

動的リコンフィギャラブルプロセッサ MuCCRA の結合網に関する研究

加東 勝† 佐野 徹† 天野 英晴†

† 慶應義塾大学工学部

〒 223-8522 神奈川県横浜市港北区日吉 3-14-1

E-mail: †muccra@am.ics.keio.ac.jp

あらまし MuCCRA(Multi-Core Configurable Reconfigurable Architecture) プロジェクトは、コンフィギャラブルな低電力マルチコア動的リコンフィギャラブルプロセッサにのアーキテクチャについて研究をしている。そのプロジェクトで開発された動的リコンフィギャラブルプロセッサ MuCCRA は多数の演算要素 (PE) から成り、その PE 同士を繋ぐ結合網は、そのアーキテクチャ全体の面積、電力やアプリケーション性能に影響を与える。そこで、本論文では、動的リコンフィギャラブルプロセッサ MuCCRA において、直結型、アイランドスタイルまたはその両方の接続を持つハイブリッドの 3 種類の結合網を設計する。そして、PE アレイの大きさを変えて、その面積、電力、アプリケーション性能の評価を行う。その結果、アルファブレンダにおいて、柔軟性が一番高いハイブリッドの結合網を持つ MuCCRA の実行時間が短い、面積が最大で約 25%、消費エネルギーでは約 15%、直結型の MuCCRA より大きかった。

キーワード 動的リコンフィギャラブルプロセッサ、結合網

Research on an Interconnection Network of the Dynamically Reconfigurable Processor: MuCCRA

Masaru KATO†, Toru SANO†, and Hideharu AMANO†

† Faculty of Science and Technology, Keio University

3-14-1, Hiyoshi, Kohokuku, Yokohama, 223-8522, Japan

E-mail: †muccra@am.ics.keio.ac.jp

Abstract In the MuCCRA(Multi-Core Configurable Reconfigurable Architecture) project, an architecture of configurable low-power multi-core dynamically reconfigurable processor has been investigated. A dynamically processor called MuCCRA is consisting of a number of PEs, and its interconnection of PEs gives a large effect on the total area, energy and performance. In this paper, we design three types of interconnection; direct connection, island-style connection and hybrid connection with three different PE array sizes, and evaluate these area, energy and performance. As a result, a MuCCRA with hybrid interconnection which has the highest degree of flexibility is able to execute alpha-blend faster than other MuCCRAs, but requires more area by 25% and consumes higher energy by 15% than MuCCRA with direct interconnection.

Key words Dynamically Reconfigurable Processor, Interconnection Network

1. はじめに

近年、高機能なモバイル機器を対象とした組み込みハードウェアの開発において、設計期間の長期化、コストの増大が問題となっている。この問題を解決するハードウェアとして、動的リコンフィギャラブルデバイスが注目されており、様々な研究が行われている [1]。動的リコンフィギャラブルデバイスは回路構成情報 (コンテキスト) を内部に保持し、これを実行時に

切り替えることにより様々な回路を実現できるプログラマブルデバイスである。この動的リコンフィギャラブルデバイスは、アレイ状に配置した多数の 8bit から 32bit の粗粒度の演算器 (Processing Element, PE) やメモリを持ち、高い並列処理能力を持っている。これに加えて、回路構成情報を追加、変更するだけで、様々な処理を行えるという拡張性、柔軟性を併せ持つため、ハードウェアの開発期間やコストの削減が期待されている。また、この動的再構成機能により、同一のアレイを多

様な目的に利用可能なことから、従来の Field Programmable Gate Array(FPGA) に代表される細粒度のリコンフィギャラブルデバイスの課題である面積効率も改善されることが期待できる。

近年では、実際の商用のデバイスを用いたアプリケーション開発やアーキテクチャ検討が進んでいる。これらの実用的なデバイスを用いて様々なアプリケーションを実装、評価することで、動的リコンフィギャラブルプロセッサの PE の粒度、アレイのサイズ、コンテキスト数、接続アーキテクチャは、アプリケーションによって向き不向きがあること、大規模なサイズのアレイを用いるよりも、複数の小規模なサイズのアレイを用いたマルチコア構成が有利なことがわかってきた。

MuCCRA プロジェクトは、これらの経験を踏まえ、低消費電力マルチコア動的リコンフィギャラブルプロセッサを、応用目的に合わせて最適な構成を生成、選択するためのアーキテクチャ技術をチップレベルから検討することを目的としている。我々は、このプロジェクトにおいて、これまで MuCCRA-1、MuCCRA-2 と呼ばれるチップを開発し、その性能、面積、消費電力について検討している [2]。

アーキテクチャ設計において演算要素 (PE) 間の結合網はアプリケーションの性能を左右する重要な要素である。MuCCRA-1、MuCCRA-2 では PE 間の結合網に FPGA およびいくつかの動的リコンフィギャラブルプロセッサに用いられているアイランドスタイルを採用している。この方式は、データ移動が柔軟に行えるが、データパスが長くなり最大遅延が大きくなる傾向がある。

動的リコンフィギャラブルデバイスの PE 同士の接続法として、もう一つ一般的な方式として直結型がある。この方式は、PE 間に直結リンクを設けてデータ転送を行う方式で、現在の商用あるいは実験用の動的リコンフィギャラブルプロセッサのいくつかはこの方式を用いている。

目的に応じて最適のアーキテクチャを生成するコンフィギャラブルな環境を実現するためには、その PE 同士の接続を変化させた場合の評価をとる必要がある。そこで本論文では、PE 間を直接接続する直結型、FPGA の結合網のようなアイランドスタイル、またアイランドスタイルに加えて PE 同士の直結のリンクを持つハイブリッドタイプの 3 種類の結合網を動的リコンフィギャラブルプロセッサの MuCCRA に実装し、その面積、電力、性能の比較評価を行う。また、PE アレイのサイズを変えた場合の評価も行う。

本論文の構成は、以下の通りである。次節では、商用化された動的リコンフィギャラブルプロセッサの結合網に注目した関連研究を述べ、3 節では、3 種類の結合網を適用した MuCCRA のアーキテクチャについて説明する。4 節では、それぞれのアーキテクチャの実装を述べ、面積等の比較を行い、5 節では、アプリケーションの性能評価、電力評価を述べ、6 節では結論を述べる。

2. 関連研究

動的リコンフィギャラブルプロセッサの PE アレイで用いられる結合網で、代表的なものはアイランドスタイルと直結型である。アイランド方式は、縦横にバスを張り巡らせ、交点に

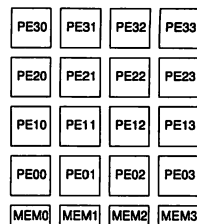


図 1 MuCCRA の基本構造

スイッチを置き、接続をコンフィギュレーションデータで制御できるようにする。PE とバス間は、コネクションブロックを用いて、データを取り込み、あるいはそこにデータを出力する。この方式は FPGA では最も一般的に使われ、動的リコンフィギャラブルプロセッサでは、初期の製品である Chameleon CS2112 [3]、NEC 社が開発した Dynamically Reconfigurable Processor(DRP) [4]、PACT Xpp [5] はこのタイプに分類される。アイランドスタイルは、PE 間を柔軟に接続可能な利点を持つ一方、配線遅延が大きく、配線とスイッチのための面積の消費が大きいなどの問題点がある。

これに対して、PE 同士を直接リンク接続する方法も広く用いられている。これらの方法ではメッシュ型が最も広く用いられている。日立製作所が開発した Flexible Engine/Generic ALU array (FE-GA) [6] は算術論理演算器 (ALU) と乗算器 (MLT) のセルを合わせて 32 個持ち、それを 2 次元アレイ状に並べた構成をとっている。各セルは隣接するセルに接続されている直結型となっている。IMEC 社の Architecture for Dynamically Reconfigurable Embedded System (ADRES) [7] も 32bit 粒度の Reconfigurable Cells (RCs) が直接接続された直結型であるが、こちらは、隣接接続の他に遠隔 PE との接続を行う特殊なリンクを設けている。直結型は、リンクで接続された PE 同士は 1 クロックでデータ転送が可能であるため、高い周波数で動作が可能である一方、遠隔 PE に転送を行う際に時間がかかる。

動的リコンフィギャラブルプロセッサが FPGA とマルチプロセッサの中間的な性格を持つとすれば、アイランドスタイルは FPGA 的であり、直結型はマルチプロセッサ的であると考えることができる。

3. MuCCRA

3.1 MuCCRA の概観

MuCCRA は、図 1 に示すように、PE がアレイ状に並び、データメモリ (MEM) が PE アレイの下側にあるのが基本的な構造である。各モジュールはそれぞれ独自のコンテキストメモリを持っており、そこに格納されている構成情報を毎サイクル呼び出すことにより、実行中に動的に再構成することが可能である。

今回設計した 3 種類の MuCCRA も図 1 と同じような構造を持っており、各 PE は 16bit 構成になっている。今回設計した 3 種類の内の 1 つは各 PE 同士が直接接続された直結型 (MuCCRA-D) である。2 つ目は、FPGA のようなアイランドスタイルの結合網 (MuCCRA-I) を持ち、3 つ目は、このアイランドスタイルの結合網にさらに PE 同士に数本の直結のリ

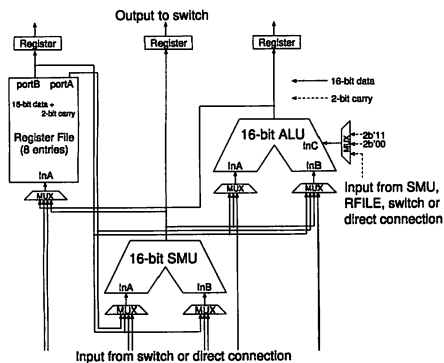


図2 演算部 (PE Core)

リンクを持たせたハイブリッドの結合網 (MuCCRA-H) を持っている。

各アーキテクチャの結合網は異なるが、演算を行うモジュール、データを保存するデータメモリ、そしてコンテキストの制御を行う機構は各アーキテクチャとも共通である。そこで、最初に各アーキテクチャの共通モジュールについて説明する。

3.1.1 演算部 PE Core

各 PE は演算を行う PE Core、データのやりとりを行うスイッチ (SW) と構成情報を保持するコンテキストメモリから構成されている。

PE Core はさらに図 2 のように、加減算、乗算、比較や論理演算を行う ALU、シフトや定数補給を行う SMU、そして、8 エントリのレジスタファイル (RFILE) から構成されている。各ユニットへの入力は結合網の種類に応じて異なる。直結型の場合は隣接する PE の出力等、アイランドスタイルの場合はスイッチに入力された値を用いることができる。また ALU の入力には SMU の出力を接続することは可能だが、この逆は許さないことで、組み合わせループが生成されることを防いでいる。一方、RFILE は、ALU、SMU の入出力の全てに接続されている。各ユニットの出力は図 2 に示すように、レジスタによりバッファリングされている。これは PE をまたいで演算を行う長いパスができるのを防ぎ、最大動作周波数を上げるためである。

3.1.2 データメモリ

データメモリは PE アレイの下側にあり、各メモリは 1-Read, 1-Write の SRAM であり、エントリ数は 256 でビット幅は 16bit である。各メモリは、そのメモリの上に存在する PE からデータとアドレスを取得することができる。またアイランドスタイルの結合網を持つ MuCCRA-I と MuCCRA-H においては、PE アレイの一番上の PE 内のスイッチからもアドレスとデータを取得できるようにループバック線が存在する。

3.1.3 コンテキスト制御

各アーキテクチャのコンテキストの切り替え制御は、単純なカウンタベースの Context Switching Controller (CSC) モジュールによって行われる。PE、MEM などの再構成モジュールは、CSC が生成するコンテキストポインタに従って、各モジュールが保持するコンテキストメモリから、自身の構成情報を読み出して再構成を行い動作する。CSC 自身も再構成モ

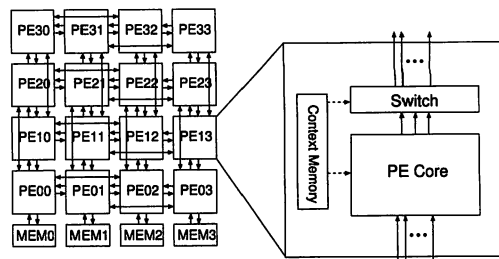


図3 MuCCRA-D

ジュールであり、コンテキストメモリから構成情報を読み出し、コンテキストを切り替える。

コンテキストポインタは、分岐しない場合は単純にコンテキストカウンタにより、インクリメントされていくが、コンテキスト分岐する場合は、特定の PE からの相対分岐アドレスをコンテキストカウンタの値に加算する。MuCCRA では、PE アレイの右端の 4 つの PE を Special PE として、この PE の出力を相対分岐アドレスとして取得することができる。

MuCCRA ではアプリケーションをタスクと呼ばれる単位で実装し、タスク管理は、Task Configuration Controller (TCC) モジュールが行う。TCC は内部に全ての構成情報を格納する中央メモリを持ち、TCC は実行するタスクに必要な構成情報のみを各モジュールのコンテキストメモリに転送する。各モジュールへの転送方式は、RoMultiC [8] を利用することでマルチキャストによる高速化を実現している。

3.2 MuCCRA-D (直結型)

MuCCRA-D は PE 同士が直接つながった直結型の結合網を持つ MuCCRA である。アーキテクチャ全体において長いパスができず、高周波数を実現することができる。今回実装した MuCCRA-D では、データ移動の柔軟性を向上させるため、図 3 に示すように、縦横に 1 つ飛ばしのリンクを設けてある。これにより、1 つ先の PE と 1 サイクルでデータのやりとりが可能になっている。隣接する PE 同士とは 3 本の独立したバスで繋がっている。PE アレイの下側にある MEM はその上にある PE とのみ接続されており、その PE からアドレスおよびデータを取得することができる。

3.2.1 MuCCRA-D の PE と Switch

MuCCRA-D の PE は図 3 に示すように、演算を行う PE Core、構成情報を保持するコンテキストメモリ、演算結果をどの PE へ出力するかを決定する Switch から構成されている。

PE Switch は ALU、SMU、RFile の出力を、それぞれ構成情報を元に隣接する PE または、2 個隣の PE へと転送する。各出力は構成情報によって、転送先の PE を決定すると共に転送先の PE の ALU、SMU、RFile のどのユニットの入力へ接続するかも決定する。そのため、2 つ以上の出力を同じ転送先の PE の同じユニットへの入力に接続することはできない。転送先のユニットが異なれば、同じ PE への転送は可能である。また、各出力が転送できるのは 1 箇所だけである。2 個隣の PE へ転送した場合は、そのデータは転送先の PE の ALU、SMU、RFile のすべてのユニットの入力に接続される。

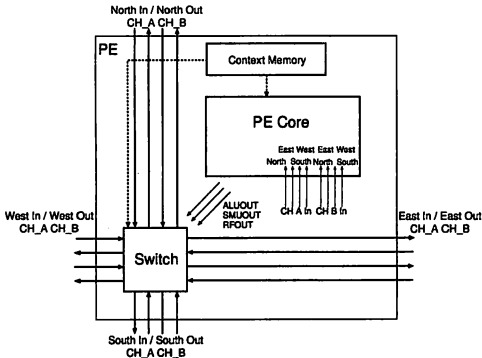


図 4 MuCCRA-I の PE

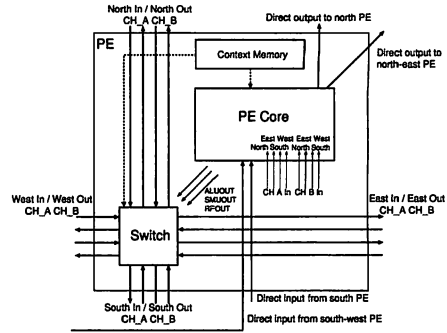


図 6 MuCCRA-I の PE

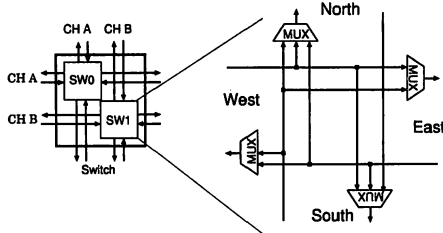


図 5 MuCCRA-I/H の Switch

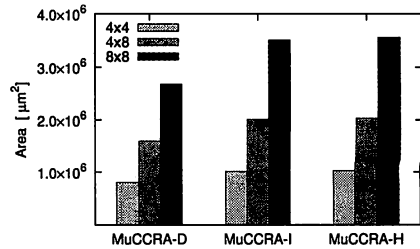


図 7 全体の面積

3.3 MuCCRA-I(アイランドスタイル)

MuCCRA-Iは図4に示すように、FPGAのようなアイランドスタイルの結合網を持つ、MuCCRAである。結合網にデータを渡すことで、遠くのPEともデータのやりとりが可能であるが、一方で、結合網に長いパスができ、高い周波数を実現するのが難しいといった欠点がある。また結合網のチャンネル数の制限により、データ転送がうまくいかないこともある。今回実装した、MuCCRA-Iでは結合網のチャンネル数は2となっているが、それぞれ双方向に別のデータを転送できるように独立の転送リンクを持っている。

3.3.1 MuCCRA-IのPEとSwitch

MuCCRA-IのPEもMuCCRA-Dと同様に、PE Core、コンテキストメモリ、スイッチから構成されている。しかし、図4に示すように、結合網がアイランドスタイルであるため、スイッチは図4のような位置関係になっている。PE Coreには東西南北からのデータを入力することができる。PE Coreの出力は一旦レジスタに格納され、スイッチに渡される。

スイッチはマルチプレクサにより構成されており、図5のような構造になっている。スイッチは構成情報を元に、東西南北から入ってきたデータおよび、PE Coreからの演算結果を選択し、各方向に出力する。ただし、結合網内でループができることを回避するために、北からの入力北東方向には出力できず、南方向にのみ出力できるようになっている。

3.4 MuCCRA-H(ハイブリッド)

MuCCRA-HはMuCCRA-Iの結合網に加えて、PEへの入力として南と南西にあるPE、出力としては北と北東にあるPEとつながった専用の直結リンクをもたせた構造になっている。これにより、近いPE同士のデータのやりとりにはこの直結リンクを用い、遠くのPEとのデータのやりとりには、アイラン

ドスタイルの結合網を用いることで、結合網のチャンネルの飽和を防ぎ、データのやりとりを柔軟に行うことが期待できる。

3.4.1 MuCCRA-HのPEとSwitch

MuCCRA-HもPE内部の構成はMuCCRA-Iとほとんど同じだが、図6のように、直結のリンクが存在する。各PEはそのPEの南と南西の位置にあるPEのALUとSMUの出力を直接取得することができる。南と南西の位置にPEが無い場合は、0が入力される。スイッチはMuCCRA-Iと同様マルチプレクサにより構成されている。

4. 実装

各アーキテクチャの実装にはFujitsu社の65nm CMOSプロセスを用いて実装した。設計にはVerilog-HDLを用い、VDECでサポートされるSynopsys社DesignCompiler 2007.12-SP3を用いて論理合成を行った。また、PEアレイの大きさは、4×4、4×8、8×8の3種類を各結合網において設計し、評価を行った。

図7に、合成後の各アーキテクチャの全体の面積を3種類のアレイサイズ毎に示す。図7より、ハイブリッドの結合網を持つMuCCRA-Hが一番面積が大きいことが分かるが、MuCCRA-Iとほとんど変わらない。また、MuCCRA-DとMuCCRA-I、MuCCRA-Hではアイランドスタイルの結合網を持つMuCCRA-IとMuCCRA-Hの方が約30%ほど面積が大きくなっている。これらの原因は、スイッチの構成情報のビット幅の問題で、アイランドスタイル内のスイッチにもコンテキストメモリが必要であるためである。一方で、MuCCRA-Dのスイッチには独自のコンテキストメモリはなく、PE Coreのコンテキストメモリと共有している。

MuCCRAでは図8に示すように、各アーキテクチャのメモリが占める割合が高いことが分かる。図8より、どのアーキテ

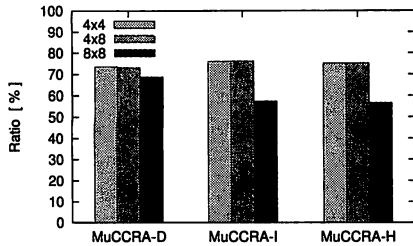


図 8 メモリが占める割合

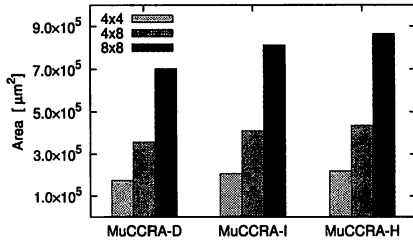


図 9 メモリを除いた面積比較

クチャにおいてもメモリが占める割合が 4×4 では 70% 以上と大きく、PE アレイのサイズが大きくなるにつれて、その割合が下がるが、50% 以上と大きい。各再構成モジュールが保持するコンテキストメモリが MuCCRA では支配的であり、アーキテクチャ全体の面積に大きく影響することが分かる。

そこで、次の図 9 に、各アーキテクチャのメモリを除いた、組み合わせ回路部分の面積を示す。メモリを除いた結果を見ても、柔軟性が高いハイブリッドの結合網を持つ MuCCRA-H が一番大きく、MuCCRA-D と比べて 4×4 の時、最大で約 25% 大きい。MuCCRA-I と比べても約 5% 大きい。

5. アプリケーションによる評価

5.1 評価アプリケーションと開発環境

実装した評価アプリケーションは、二画像を半透明合成するアルファブレンダフィルタである。アルファブレンダは、データメモリに格納された複数の RGB データに対して積和を行い、その結果をデータメモリに書き戻す処理を行うアプリケーションである。

アプリケーションの実装には、MuCCRA 向けに独自に開発した BlackDiamond Compiler [9] を使用した。図 10 に、BlackDiamond Compiler でアプリケーションを PE へマッピングした際の結果の GUI を示す。BlackDiamond Compiler は、C 言語ライクな記述で書かれたアプリケーションを自動的に PE へマッピングし、Verilog-HDL シミュレーションに必要な、構成情報ファイルを出力することができる。

5.2 実装結果

実装したアプリケーションのコンフィギュレーションデータと、合成後のネットリストを用いてシミュレーションを行った。そして、実行サイクル数および実行時間を測り、各アーキテクチャについて比較を行った。

表 1 に各アーキテクチャの 4×4 の PE アレイにアルファブ

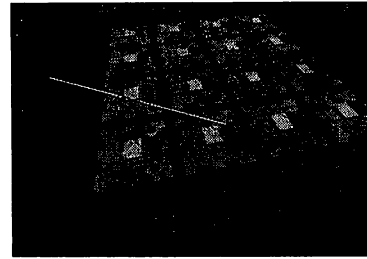


図 10 アプリケーション開発環境
BlackDiamond Compiler

レンダを実装した際の、必要コンテキスト数、構成情報の数、そして実行サイクル数および実行時間の結果を示す。実行クロック数には、外部との通信に要する時間は含まれていない。また、1 サイクルはどのアーキテクチャも共通で 24nsec(42MHz) とした。

表 1 アルファブレンダの実装結果

| | MuCCRA-D | MuCCRA-I | MuCCRA-H |
|-------------|----------|----------|----------|
| コンテキスト数 | 8 | 8 | 8 |
| 構成情報数 | 29 | 27 | 26 |
| サイクル数 | 54 | 52 | 51 |
| 実行時間 [nsec] | 1296 | 1248 | 1224 |

アルファブレンダは小規模なアプリケーションであるため、各アーキテクチャにおいて大きな差異は見られないが、データ移動の柔軟性が低い直結型の MuCCRA-D の性能が若干低くなった。

MuCCRA-H においては、直結のリンクを持っているため、隣接する PE と通信する際に、アイランドスタイルのスイッチを使用しないで、直結のリンクを使用することで、構成情報の数を減らすことができる。つまり、PE Core とスイッチの構成情報は PE の構成情報として一緒に転送されてくるため、スイッチを使用した場合、PE Core を使用していても、いなくてもどちらの場合でもその PE の構成情報が必要になる。しかし、直結のリンクにデータを流した場合、その直結の値を使うかどうかの構成情報は、値を受け取った方の PE の構成情報に含まれるので、転送側の構成情報は不要になる。そのため、MuCCRA-H において、直結のリンクを多く使用することで、MuCCRA-I より構成情報の数を減らし、実行時間を減らすことが期待できる。

どのアーキテクチャも演算の結果はレジスタによりバッファリングされているので、PE を複数またいで演算をするような長いパスはできないが、アイランドスタイルの結合網を持つ、MuCCRA-I と MuCCRA-H では、その結合網内に長いパスができることがあり、それによって、クリティカルパスが長くなる場合がある。合成の結果によると、アイランドスタイルのスイッチを通過するのに、450psec から 500psec かかる。そのため、 4×4 のアレイにおいては、最大で 18 個、 4×8 では最大で 34 個、 8×8 では最大 70 個のスイッチを 1 サイクルで通過可能で、それぞれ 9nsec、17nsec、34nsec 結合網で遅延生じる可能性がある。そのため、MuCCRA-I および MuCCRA-H ではアプリケーションを実装する際に、データ移動の柔軟性に重

きを置くと、遅延が大きくなり最大動作周波数が著しく小さくなる可能性がある。

5.3 消費電力

各アーキテクチャにおいて、アルファブレンダを実行した際のスイッチング確率を抽出し、Synopsys 社の PrimeTime 2007.12-SP3 を用いて電力の評価を行った。

表 2 に、4×4 の各アーキテクチャにおける、アルファブレンダを実行した際の単位時間あたりの動的電力とリーク電力、および消費エネルギーを示す。消費エネルギーは、単位時間あたりに消費される電力（動的とリークの合計）と表 1 の実行時間をかけた値である。尚、各アーキテクチャは 42MHz(24nsec) で動作し、想定する電圧は 1.2V である。

表 2 電力評価

| | MuCCRA-D | MuCCRA-I | MuCCRA-H |
|--------------|----------|----------|----------|
| 動的電力 [mW] | 16.29 | 19.99 | 20.51 |
| リーク電力 [mW] | 1.88 | 2.44 | 2.46 |
| 消費エネルギー [nJ] | 23.55 | 28.00 | 28.12 |

電力評価の結果においても、面積が大きかったハイブリッドの結合網を持つ MuCCRA-H の電力、消費エネルギーがともに大きかった。

表 3 に、アルファブレンダを実行した際の、16 個ある PE のうちの最大、最小の電力と各 PE の平均を示す。また、アーキテクチャ全体の電力で 16 個の PE が占める割合を示す。

表 3 PE に関する電力評価

| | MuCCRA-D | MuCCRA-I | MuCCRA-H |
|---------------|----------|----------|----------|
| 最大電力 [mW] | 1.23 | 1.58 | 1.57 |
| 最小電力 [mW] | 0.77 | 0.96 | 0.98 |
| 平均 | 0.91 | 1.15 | 1.18 |
| PE の電力の割合 [%] | 80.11 | 81.87 | 82.16 |

各 PE の演算部分は共通であるため、各アーキテクチャにおける電力の違いは、スイッチ、演算部分への接続および構成情報を保持するメモリの違いから生じる。直結型の MuCCRA-D における PE の最大電力時は、他の二つの PE の最大電力時と比べて、約 20% 低電力で動作することが分かる。平均電力を見ても同じくらい低い。尚、各アーキテクチャにおける、最小電力時の PE の状態は、各演算ユニットおよびスイッチが全く動作せず、常に 0 をだしている状態である。逆に、最大電力時は、演算部およびスイッチがともに動作している状態である。今回のアルファブレンダの実装においては、メモリに近い PE アレイ下側の PE の使用頻度が高く、これらの PE で最大の電力を使用し、逆に、メモリから遠い、PE アレイの一番上の PE で最小の電力を使用している。

また、表 3 より、PE アレイはアーキテクチャ全体の 8 割の電力を占めていることが分かる。残りの 2 割は、コンテキストの管理を行う制御機構、PE 同士を繋ぐ結合網に挿入されたバッファおよび、入出力バッファなどが考えられる。

6. おわりに

本論文では、PE 間を直接繋いだ直結型動的リコンフィギュラブルプロセッサ MuCCRA-D、アイランドスタイルの結合網を持つ MuCCRA-I とその両方の結合網を持つハイブリッドの MuCCRA-H の 3 種類のアーキテクチャにおいて、PE のアレ

イサイズを変えて、面積、電力、アプリケーションの性能の評価を行った。

実装結果より、直結型の MuCCRA-D はアーキテクチャのメモリを除いた面積でも、MuCCRA-I と MuCCRA-H と比べて、最大で 25% 小さいことが分かった。また、実装したアルファブレンダの実行結果より、実行時間はハイブリッドの MuCCRA-H の方が短かったが、電力および消費エネルギーは MuCCRA-D と比べて約 15% ほど大きかった。PE 単体の電力を見ても、直結型の結合網を持つ PE の方が電力が約 20% 小さかった。アイランドスタイルの結合網を持つ場合、データ移動の柔軟性は高いが、アレイサイズが大きくなるにつれて、結合網内の遅延が著しく大きくなるため、クリティカルパスが長くなり最大動作周波数が小さくなる可能性がある。一方で、直結型の場合は、アレイサイズによらず一定の動作周波数を維持できるが、遠隔の PE とのデータ通信には数サイクル必要となってしまうことが考えられる。

今後の研究は、アプリケーションを豊富に増やし、実際にレイアウトまで行い、レイアウト後のネットリストを用いて電力等の評価を行うことである。

謝 辞

本研究は、科学技術振興機構「JST」の戦略的創造研究推進事業「CREST」における研究領域「情報システムの超低消費電力化を目指した技術革新と統合化技術」の研究課題「革新的電源制御による次世代超低電力高性能システム LSI の研究」による。

また、本チップ試作は東京大学大規模集積システム設計教育研究センターを通し、株式会社半導体理工学研究所、(株)イー・シャトルおよび富士通株式会社の協力で行われたものである。

本研究を進めるにあたり、多忙の中たくさんの時間を割いてご助言いただいた Vasutan Tunbunheng 氏に感謝いたします。

文 献

- [1] 末吉, 天野 (編): “リコンフィギュラブルシステム”, オーム社 (2005).
- [2] H.Amano, Y.Hasegawa, S.Tsutsumi, T.Nakamura, T.Nisimura, V.Tanbunheng, A.Parimala, T.Sano and M.Kato: “MuC-CRA Chips: Configurable Dynamically-Reconfigurable Processors”, Proc. of ASSC 2007, pp. 384-387 (2007).
- [3] X.Tang, M.Aalsma and R.Jou: “A compiler directed approach to hiding configuration latency in Chameleon Processors”, Proc. of FPL, pp. 29-38 (2000).
- [4] M. Motomura: “A Dynamically Reconfigurable Processor Architecture”, Microprocessor Forum (2002).
- [5] M.Petrov, T.Murgan, F.May, M.Vorbach, P.Zipf and M.Glesner: “The Xpp Architecture and its co-simulation within the simulink environment”, Proc. of FPL, pp. 761-770 (2004).
- [6] 津野田, 高田, 秋田, 田中, 佐藤, 伊藤: “デジタルメディア向け再構成型プロセッサ FE-GA の概要”, 電子情報通信学会技術研究報告 RECONF, 105, 451, pp. 37-42 (2005).
- [7] F.J.Veredas, M.Scheeppler, W.Moffat and B.Mei: “Custom Implementation of the Coarse-Grained Reconfigurable ADRES Architecture for Multimedia Purposes”, International Conference on Field Programmable Logic and Applications, pp. 106-111 (2005).
- [8] V.Tanbunheng, M.Suzuki and H.Amano: “RoMultiC: Fast and Simple Configuration Data Multicasting Scheme for Coarse Grain Reconfigurable Devices”, Proc. of IEEE FPT, pp. 129-136 (2005).
- [9] V. Tanbunheng and H. Amano: “DisCounT: Disable Configuration Technique for Representing Register and Reducing Configuration Bits in Dynamically Reconfigurable Architecture”, Proc. of SASIMI 2007 (2007).