システムＬＳＩ設計技術　93－14
（1999. 11. 27）

非同期式動的再構成論理LSIの設計

伊藤 秀之　小栗 清　小西 隆介　中田 広

NTT 未来ねっと研究所

〒 239-0847 神奈川県横須賀市光の丘 1-1

hi@exa.onlab.ntt.co.jp

あらまし　当研究グループで提案する動的再構成可能計算機アーキテクチャである，Plastic Cell Architecture (PCA) の LSI (PCA chip) について述べる．PCA は，可変論理部分と，その構成機能を持つ固定論理部分の二重構造を持つ可変論理素子上に，オブジェクトという形で布線論理を構成し，それらの間のメッセージ通信で機能を実現する計算機アーキテクチャである．本稿では，PCA chip による動的再構成機能の実現方法と，その非同期設計について説明する．

キーワード　動的再構成，自律再構成，非同期システム，FPGA，VLSI，論理回路，並列処理

PCA chip : Asynchronous Design of Dynamically Reconfigurable Logic LSI

Hideyuki ITO　Kiyoshi OGURI　Ryusuke KONISHI　Hiroshi NAKADA

NTT Network Innovation Laboratories

1-1 Hikari-no-oka, Yokosuka-shi, Kanagawa 239-0847 Japan

hi@exa.onlab.ntt.co.jp

Abstract　In this paper, we report design of PCA chip, which is a LSI chip to realize Plastic Cell Architecture (PCA). PCA is a dynamically reconfigurable computer architecture based on the logic device that has both reconfigurable part and built-in function of its configuration. On PCA device, function and its reconfiguration is realized by object and message. We describe reconfiguration mechanism and its asynchronous design methodorogy of PCA chip.

key words　dynamic reconfiguration, autonomous configuration, asynchronous system, FPGA, VLSI, wired-logic, parallel compuing

1 はじめに

当研究グループでは，Plastic Cell Architecture (PCA) と呼ぶ動的再構成可能計算機アーキテクチャを提案し，現在この実現に向けて，HDL，CAD/DA システム，アプリケーション，再構成可能論理 LSI (PCA chip) の開発を行なっている．

PCA は，布線論理の構成 (configuration, programing) により汎用の情報処理を実現することを目指した計算機アーキテクチャである [1][2].

布線論理で汎用の情報機能を実現するためには，布線論理が布線論理を新たに構成する機能，すなわち布線論理の自律的構成機能が不可欠である．この自律的構成機能を実現するには，布線論理を構成するための可変部分と，可変部分の構成を行なうための組み込み機能，そして組み込み機能と可変部分に構成される布線論理の間のインターフェースの確立が必要である．このような可変部分と組み込み機能の二重構造が，汎用情報処理機構を実現する上で不可欠と考えられる．

また，布線論理によるプログラマビリティを高めるには，布線論理の動的な再構成が必要である．動的再構成機能の実現には，構成する布線論理のモジュール性を高め，その物理的な境界を動作時の再構成に適した形にすることが必要である．

以上の概念を実現するため，現在 PCA の VLSI (PCA chip) の設計を行なっている [3]. 本稿では，この PCA chip の設計の詳細について述べる．

2 Plastic Cell Architecture

構造の基本となるのは，可変部分と組み込み機能の二重構造を持つセル (PCA セル) である．PCA セル内部で，可変部分は “可変部”(plastic part, PP) と呼ぶ可変論理素子部分であり，組み込み機能は “組み込み部”(built-in part, BP) と呼ぶ部分が実現している．PCA のハードウェアは，この PCA セルがメッシュ結合されたセルアレイ型のアーキテクチャである．セルアレイ上では，絶対的なアドレス付はせず，*N, E, W, S* の方向を定め，方向とセル間の距離を用いた相対的な位置指定を行なう (図 1).

セル間の PP と BP は，個別の信号線でメッシュ結合されており，それぞれ PP 面，BP 面を構成している．PP 面は一つの可変論理 LSI として機能する．BP 面は，可変論理の構成機能を実現する．また，BP 面は PCA セルアレイ上でメッシュ型ネットワークを構成する．また，PCA セル内部で，PP と BP の間は，双方向の通信用と PP の構成用の 2 系統の信号線により接続されている．

PCA の機能実現の基本となるのは，オブジェクトとメッセージである．

オブジェクトは，メッセージ通信のためのインターフェースを持った布線論理のモジュール (機能オブジェクト) である．また，処理に必要なデータの記憶領域もオブジェクト (メモリオブジェクト) として扱う．これらのオブジェクトは PP 面で作る可変論理素子上に実現する．

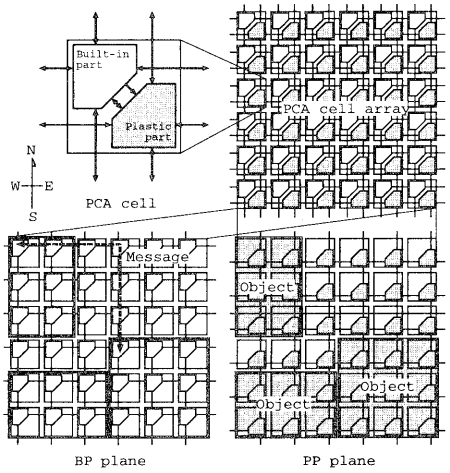

図 1: PCA の基本構造

メッセージは，オブジェクト間通信を行なうデータと命令の列である．メッセージの転送は，BP 面で作るメッシュ型通信ネットワークで行なう．また，オブジェクトの生成・消滅機能も，BP によるメッセージの処理で実現する．したがって，オブジェクトがオブジェクトの生成・消滅用のメッセージを作り出すことで，布線論理の自律的構成が可能である．

オブジェクトは，メッセージ通信のインターフェース以外には布線論理としての接続を持たず，PP 上の閉じた回路となっている．したがって，オブジェクトを再構成の単位にすることで，他の回路に影響を与えずに，動的に布線論理を再構成することができる．また，オブジェクト間のデータ転送をメッセージ通信で行なうため，オブジェクト間には静的なデータパスを付設する必要がなく，オブジェクトの配置によらず，自由度の高いデータ転送が可能である．

3 PCA chip

PCA chip は，同一の PCA セルを並べたセルアレイとして設計している．以下本節では，PCA chip の特徴と，PCA セルの構成と機能について説明する．

3.1 PCA chip の特徴

3.1.1 均質性

PCA chip は PCA セルの繰返しで設計できるため，PCA セルを高集積化することで，システムの高集積化が可能である [2].

そこで，PCA セルの高集積化を容易にするため，PP を

LUTの敷き詰め構造としている．つまり，他のFPGAが，その基本セルをLUT，レジスタ，スイッチングマトリクスの混在型としているのに対して，PCAでは，これら全てをLUTで実現することで，PPを均質な構造にしている．この構造にすることで，PP全体がSRAMとして設計でき，SRAMと同等の高集積化が可能となる．

また，論理ゲート，配線，ラッチのすべてをLUTの組合せで実現するため，デバイス上のリソースの制約を受けずに，PP上に自由度の高い配置配線を行なうことができる[4]．

3.1.2 非同期システム

PCAの構造はPCAセルの繰返しであり，chipの接続で，並列に動作するオブジェクトを構成するためのセル空間を拡大することが可能である．このように規模の拡張を前提としたハードウェアの物理的な実現には，同期設計よりも，非同期設計の方が適している[5]．

第一に，大規模回路の同期システム化にはクロックスキューなどの点で限界がある．PCAの処理は，オブジェクト内の局所的な布線論理の動作と，離れたオブジェクト間のメッセージ通信が基本である．Chip接続による規模の拡張を考えると，chip間の接続を含んで離れた場所に配置されるオブジェクト間の通信が必要である．これを単一の同期システムとして設計することは不可能で，要素間での非同期的な通信が必要である．しかし，同期要素間で非同期通信を行なう場合には，非同期信号を取り込むために設ける双安定素子のメタスタビリティによる誤動作の危険性を回避することは原理的に不可能である[6]．

また，処理のほとんどが局所的な回路内の動作であることから，最大遅延が性能を支配する同期設計システムよりも，平均遅延を小さくすることで性能の向上が期待できる非同期システムとする方が有利である．

そこで，PCA chipはクロック信号を排除した，非同期システムとして設計している．BPは，ハンドシェイクを用いたメッセージ通信の実現に適した非同期回路として設計している．また，オブジェクトの布線論理も非同期回路とすることで，PPの可変論理素子上からもクロック信号を排除している．これにより，可変論理素子上のクロック信号の配線，クロック信号に依存するリソース配置などの制限もなくし，PPをより均質で高集積化に適した構造にしている．

3.2 PCA セルの構成

3.2.1 Plastic part

PPは，以下に述べる *Sea of LUT* 構造で，4入力1出力のLUTの結合網になっている．このLUT 4個が基本セル (basic cell, BC) を構成する．BCには，*N,E,W,S* の方向に隣り合う4BCから1ビットずつ，4ビットが入力する．BC内の各LUTは，この4ビットを共通に入力し，1ビットを出力する．4つのLUTの出力が隣接する4つのBCへ出力される (図2)．

PPは，このBCを 8×8 個アレイ上に並べたものである．可変論理素子としては，PPは $256 = 8 \times 8 \times 4$ 個のLUTで構成されている．これがPCA chipの可変論理の書き換えの単位となる．布線論理の構成は，PPにLUTの構成データを書き込むことで行なう．そのため，PPはアドレス信号，データ入出力信号，書き込み信号，読み出し信号のSRAMのインターフェースを持ち，BPからはメモリとして扱うことができる．メモリとしては，PPは容量が4096ビット，アドレス10ビットのSRAM構造となっている．このSRAM機能により，PPは汎用メモリとしても利用できる．

PPは *N,E,W,S* 方向のPPと隣接しており，それぞれの方向のPPと，8ビットずつの入出力信号で結合されている．そして，PCA chip全体では，6×6 個のPPが一つの可変論理素子を構成している．

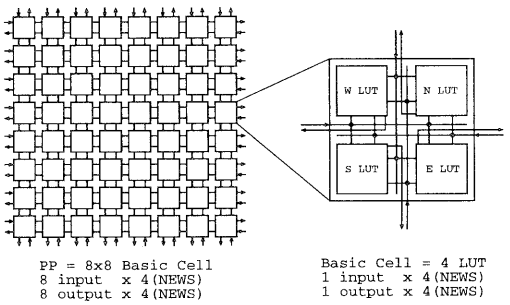

図2: Plastic partと basic cell

3.2.2 Built-in part

BPは *N, E, W, S* の方向の4つのBPと隣接し，それらのBPとの間で双方向のメッセージ通信を行なう．また，PCAセル内部のPPの可変論理部を構成する機能を持ち，更にPP上に構成するオブジェクトとの間で，BP間と同等の双方向のメッセージ通信を行なうことができる．

これらの機能の実現のため，BPは，*N, E, W, S, PP* の5方向に入力ポートと出力ポートを備えている．この他，PPの構成機能のためのLUT制御部を持ち，各出力ポートとLUT制御部に対応した6個のアービタモジュールを備える (図3)．

入力ポートは，隣接BPとオブジェクトからメッセージを入力する．メッセージは，5ビットのワードからなる可変長のストリームであり，メッセージ通信には，5ビット幅のデータ信号線，*Request*，*Acknoledge* の2本のハンドシェイク信号線を用いる．入力ポートは，メッセージを受けると，メッセージに含まれる命令に従って動作する．この部分はステートマシンとして実現している．

命令は5ビット幅で，全部で12個である (表1)．入力

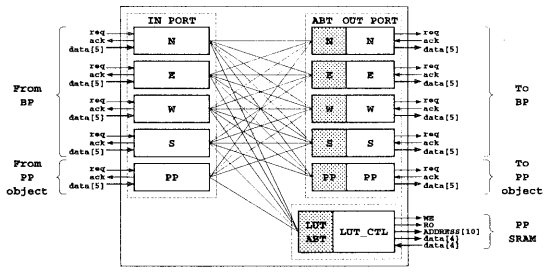

図 3: 図:Built-in part のモジュールと入出力

ポートの動作は，メッセージの経路設定と転送，PP のメモリ操作，BP-PP 間の通信インターフェースの制御である．

出力ポートは，メッセージの出口で，隣接する BP の入力ポートまたはセル内の PP と接続されている．メッセージの経路設定は，入力ポートが，出力ポートとの間に，ハンドシェイク回路を設定して，メッセージの転送路を確保することで実現する．その後，入力ポートが，設定されたハンドシェイク回路を通して，隣接 BP あるいはオブジェクトとの間でハンドシェイクを行なうことで，メッセージを転送する．

LUT 制御部は，入力ポートの制御を受けて，PP のメモリへの Read/Write 等の動作を行なう．LUT 制御部は，10 ビットのアドレスカウンタを持ち，入力ポートから制御されるステートマシンとして実現している．

BP 内の 5 つの入力ポートは，独立にメッセージを受けて動作する．BP の動作は，入力ポートが，出力ポートや LUT 制御部と連係することで実現している．これらのモジュールは BP 内の共有資源となっている．従って，入力ポート間での共有資源の排他制御が必要である．

アービタモジュールは，入力ポートからの共有資源の利用要求の排他制御を行なう．入力ポートは，利用する共有資源に対応したアービタモジュールに *Request* を送り，*Acknoledge* を受けてから動作を開始する．

表 1: 命令セット

命令	意味
CLEAR	メッセージの終端
PP_OPEN	PP-BP 間の通信口を開く
PP_CLOSE	PP-BP 間の通信口を閉じる
CO	データ読み出し
COCI	オブジェクトのコピー
CIM	データ書き込み
CIM	構成データ書き込み
NORTH	N 出力ポートへ経路設定
EAST	E 出力ポートへ経路設定
WEST	W 出力ポートへ経路設定
SOUTH	S 出力ポートへ経路設定
PP_OUT	PP 出力ポートへ経路設定

3.2.3 PP-BP 間の通信インターフェース

BP-PP 間には，先に述べた PP の SRAM インターフェースの他に，メッセージ通信のインターフェースを設けている．BP の PP 入出力ポートと PP 上のオブジェクトは，BP 間と全く同じハンドシェイクによる通信を行なうよう設計している．

その他に，PP 上に構成するオブジェクトのリセット動作も BP から行なうため，リセット用のインターフェースも設けている．

従って，PP-BP 間には，7 本のメッセージ通信用の信号線 (`data[5]`, `req`, `ack`) を 2 組と，オブジェクトのリセット用の信号線を 2 本 (`reset_req`, `reset_ack`)，合計 16 本 (入力 8, 出力 8) の信号線を設ける．

PP は，これらの信号線を W 辺の入力信号，N 辺の出力信号に接続している．入力信号は，隣接する PP からの入力信号と共有できるように，セレクタを設け，これを BP が制御している．

また，PP 側から BP 側へのハンドシェイクの入力信号となる信号線については，論理ゲートを設けて開閉を BP が制御している．これは，構成中，あるいはメモリとして使用中の PP から，BP のハンドシェイク回路に不要な信号を入力させないためである．

4 PCA chip の非同期設計

4.1 設計方式

4.1.1 束データ方式

非同期回路の設計には，採用する遅延モデルによって多くの異なる方式が存在している [5]．PCA chip の場合，クロック信号に依存しないシステム構成とすること，この一点が非同期設計を採用する理由となっている．そのため，遅延モデルには同期設計と同じもの，すなわち，ゲート遅延，配線遅延は共に見積り可能というモデルを採用している．この遅延モデルに基づき，組合せ回路の出力が安定するまでの遅延時間は設計時に見積り可能，という前提で設計を行なう．

回路構成は束データ方式に基づいている．束データ方式の回路構成は，レジスタ間のデータ転送において，データ線をまとめて束データとして扱い，束データの安定後に，受信側レジスタに *Ready* 信号を送ることで，データの取り込みを行なう方式である．

PCA chip の設計では，束データ方式の *Readey* 信号を生成する回路として，Muller-C 素子 (C ゲート) と，*Request*, *Acknoledge* の信号，束データの遅延時間を模擬する遅延素子で構成するハンドシェイク回路を用いている．

4.1.2 PP と BP の設計方式の違い

PCA chip では，BP の回路，PP 上のオブジェクトの回路とも，同じ遅延モデルに基づき，束データ方式による非同期回路設計を行なっている．しかし，BP とオブジェクトの回路では設計の条件が異なるため，幾つか異なる点がある．

BP の動作は，メッセージを受け，命令を解釈し，命令に応じてメッセージを転送するというストリーム型の処理が基本である．複数のメッセージが独立に転送され，合流点の BP では，メッセージは到着順に処理される．従って，BP の回路の動作は，独立したタイミングで動作する入力ポートのハンドシェイク信号の排他制御と，切り替え制御が中心である．

そのため，BP の回路設計は，アービタによる非同期信号の排他制御を導入している．この方式では，メッセージを処理する BP のみを動作させ，他を停止させることが可能である．

一方，オブジェクトの回路は，PCA chip のユーザが，CAD/DA システムを用いて設計するものである．そのため，ハザードなどの非同期回路固有の問題がなく，自動設計技術が確立されている同期設計との親和性の高い回路構成とするのが望ましい．

そのため，オブジェクトの回路では，ハンドシェイクの合流点で，全ての信号を待ち合わせる方式としている．つまり，オブジェクト内のデータパスにあわせて，タイミング制御回路のハンドシェイクも合流・分岐させる．合流点では，非同期信号間のアービトレーションは行なわず，複数のデータパスからのハンドシェイク信号の待ち合わせを行なう．

この方式を用いることで，オブジェクトの回路は，同期回路と同様の方法で構成することが可能である．また，そのような回路構成に必要な C ゲート，D ラッチは LUT の組み合わせで構成できることもわかっている [7].

4.2 BP の制御方式

以下では，BP の動作に必要な制御の実現方法を説明する．ハンドシェイク信号は *High* の時に真で，以下の説明では，これを H と表記し，偽を L と表記する．また，信号の立上りを+で，立ち下がりを-で表す．

4.2.1 ハンドシェイクとデータ転送

PCA chip のハンドシェイク回路は，待ち合わせ要素に C ゲートを用い，その間を *Request* と *Acknoledge* の 2 本のハンドシェイク信号で結んだものである [8]. C ゲートは，送り側からのデータ更新の要求信号 (reqi)，受け側からのデータ更新の応答信号 ($\overline{\texttt{acki}}$) を待ち合わせて，出力を生成する．この信号を，自身のデータの更新に使用し，それを送り側へのデータ更新の応答信号 (acko) と受け側へのデータ更新の要求信号 (reqo) として送信する．

このハンドシェイク回路で，送り側 (*Sender, S*) と受け側 (*Receiver, R*) の間で，S から R への Req 信号と，R から S への Ack 信号を用いて，4 サイクルハンドシェイクに従ったデータ転送を行なう．

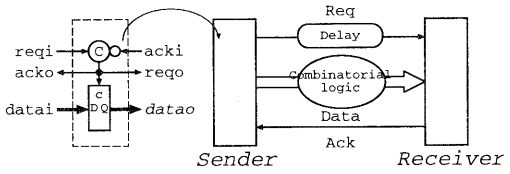

図 4: ハンドシェイク回路とデータ転送

4 サイクルハンドシェイクでは以下のような手順でハンドシェイクを行ないデータを転送する (図 4, 5)[6].

1. S でデータが確定.
2. S が req 信号を上げる (req+).
3. R にデータが届く.
4. R が req+ を受けてデータを取り込み，ack 信号を上げる (ack+).
5. S が ack+ を受けて req 信号を下げる (req−).
6. R が req− を受けて ack 信号を下げる (ack−).
7. S が ack− を受けてデータを更新し，1 に戻る

ただし，*S-R* 間の Req 信号と Data 信号の間に以下の遅延関係が成り立つように，req 信号線上に遅延素子を挿入する．

$$\texttt{Data}\text{の遅延} < \texttt{Req}\text{信号の遅延} \qquad (1)$$

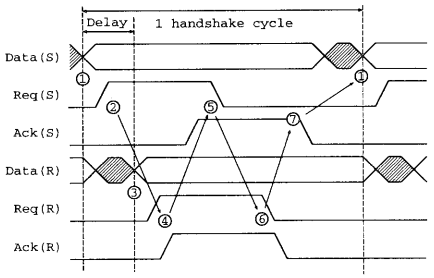

図 5: 4 サイクルハンドシェイク

4.2.2 ステートマシン制御

入力ポートは，メッセージを受けて，メッセージ中の命令に応じた動作を行なう．この機能は，ステートマシンとして実現している．また，LUT 制御部も，入力ポートからの制御されて動作するステートマシンとして実現している (図 6-b).

ステートマシンの制御は，基本的にはデータ転送の場合の制御と同じである．図に示すように，同一の C ゲートで制御する FF に格納された入力値，状態値から，組合せ

論理回路により次状態値と出力値を決定し，次状態値は状態値のFFに，出力値は別のCゲートで制御するFFに転送する順序回路を考える．

この順序回路の制御は，入力値のFFを制御するCゲートと出力値のFFを制御するCゲートの間の4サイクルハンドシェイクで行なう．ただし，出力値がFFに到着するまでの遅延と，次状態値がFFに到着するまでの遅延を求めて，大きい方の遅延よりも大きい遅延素子をCゲート間のreq信号線上に挿入する．

このような回路で，入力値と状態値(初期値)が確定した後，4サイクルハンドシェイクを開始し，組合せ論理の出力が確定し，出力値と次状態値がともにFFの入力で確定すると(4サイクルハンドシェイクの手順3)，出力値側のCゲートとのハンドシェイクが行なわれて，出力値がFFに取り込まれる(同じく手順4)．その後，4サイクルハンドシェイクの1周期分が終了して，入力値と状態値のFFにデータ更新のための取り込み信号が送られて，次の入力値とともに，組合せ回路の出力である次状態値がFFに取り込まれる(同じく手順7→1)．

4.2.3 ハンドシェイクの分岐点の制御

BPの動作は，入力ポートが他のモジュールを利用して連係動作することで実現している．これらの連係動作は，入力ポートとモジュールの間のハンドシェイク回路を切替えることで実現する(図6-c)．例えば，メッセージの転送経路の設定は，経路設定系の命令を受けて，入力ポートから出力ポートへのハンドシェイク回路を切替えることで行なっている．

非同期回路では，データパスを切替える場合，実際にデータを流す方向のハンドシェイク回路だけ動作させれば良い．つまり，送り側Sから受け側$R1$,$R2$への2つのデータパスがある場合，S側の制御で，データをどちらか一方(例えば$R1$)に送る場合，Sは$R1$とだけハンドシェイクすればよい．こうすることで，$R1$のデータパスにはデータを送り，$R2$側のデータパスは停止させることができる(図??)．

BP内部では，このデータパスの切替えをハンドシェイク回路上の論理ゲートでスイッチすることで行なっている．つまり，Sからのreq信号を，$R1$へのreq1信号と$R2$へのreq2信号との間で切替えて，一方のハンドシェイク回路だけを動作させることでデータパスを切替えている．

ただし，ハンドシェイク回路の*Request*線の切り替えは，ハザードが混入しないよう行なわなくてはならない．そのために，4サイクルハンドシェイクの手順2の前の段階，ハンドシェイク信号線が全てLの状態で，S側からの制御論理信号をスイッチとなる論理ゲートに送っておく必要がある．つまり，スイッチの*Request*信号と制御論理信号の間で，以下の順序関係が成り立つ必要がある．

$$\text{制御論理の確定} \rightarrow \textit{Request}\text{の到着} \qquad (2)$$

この関係を満たすように*Request*信号線上に遅延素子を挿入している．

4.2.4 ハンドシェイクの合流点の制御

BPの動作は，複数の入力ポートが，共有資源であるモジュールを排他的に利用することで実現している．この共有モジュールの排他利用は，アービタモジュールによるハンドシェイク信号の排他制御と，ハンドシェイク信号の合流点で制御で実現する(図6-d)．

例えば，複数の入力ポートが同じ方向にメッセージを転送する場合，これらの入力ポートは，同じ出力ポートとの間にデータパスを設定するため，この出力ポートを排他利用しなくてはならない．そのため，入力ポートは，まずアービタモジュールに資源利用のためのハンドシェイクを行ない(*Request*を送る)，先行した入力ポートがアービタとのハンドシェイクを完了し(*Acknoledge*を受ける)，出力ポートとハンドシェイクを開始する．

BP内部では，全入力ポートと出力ポートの間に，ハンドシェイク信号線の対を設けてある．そこで，複数の入力ポートと出力ポートの間で，選択的なハンドシェイクを行なう機構が必要である．

そのために，入力ポートと出力ポートの間のreq信号線，ack信号線，およびデータ線を，アービタからの制御論理で選択している．アービタからの制御論理信号は，出力ポートとのハンドシェイクに先だって行なうアービタとのハンドシェイク完了時に確定している．従って，ハザードを混入させることなく，ハンドシェイク回路を切り替えることができる．

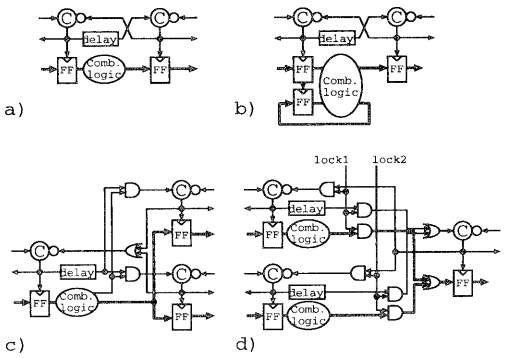

図6: ハンドシェイク回路の構成 a) データパス b) ステートマシン c) 分岐 d) 合流

4.3 アービタによる非同期信号の排他制御

BPの非同期設計では，複数のモジュールから独立したタイミングで入力する*Request*信号のうち，先に入力した信号に対して*Acknoledge*信号を返すアービタモジュール

を使用している．以下では，BP 内のアービタモジュールについて説明する．

4.3.1 2-1 アービタモジュール

アービタモジュールの中心部は，相互排除 (interlock) モジュールと呼ぶ回路である (図 7-a)[6]．相互排除モジュールは，req1，req2 の 2 入力と，ack1，ack2 の 2 出力を持ち，先に H に変化した入力に対応した出力を H にする．後から H となった入力に対応した出力は，先の入力が L に変化した後，H にする．相互排除モジュールの特徴は，2 つの入力が同時に入力した場合でも，内部の双安定素子のメタスタビリティ状態で生じる中間電位の信号を出力せず，どちらか一方に規定電位の信号を出力できる点である．

この要素に C ゲートを用いた付加モジュール (call モジュール) をつけて，2-1 アービタモジュールを構成する (図 7-b,c)．この付加モジュールは，2 つの入力側ハンドシェイク (reqi1-acko1，reqi2-acko2) と 1 つの出力側ハンドシェイク (reqo-acki) を持ち，入力側ハンドシェイクと出力側ハンドシェイクを結びつける働きをする．ただし，reqi1，reqi2 は，排他的に入力されることが前提である．

2-1 アービタモジュールは，2 つの入力側ハンドシェイク (reqi1-acko1，reqi2-acko2) と 1 つの出力側ハンドシェイク (reqo-acki) を持ち，入力側ハンドシェイクの 2 つの Request 信号のアービトレーション機能と，入出力ハンドシェイクを結びつける機能を実現する．

2-1 アービタを用いると，以下に示すように，独立して動作する 2 つの送り側モジュール ($S1$, $S2$) と受け側モジュール (R) の間で，排他的な 4 サイクルハンドシェイクが可能である．

1. 先に入力した reqi1/2+ に対して reqo+ を出力
2. acki+ を入力して acko1/2+ を出力
3. reqi1/2− を入力して reqo− を出力
4. acki− を入力して acko1/2− を出力
5. 後続の reqi2/1+ に対して reqo+ を出力 (2 へ)

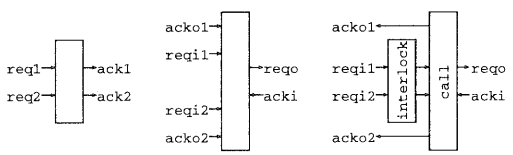

図 7: 相互排除要素と 2-1 アービタ

4.3.2 アービタモジュール

3 つ以上のハンドシェイクを排他制御するアービタモジュールは，2-1 アービタモジュールの組み合わせで実現できる．

BP 内では，5 つの入力ポートからのハンドシェイクの排他制御が必要である．そのため，4 つの 2-1 アービタを組み合わせて，5 つのハンドシェイク入力に対するアービタを構成している．要求の順番を対称的に制御するには，ツリー状の構成が適しているので，N, E, W, S のチャンネルについてはツリー状に対称化し，PP のチャンネルについてはカスケードにした 5-1 アービタツリーを用いている (図 8-a)．

5-1 アービタツリーを用いると，5 個の送り側 ($S1$, $S2$, $S3$, $S4$, $S5$) と 1 個の受け側 (R) の間で，“1 回の” 4 サイクルハンドシェイクの排他制御ができる (図 8-a)．

しかし，BP の動作はメッセージ処理が基本であり，最初のハンドシェイクを排他制御した後も，メッセージの処理が完了するまでの間，ロックした資源と連続した 4 サイクルハンドシェイクを行なえる機能が必要である．そこで，5-1 アービタツリーの入力にラッチ機能を持つモジュールを付加し，アービタにロック機能を付加したアービタモジュールを構成している (図 8-b)．

このアービタモジュールには，各チャンネル毎に，共通資源のロック動作を行なうための setreq-setack と，そのリリース動作を行なうための relreq-relack の 2 種類のハンドシェイクを設けている．

入力ポートは，共通資源を使用する場合，このアービタモジュールに対しロック動作のためのハンドシェイクを行なう．アービタモジュールは，複数の入力ポートからの *Request* 信号 (setreq) を受けて，先行したチャンネルに対し *Acknoledge* 信号 (setack) を返すと，共通資源のハンドシェイク回路を選択するための，ロック制御論理 (lock1,lock2,lock3,lock4,lock5) を出力する．入力ポートは，アービタモジュールのロックを完了すると，共通資源に対してハンドシェイクを行なう．資源に対する動作を終了すると，アービタモジュールとリリースのためのハンドシェイクを行ない，共通資源を解放する．

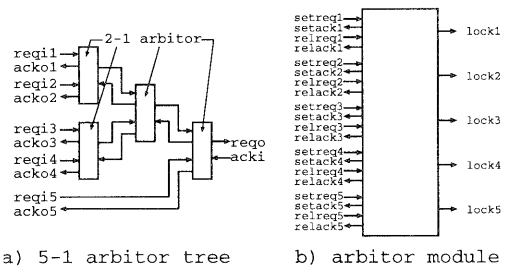

図 8: アービタモジュールの構成

5 PCA chip の実装について

PCA chip の実装は，0.35μ の CMOS スタンダードセルで行なっている．PCA chip 内には 6 × 6 個の PCA セ

ルアレイを実装し，LSIのダイサイズは$10mm \times 10mm$となっている．パッケージングは，chipの単純接続による拡張を考慮してQFPとしている．

セルアレイ外周部の20個のPCAセルでは，PPの外周部方向の入出力は終端しており，PPの信号線は外部との入出力には使用していない．また，同じく外周部方向のBPのハンドシェイク信号線についても，3/5に当たる12個のBPのハンドシェイク信号線のみを外部との入出力に使用している(図9)．

実装上の問題で，chipのパッド数に制限があるため，外部入出力にはこのような制限を設けている．しかし，PPの信号線については，LSI境界を跨るオブジェクトは，回路の遅延条件が複雑になり，オブジェクトの設計上，問題があるため，PPについては外部入出力を持たせる必要はないと考えている．また，PCAでのアプリケーション記述方法として，複雑で機能の大きいオブジェクトで実現するのではなく，単純な機能のオブジェクトを，BPのメッセージ通信機能で組み合わせて実現する方式が有効であるというわかってきている[9][10]．そのため，オブジェクトのサイズは，本chip内のセルアレイ上でも，十分実現可能なものとなっている．

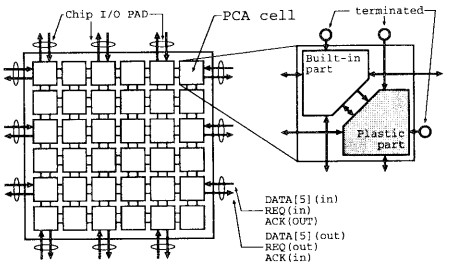

図9: PCA chipの外部入出力

6 まとめ

本稿では，現在開発中のPCA chipの設計について述べた．PCA chipの特徴は，布線論理の自律的かつ動的な再構成機能の実現，均質なハードウェア構造，そして，スケーラブルな拡張に対応するための非同期設計である．

PCA chipの非同期設計には，同期式回路と同じ遅延モデル，束データ方式を用いている．レジスタ間のデータ転送の制御には，Muller-C素子を用いたハンドシェイク回路を使用している．このハンドシェイク回路では，束データの遅延に相当する遅延素子を挿入して，タイミングの調整を行なっている．BPの機能は，セルアレイ上を独立のタイミングで流れるメッセージの処理が中心で，これをアービタモジュールによるハンドシェイク信号の排他制御とハンドシェイク回路の切り替えによるデータパスの構成変更で実現している．

現在，PCA chipの設計は論理設計等が終了しており，レイアウトと挿入遅延のバックアノテーションを経て，製造に移る予定である．今後，チップの完成を待って，動作評価，性能評価を行なう予定である．

謝辞

日頃より，PCA chipの設計に関して有用な議論を交わして頂いております，NTT未来ネット研究所プログラマブル通信処理研究グループの皆様，NTTエレクトロニクスカスタムLSI設計事業部の皆様に感謝致します．

参考文献

[1] K. Nagami, K. Oguri, T. Shiozawa, H. Ito and R. Konishi: "Plastic Cell Architecture: Towards Reconfigurable Computing for General-Purpose", The IEEE Symposium on FPGAs for Custom Computing Machines (1998).

[2] 小栗："VLSIアーキテクチャと設計自動化技術の将来", 信学技術報 VLD98-40 (1998).

[3] 伊藤, 小栗, 小西, N. Imlig："動的再構成可能計算機構のためのプラスティックセルアーキテクチャの設計", DAシンポジウム'98 (1998).

[4] 稲森, 中田, 小西, 小栗："再構成可能なセルの二次元アレイへの非同期回路配置配線法の提案", DAシンポジウム '99, pp. 219–223 (1999).

[5] 南谷："非同期式プロセッサ —超高速vlsiシステムを目指して—", 情報処理, **34**, 1, pp. 72–80 (1993).

[6] C. L. Seitz: "Introduction to VLSI Systems", chapter System Timing, Addison-Wesley (1980).

[7] 小西, 小栗, 伊藤, 永見："LUTアレイ用非同期論理回路の合成法", 信学技術報 VLD98-113, CPSY98-133, pp. 107–114 (1998).

[8] I. E. Sutherland: "Micropipelines", CACM, **32**, 6, pp. 720–738 (1989).

[9] 塩澤, 小栗, 永見, 伊藤, 小西："汎用計算機構を実現する再構成可能LSIアーキテクチャ", 信学技報 VLD97-114, ICD97-219, pp. 9–16 (1997).

[10] N. Imlig, T. Shiozawa, K. NAgami, R. Konishi and K. Oguri: "Communication Logic: Digital Circuit Compilation for the PCA Architecture", DAシンポジウム '99, pp. 101–106 (1999).