システムＬＳＩ設計技術　103－22
（2001. 11. 30）

PCAのLSI実装とシミュレーションに関する研究

溝田 庄三†　後藤 敏広†　竹本 潤一郎‡
小佐々 武志‡　柴田 裕一郎‡　小栗 清‡

†長崎大学大学院 生産科学研究科 電気情報工学専攻
‡長崎大学 工学部 情報システム工学科
〒852-8521 長崎県長崎市文教町 1-14

E-mail: †{mizota,tgoto}@cis.nagasaki-u.ac.jp,
‡{b698433,b698420,shibata,oguri}@cis. n agasaki-u.ac.jp

概要： PCA（Plastic Cell Architecture）はFPGA(Field Programmable Gate Array)にさらに汎用性を持たせた新しいアーキテクチャである.既に最初のPCAチップはNTTにより実装されており，その実現可能性は示されている．我々は既存のPCAにはまだ実装されていないOSの機能などを組み込んだ，より完成度の高いPCAの実現を目指している．本稿では，そのために必要なVerilog-HDLに基づくシミュレーション環境の実装について述べる．また，並行して行っているPCAの可変部のレイアウト設計との連携についても述べる．
キーワード： PCA，FPGA，Verilog-HDL

A Study on LSI Implementation and a Simulation Framework of PCA

Shouzou Mizota†, Toshihiro Goto†, Jun'ichirou Takemoto‡,
Takeshi Kozasa‡, Yuichiro Shibata‡, Kiyoshi Oguri‡

† Department of Electrical Engineering and Computer Science,
Graduate School of Engineering Science, Nagasaki University

‡ Department of Computer and Information Sciences,
Faculty of Engineering, Nagasaki University
1-14 Bunkyo-machi, Nagasaki, 852-8521, Japan

E-mail: †{mizota,tgoto}@cis.nagasaki-u.ac.jp,
‡{b698433,b698420,shibata,oguri}@cis.nagasaki-u.ac.jp

Abstract : PCA(Plastic Cell Architecture) is a novel architecture that enhances FPGAs in terms of flexibility and generality . The first implementation of PCA has already been fabricated as real LSI by NTT. Our goal is to implement more highly complete PCA compared to existing one by incorporating functions of a operating system. In this paper, implementation of simulation framework based on Verilog-HDL that is essential for design of a new PCA chip. A data sharing process between the simulation and layout design of a plastic part of PCA is also described.
Key words : PCA, FPGA, Verilog-HDL

1 はじめに

現在様々なLSIが開発，設計されているが，その中でも製造後に布線論理，すなわちハードウェアをプログラムすることができるFPGA(Field Programmable Gate Array)は，ハードウェアが固定的であるという既成概念をくつがえすもので，多くの可能性を持っていると考えられる．

このFPGAの汎用性をさらに増し，よりコンピュータのCPUとメモリの関係に近づけたPCA（Plastic Cell Architecture）についての研究が進められている[1]．このPCAとは，ルーティングネットワークの中にメモリとFPGAをちりばめたアーキテクチャになっており，処理の本質となっているメモリ上のデータ間の加工・転送を，CPUを介せずにFPGA上の論理回路とルーティングネットワークにより，直接かつ並列的に行うものである．このPCAはPCのCPUに相当する本能部，メモリ部分に相当する可変部よりなっている．

本研究室では既存のPCAよりも集積度が高く，OS機能も含めたより完成度の高いPCAの作成を目指している．本稿ではそのPCA作成のためのテスト環境として構築したVerilog-HDLによるPCA全体のシミュレーションについて報告する．また，PCAのメモリ部分である可変部のフルカスタム設計との連帯についても述べる．

2 PCAの概要

2.1 基本構造

PCAとは構成可能な可変部，そしてこれを制御する本能部よりなるプラスティックセル（図1）を2次元のメッシュ状に並べたもの（図2）である．

プラスティックセルの可変部は隣接する可変部と回路を作るために接続され，本能部は隣接する本能部と通信するために接続される．また，本能部と可変部は本能部から可変部を制御するために，本能部が可変部に通信要求を送るために，可変部が本能部に通信要求を送るために接続される[2]．

メモリとプロセッサの組み合わせではプロセッサ（すなわちPCAでは本能部）で処理が行われるのに対し，PCAでは回路，すなわち可変部にて処理が行われるため，処理を行う部分を動作中に増やす事ができる．

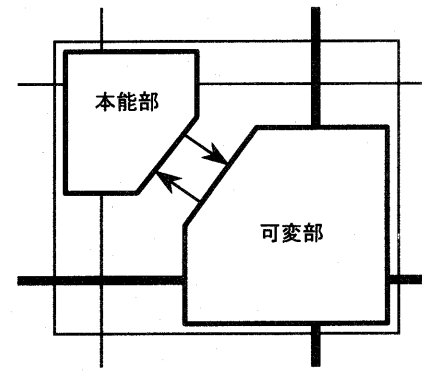

図1 プラスティックセル

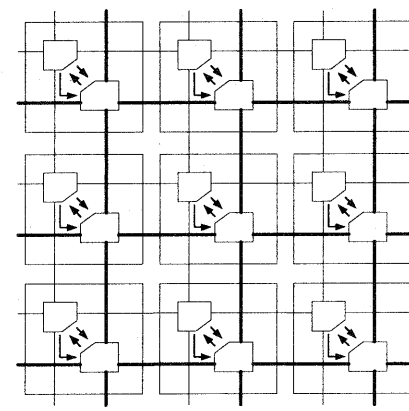

図2 PCA（Plastic Cell Architecture）

2.2 可変部の構造

プラスティックセルの可変部はさらに8×8のメッシュ構造になっており，そのメッシュの単位を基本セルという．この基本セルは隣接する4つの基本セルと入力，出力ともに1ビットで接続する．この基本セルには64ビットの記憶素子でできており，これで4つのLUT（Look Up Table）を構成している．この1つのLUTは入力4ビット，出力1ビット（図3）である．すなわち可変部にはこのLUTに適切な値を

書き込むことにより意味のある回路を構成できる．

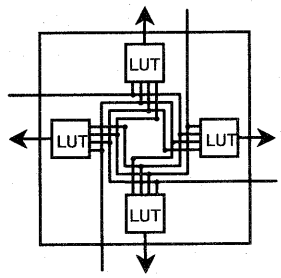

図3　PCA の基本セル

2.3　本能部の構造

プラスティックセルの本能部には，独立に動作する 5 つのステートマシン（入力ポート）があり，これらは図 2 上でルーティングスイッチを形成するとともに可変部の制御を行う．5 つの入力ポートは同一構造でそれぞれ，東西南北あるいは可変部からの要求を受け付け，入力ポートの状態に応じた処理を行う．要求は 11 種のコマンドで表現され，これらは 5 ビットでコーティングされている．そのコマンドの一覧を図 4 に示す．

clear	0b00000	メッセージの最後である
pp_open	0b10000	可変部との接続を開けよ
pp_close	0b10010	可変部との接続を閉じよ
config_in	0b10110	構成データを入力せよ
config_out	0b10100	構成データを出力せよ
coci	0b10101	コピーメッセージの出力
west	0b11000	以後のデータを西へ
north	0b11001	以後のデータを北へ
east	0b11010	以後のデータを東へ
south	0b11011	以後のデータを南へ
pp_out	0b11100	以後のデータを可変部へ

図4　コマンド一覧

2.4　コマンド

● 経路の設定と開放

11 種のコマンドのうち west コマンド，north コマンド，east コマンド，south コマンド，pp_out コマンドは経路を設定するためのコマンドで，入力ポートはこれらのコマンドを初期状態で受け取ると，これらのコマンドを消費するとともに後続のデータを指定された向きへ転送する，すなわちシフトレジスタとなるように設定される．clear コマンドは，その入力ポートでは消費されずに経路設定された方向に伝播されて，次々とシフトレジスタを構成していた入力ポートの初期化を行う．

●回路やデータの書き込み

config_in コマンドを受け取ると，入力ポートはこれを消費したのち，後続のデータ，4 ビットを 1 語として 1024 語を，可変部の記憶素子に書き込み，初期状態へ戻る．config_in コマンドと経路設定コマンドを組み合わせて用いることにより複数のプラスィックセルの可変部にデータを記憶させることができる．

● データの読み出し

config_out コマンドを受け取ると，これを消費するとともに config_out コマンドを受け付けたことを記憶する．この記憶は clear コマンドを受け付けたときの動作にのみ影響を与える．通常であれば clear コマンドを経路設定された方向へ転送して初期状態へ戻るが，config_out コマンドを受け付けた記憶がある場合には，まず可変部の記憶を経路設定された方向へ送り出してから clear コマンドを転送し初期状態へ戻る．

● 回路の動作開始と停止

pp_open コマンドは config_in コマンドで書き込んだ回路にリセット信号を送り回路の初期化を行う．その後本能部と可変部の回路との接続を行う．pp_close コマンドは本能部と可変部の回路との接続を切断する．

● 可変部のコピーコマンド

coci コマンドは config_out コマンド同様 clear コマンドを受け付けた時の動作にのみ影響を与える．coci コマンドを受け付けた記憶がある場合に clear コマンドを受け付けると ci コマンドを経路設定された方向へ送り出す．その後可変部のデータを経路設定された方向へ送り，さらに自身を経路設定した経路コマンドを送る．こうして可変部のコピーを作ることができる．そして最後に clear コマンドを送り初期状態へ戻る．

3 PCAのシミュレーション環境の構築

PCAは非同期回路であり，PCAの本能部及び可変部のシミュレーション環境は全てVerilog-HDLを用いて設計した．

3.1 プラスティックセルのVerilogモデルの実装

今回は，図5に示す構造を持つプラスティックセルのVerilogモデルを実装した．1つのプラスティックセルは5つの入力ポートと5つの出力ポート及びLUT，arbiter，or回路で構成されている．5つの入力ポートから各方向へ向かう配線は出力ポートでまとめられ隣接の入力ポートに接続している．LUTはPCAの可変部にあたる部分で4096ビットの記憶素子を持っており，この記憶素子に適切な値を書き込むことによって意味のある回路として構成される．入力ポートでは同時に実行されるコマンドが競合することがあり，その場合，待たされるコマンドが発生する．このためコマンド，データの受付はハンドシェークで制御される．また入力ポート，出力ポートにはarbiterが接続されており，入力ポートがarbiterに対してその出力ポートを使用して良いか，またLUTを使用して良いかの要求を出し出力ポートとLUTの使用の制御を行っている．このため1つのin_portが複数のout_portへ接続されることも，1つのout_portが複数のin_portへ接続されることもない．今回の設計ではc素子やand回路などの回路はVerilogの動作記述で設計しているがそれ以外は構造記述での設計となっている．今回のプラスティックセルのVerilog記述は本能部は1837行，可変部は570行の記述となった．

3.2 可変部への回路の書き込み

まず本能部でconfig_inコマンドを受けつけ，後続の回路の構成データ，つまり4ビットを1語として1024語を，可変部の記憶素子に書き込み，最後にpp_openコマンドを受けつけることにより書き込んだ回路にリセット信号を送って初期化を行い本能部と可変部の回路接続を行う．また可変部には回路を構成するデータだけではなく任意のデータを記憶させることにより，メモリとして使用することもできる．

今回はNTTより支給されたより抽象度の高いPCASIMという設計・シミュレーションツールを用いて設計した自己増殖するbiosome回路をVerilogのシミュレータ上で書き込みを行い動作確認を行った．この時PCASIMにより作られた回路データ(cirファイル)をVerilogのシミュレータ上で読み込ませるためにはファイルの形式を変換する必要がある．その変換は図6のようになり，この変換をするためのプログラムはC言語を用いて作成した．

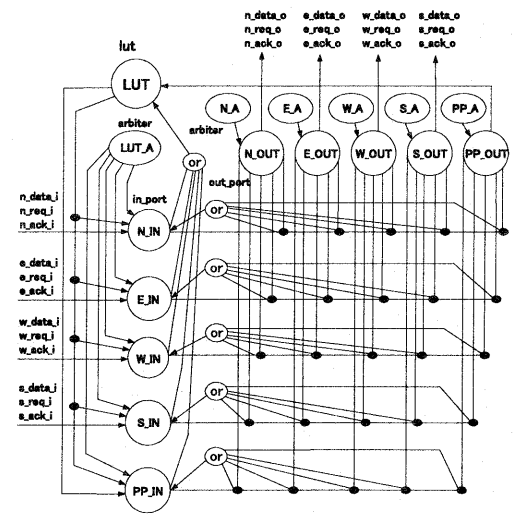

図5 プラスティックセルの構造

```
0 0 0 0 0 cccc
1 0 cccc 0 0 f0f0
2 0 f0f0 f0f0 cccc 0
3 0 f0f0 0 cccc aaaa
4 0 f0f0 0 cccc aaaa
5 0 f0f0 0 cccc aaaa
6 0 f0f0 aaaa cccc aaaa
.
.
.
```

変換 →

```
10000
10000
10000
10000
11100
11100
11100
11100
10000
10000
.
.
.
```

図6 cirファイルの変換

3.3 テストフィクスチャの記述

可変部に biosome 回路を書き込み動作確認をするために図 7 のようにテストフィクチャを記述した．まずリセット信号を与え，その後変換した biosome の cir ファイルのデータを読み込ませ，書き込みを行うと言った記述となっている．この時読み込む cir ファイルの先頭には書き込みのための ci コマンドと後尾には本能部と可変部の回路接続のための open コマンドのデータを記述している．

3.4 シミュレーション結果

ci コマンドを実行し biosome 回路が書き込まれた後 pp_open コマンドが実行されると biosome 回路は自己増殖を行うため，自ら本能部に coci コマンドを出力した後 esat コマンド，clear コマンドを出力していることが確認できた．これは PCASIM のシミュレーション結果と同じものとなった．また今回のシミュレーションは Verilog モデルでは CPU は UltraSPARC-Ⅱi×1，周波数 440MHz，メモリ 1G，OS は Solaris8 を用いてシミュレーションを行った．PCASIM の方は CPU は Athron，周波数 1G，メモリ 128M, OS は Windows Me を用いてシミュレーションを行った．この場合 biosome 回路を書き込むのに Verilog モデルでは 5 分 44 秒かかるのに対し PCASIM では 0.26 秒かかった．

```
initial begin

        rst <= 1'b0;
        s_req_i <= 1'b0;
        s_data_i <= 5'b00000;

    #1000
        rst <= 1'b1;
    #1000
        rst <= 1'b0;


        $readmemb( "biosome.cir" , pattern );

        for( i=0; i<4000; i=i+1  ) begin
            pat=pattern[i];

            #14000

                        s_data_i <= pat[4:0];
            #500
                        s_req_i <= ~s_req_i ;
            #1000
                        s_req_i <= ~s_req_i ;

        end


        $shm_close();
        $finish;
```

図 7　テストフィクスチャの例

4　レイアウト設計との連帯

ＰＣＡの可変部は前述のように８×８のメッシュ構造に，そしてそのメッシュ状に並んだ構成を持つ．各基本セルは４つのＬＵＴよりなっており，それぞれのＬＵＴは１６ワード×１ビット，すなわち４入力１出力のメモリとなっている（図 3）．

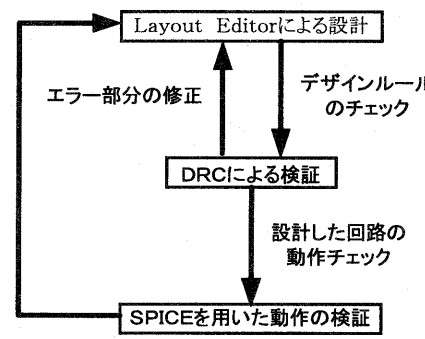

図 8　製作の流れ

このＬＵＴの部分は，集積度を上げるために図 8 に示す手順によってフルカスタム設計を行っている．現在のところ，LUT 基本メモリセルのレイアウトが終了しており，SPICE による遅延評価も行っている．今後この遅延情報を Verilog 側にバックアノテーションすることにより，より正確なチップの動作を検証できる環境を構築する必要がある．このためレイアウトからの遅延抽出と Verilog へのアノテーションを自動化する枠組みについて検討している．

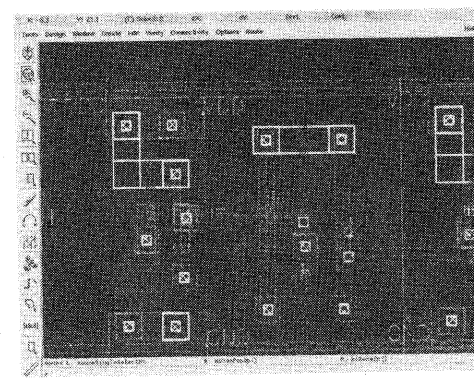

図 9　LUT のレイアウト

5 まとめと今度の予定

Verilogのシミュレーション環境を構築し，NTTによるPCAレイアウトエディタであるPCASEで作成した回路をVerilogのシミュレータ上で動作確認させることができるようになった．

今後は，今回構築したVerilogのシミュレーション環境では構造記述の部分が多いためその部分を動作記述に変え，コマンドが増えた場合などにも簡単に対処できるよう全体をよりコンパクト化する．

また，LUTのレイアウト設計が完成し次第遅延情報等のデータをVerilogシミュレータにフィードバックし，より繊細な評価が行えるようにしていく予定である．

参考文献

[1] 小栗清．布線論理による新しい汎用情報処理アーキテクチャPCA．January 2000/Vol.32,NO1/**bit**

[2] 永見康一，塩澤恒道，小栗清．自律再構成可能アーキテクチャ．情処研報 **Vol.98**, **NO.10**（98－DA－87），pp．17－24，情報処理学会研究報告1998年1月