマンドセットに比べ大きくなったと考えることができる．しかし，移動しやすいので適当な隙間が空き，複製を作りたいオブジェクトはすぐ隣に複製を作れる場合が多いため平均距離は少ない結果となっている．

6.3 有効利用率

有限範囲内でオブジェクトが複製を作ることができなくなるまで増殖させたときの利用率について測定した．実験方法は，1 個のオブジェクトを作り，2 種類の方法で複製コマンドを与えた続けた．そして，複製することができなくなったときのオブジェクト数から利用率を割り出した．測定は 12×12 個の PCA セルで行った．これは，PCA-2 のセル数と同じである [3]．この値は 100% のときすべての PCA セル ($12 \times 12 = 144$ 個) を使うことができることを意味する．

複製コマンドを与える方法は以下の 2 種類である．

- 新しくできたオブジェクト（子）だけに与える方法（6.2 節の与え方と同じ方法）
- ランダムに与える方法

はじめの方法は，特定の機能を持つ（メモリの機能な

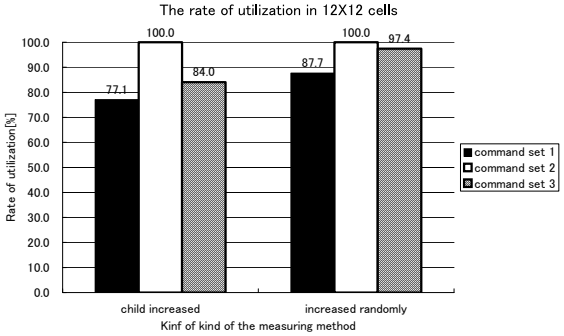


図 10　利用率 12×12 個
Fig. 10　Rate of utilization in 12×12 cells

ど）オブジェクトが複製を繰り返す場合を想定しており，あとの方法は複製するモジュールが複数あり不規則に複製していく場合を想定している．これは 10 回の平均の値を測定した．図 10 にその結果を示す．

横軸は複製コマンドを与える方法別になっており，縦軸は利用率を表す．左のグラフは子だけに複製コマンドを与えた場合の利用率で，右のグラフはランダムに与えた場合の利用率である．

子だけに複製コマンドを与えた場合の利用率は command set 2 が 100% で，最も利用率が低いのは command set 1 の 77.1% であった．ランダムに複製コマンドを与えた場合の利用率は command set 2 が 100% で，最も利用率が低いのは command set 1 の 87.7% であった．

これらの結果からもわかるように command set 2 はどのような場合でも無駄なく PCA セルを使うことができる．これはメイン圧力コマンドとサブ圧力コマンドによってすべての方向へコマンドを伝えることができることが原因である．

command set 2 以外のコマンドセットの利用率が低い原因として次のことが言える．command set 1 は十字方向すべてにオブジェクトがあると斜め方向に空き領域があってもオブジェクトが移動できないため複製を作れないことが原因である．command set 3 は外周のすべてがオブジェクトで埋め尽くされると複製を作りたいオブジェクトは外周を回り続けるため中央に空き領域があってもそれ以上複製を作ることができなくなることが原因である．

6.4 結　　果

前節の評価によってそれぞれのコマンドセットの特質がわかった．その結果，command set 2 が Bit Serial PCA に適していると考えられる．その理由は３つのコマンドセットの中で最も PCA セルを有効に利用でき，またオブジェクト間の平均距離が最も短く，実装面積も 2 番目に少ないからである．しかし，オブジェクト間の最大距離は 3 つのコマンドセット中で最も大きいため，処理のクリティカルパスが大きくなる危険があり，最大距離を短くする工夫が今後必要となると考えられる．

7. Command set 2 の詳細

7.1 移動，複製の手順

図 7(a) の 3 番のオブジェクトが同図 (b) のように移動する過程ついて詳細に説明する．同図 (a) において，3 番の北方向に隣接している空き状態のセルが 13 番から伝わってきたメイン圧力コマンドを受け付けると，“予約済み状態” へ遷移し，コマンドを南方向しか受け付けないよう限定（以降，ロックオンと呼ぶ）し，“予約済みコマンド” を 3 番へ戻す．3 番は予約済みコマンドを受け取ると，北方向にロックオンし，“作成コマンド” を北方向へ発行する．これでお互いがロックオンした状態となり，本来はここで回路データの移動が行われる．予約状態のセルは作成コマンドによってオブジェクトの状態になり，ロックオンを解除するとともに，“完了コマンド” を戻す．13 番はそのコマンドによって，ロックオンを解除し，自分自身を削除して移動を終了する（図 7(b)）．複製の過程（図 7(c) の 13 番が同図 (d) になる過程）は移動とほぼ同じで，完了コマンドで自分自身を削除しないだけの違いである．

7.2 コマンドの例外処理

複製を作りたいオブジェクトの周りがすべて空いていた場合，圧力コマンドによって 4 方向すべてのセルが予約済みの状態となり，予約済みコマンドが 4 方向から戻ってくるが，4 方向のうち 1 方向しか複製は行わない．従って残り 3 方向の予約された状態にあるセルに対して，もとの空きの状態へ戻すためのコマンドを発行しなければならない．このような例外的なコマンドを発行した相手や自分自身に対して，複製のやり直し，移動の強制終了を意味するコマンドを発行するなどの工夫をしている．

8. む　す　び

本論文では，圧力の概念を基本とした新しい領域管理方を提案した．そして，それを実現するための 3 種類のコマンドセットを簡単に紹介するとともにその評

価を行い，コマンドセット 2 が優れていることを述べた．コマンドセット 2 については完成できるまでに発生した問題の関係などを含め詳細な説明を行った．いずれにしても管理者のない全く新しい領域管理法のための基本的な動作を明らかとすることができた．

文　献

[1] K. Oguri, N. Imlig, H. Ito, K.Nagami, R. Konishi and T.Shiozawa, “General purpose computer architecture based on fully programmable logic,” Proc. International Conf.Evolvable System (ICES’98), pp.323-334, 1998.

[2] K.Nagami, K. OGURI, T. Shiozawa, H. Ito, and R.Konishi, “Plastic cell architecture: A scalable device architecture for general-purpose reconfigurable computing,” IEICE Trans.Electron.,vol.E81-C, no.9, pp1431-1437,Sept.1998.

[3] H. Ito, R. Konishi, H. Nakada, H. Tsuboi, Y. Okuyama and A. Nagoya, “Dynamically Reconfigurable Logic LSI: PCA-2,” IEICE Transactions on Information and Systems, Vol. E87-D, No. 8, pp. 2011-2020, Aug. 2004.

[4] H. Ito, R. Konishi, H. Nakada, H. Tsuboi and A. Nagoya, “Dynamically Reconfigurable Logic LSI designed as Fully Asynchronous System - PCA-2” , Proc. of COOL Chips VI, pp. 84, Apr. 2003

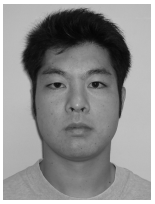
[5] Kiyoshi OGURI, Yuichiro SHIBATA, Nobuyoshi MURAKAMI, Akira KAWANO and Akira NAGOYA, “Asynchronous Bit-Serial Datapath for Object-oriented Rconfigurable Architecture PCA”, IEICE TRANS.INF.&SYST. ,VOL.E82,No.1,JANUARY 2005

[6] 神山 知己　倉田 圭吾　池畑 陽介　北道 淳司　黒田 研一, “動的再構成デバイス PCA 上での自己複製型アプリケーション設計容易化手法の提案と実装”, 情報処理学会研究報告 IPSJ SIG Technical Reports, vol.2005 No.8, pp.141-146, 2005

[7] 小佐々 武志，柴田 裕一郎，小栗 清，“PCA アーキテクチャの OS 機能に関する研究” 電子情報通信学会九州支部学生会講演論文集， Vol. 10， pp. 97，2002 年 9 月

[8] 小栗 清 , 佐藤 晶浩, 小佐々 武志, 永本 太一, 柴田 裕一郎: ” プラスティック・セル・アーキテクチャと非同期ビットシリアルデータパスパルテノン研究会資料集,vol. 21,pp. 70–78(2002).

（平成 xx 年 xx 月 xx 日受付）



永本　太一

平 14 長崎大・工・情報システム卒．平 16 同大学大学院・生産科学研究科博士課程前期了．同年同大学大学院生産科学研究科システム科学専攻博士課程後期入学，現在に至る．プラスティックセルアーキテクチャの研究に従事．

矢野　智史

2004 長崎大・工・情報システム卒．同年同大大学院生産科学研究科電気情報工学専攻修士課程に入学，現在に至る．計算機アーキテクチャ，ビットシリアル処理回路設計環境の研究に従事．

内田　　満

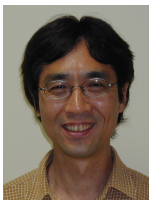
2004 長崎大・工・情報システム卒．同年同大大学院生産科学研究科電気情報工学専攻修士課程に入学，現在に至る．計算機アーキテクチャ，ビットシリアル処理回路設計環境の研究に従事．

高野　智明

2004 長崎大・工・情報システム卒．同年同大大学院生産科学研究科電気情報工学専攻修士課程に入学，現在に至る．計算機アーキテクチャ，ビットシリアル処理回路設計環境の研究に従事．

柴田裕一郎　（正員）

平 8 慶大・理工・電気卒．平 13 同大大学院理工学研究科計算機科学専攻後期博士課程了．博士（工学）．同年長崎大学工学部情報システム工学科助手．現在，同講師．平 10〜平 13 日本学術振興会特別研究員．リコンフィギャラブルアーキテクチャの研究に従事．平 15 年度本会論文賞．



小栗　　清　（正員）

昭和 49 年九州大学理学部物理学科卒業．昭和 51 年九州大学大学院理学研究科修士課程修了．昭和 51 年日本電信電話公社（現 NTT) 入社．平成 12 年同社退職．平成 12 年長崎大学工学部教授，現在に至る．計算機アーキテクチャ，設計自動化，ハードウェア記述言語，論理合成システム，再構成可能ハードウェア，非同期回路，ビットシリアル処理の研究に従事．工学博士．電子情報通信学会，情報処理学会各会員．昭和 62 年元岡賞受賞．平成 2 年情報処理学会論文賞受賞．平成 4 年大河内記念技術賞受賞．平成 12 年電子情報通信学会業績賞受賞．

Abstract In this paper, we propose a novel area management method on the basis of a concept of "pressure". PCA (Plastic Cell Architecture) is a dynamically reconfigurable device which was proposed paying attention to the essential of processing and the flexibility of Von Neumann architecture. The mechanism in which domains are managed requires the function like "malloc" of the C language. However, for the dynamically reconfigurable device as PCA, uniform management and parallelism are incompatible. Therefore, it is necessary for any circuit of PCA to find out their vacant area without a administrator. And so, we introduced a new domain management method which obtains the vacant areas by moving the surrounding modules by "pressure". In order to realize it, a new command set is needed. We describe the problems and solutions which we have considered in development of three command sets. Finally, we evaluate these command sets about the execution area, the processing efficiency, the distance among modules and the rate of effective use. In these results, the command set 2 is the best command set.

Key words PCA, Dynamic reconfigurable device, area management, pressure, command set.