社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2004－ＳＬＤＭ－113 （７）
2004／1／22

リコンフィギャラブルプロセッサ DRP 上での エッジ近傍合成機能付き α ブレンダの実装

鈴木 正康† 山田 裕† 出口 勝昭† 安生健一朗†† 粟島 亨†††
天野 英晴†

† 慶應義塾大学大学院理工学研究科
〒 223-8522 横浜市港北区日吉 3-14-1
†† NEC エレクトロニクス株式会社
††† NEC マルチメディア研究所
〒 211-8668 川崎市中原区下沼部 1753
E-mail: †drp@am.ics.keio.ac.jp, ††k.anjo@necel.com, †††awash@ccm.cl.nec.co.jp

あらまし NEC エレクトロニクスが開発した Dynamically Reconfigurable Processor (DRP) は、粗粒度のリコンフィギャラブルプロセッサで、内部に持つ 16 のデータパスの構成情報を切替えることによって、様々な処理を実現する。本稿では、リコンフィギャラブルプロセッサ DRP 上でのエッジ近傍合成機能付き α ブレンダの設計事例を紹介し、DRP の処理能力を検証するため、Pentium 4、Athlon XP、DSP (TI C6713) などのアーキテクチャと比較した。その結果、並列処理の効果的な利用により、エッジ近傍合成機能付き α ブレンダを実行した場合、DRP は Pentium 4、Athlon XP の 3 倍、DSP の 17 倍 の処理性能を達成することができた。

キーワード リコンフィギャラブルプロセッサ、DRP、動的再構成、並列処理、α ブレンダ、エッジ近傍合成処理

Implementation of Anti-aliasing Alpha Blender on the Reconfigurable Processor DRP

Masayasu SUZUKI†, Yutaka YAMADA†, Katsuaki DEGUCHI†, Kenichiro ANJO††, Toru AWASHIMA†††, and Hideharu AMANO†

† Graduate School of Science and Technology, Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, 223-8522 Japan
†† NEC Electronics Corporation
††† NEC Multimedia Research Laboratories
1753 Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa, 211-8668 Japan
E-mail: †drp@am.ics.keio.ac.jp, ††k.anjo@necel.com, †††awash@ccm.cl.nec.co.jp

Abstract Dynamically Reconfigurable Processor (DRP) developed by NEC Electronics is a coarse grain reconfigurable processor that selects a data path from the on-chip repository of sixteen circuit configurations, or contexts, to implement different logic on one single DRP chip. This paper describes our implementation of an alpha blender with anti-aliasing capabilities on the DRP. Comparison with various architectures including Pentium 4, Athlon XP, and DSPs (TI C6713) are done to evaluate the potentials of the DRP. Our results show that the DRP outperforms Pentium 4 and Athlon XP by three times, and DSP by seventeen times when compared against the implementation of anti-aliasing alpha blender.

Key words Reconfigurable Processor, DRP, Dynamic Reconfiguration, Parallel Processing, Alpha Blending, Anti-aliasing

1 はじめに

近年、デバイス内部に複数の構成情報を保持し、それを高速に切替えることでハードウェアの構成を動的に変化させる技術が確立されつつある。中でもリコンフィギャラブルプロセッサは、固定ビット幅の演算器、シフタ、レジスタ、分散メモリからなる処理ユニットを多数搭載し、それらの処理ユニットの構成及び接続を短時間で切替えることにより、ハードウェアの高速性とソフトウェアの柔軟性を併せ持った機能を実現する。

特にリコンフィギャラブルプロセッサの研究と実用化が進み、様々なデバイス[7] [2] [8] [9] が登場している。高速な動的再構成技術により、一定の物理的資源を複数の処理に時分割して割り当てることによって、高い面積効率を達成することができるリコンフィギャラブルプロセッサは、画像、音声データのコード圧縮・復元、AES、DES 等の暗号化・復号化などの分野での応用が報告されている[4] [3]。

本稿では、エッジ近傍合成処理機能付き α ブレンダを DRP 上に実装した設計事例を紹介する。まず、2 節で DRP の詳細を述べた後、3 節で エッジ近傍合成機能付き α ブレンダについて説明し、4 節で DRP 上への実装についてまとめる。最後に 5 節で結果およびその評価について考察する。

2 DRP

Dynamically Reconfigurable Processor (DRP) [7] は NEC エレクトロニクスが 2002 年に発表した、粗粒度なリコンフィギャラブルプロセッサである。回路構成情報をチップ内部のメモリに格納することで、高速且つ動的に内部構成を変更でき、複数の処理を単一のチップで排他的に処理するシステムにおいては、面積を有効に利用できるデバイスである。

そのプロトタイプチップ DRP-1 は PCI インターフェース、乗算器、メモリモジュール、DRAM インターフェースなどを周囲に持つシステム LSI である。その構造は、図 1 に示す通りである。

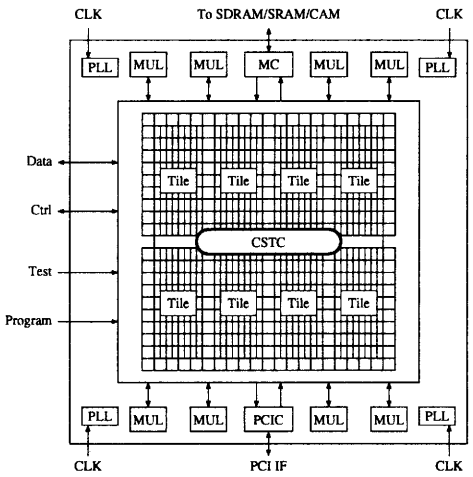

図 1 DRP-1 の構造

DRP-1 は Core 部分に Tile と呼ばれるリコンフィギャラブルユニットが 4 × 2 個配置されており、中央にチップ全体の回路構成 (状態遷移) を制御するためのシーケンサである Central State Transition Controller (CSTC) が配置されている。

各 Tile は図 2 に示す通り、8 × 8 の Processing Element (PE) アレイと状態遷移を行うシーケンサである State Transition Controller (STC) から構成されている。また、8bit × 256 エントリのメモリ 8 セットとこれらを制御するメモリコントローラ 2 セットを両側に持ち、8bit × 8192 エントリのメモリ 4 セットとメモリコントローラを Tile の上部もしくは下部に持つ。DRP 全体では 8bit × 256 エントリのメモリを合計 80 セット、8bit × 8192 エントリのメモリを合計 32 セット持つ。

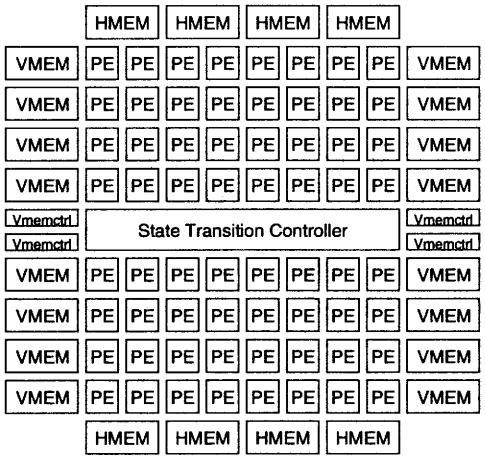

図 2 Tile の構造

各 PE は図 3 に示す通り、8bit の ALU、シフトやデータ制御、簡単な論理演算を行なう Data Management Unit (DMU)、8bit の Flip Flop、レジスタファイルから構成される。また、コンフィギュレーション時には命令データが命令メモリに書き込まれ、実行中に STC が発行した命令ポインタを命令メモリが受け、利用する命令データをロードすることでハードウェア構成を変更する。

DRP は、NEC が 1998 年に開発した DRL [6] 同様、1 クロックでコンテキストの切替えが可能なマルチコンテキストデバイスで、Tile 単位の部分再構成が可能である。一方で、DRL が Look Up Table (LUT) を構成要素とした細粒度のリコンフィギャラブルデバイスであったのに対し、DRP は 8bit の PE を構成要素としたリコンフィギャブルなプロセッサである点に大きな違いがある。その特徴をまとめると以下のようになる。

マルチコンテキスト機能: DRP は内部のメモリに最大 16 コンテキスト分の情報を蓄えることができ、最大で 5 コンテキストの中から次のコンテキストを選択し、動的に再構成をすることができる。次のコンテキストの選択は STC に信号を送ることで行なう。さらに、動作中に、切り替えの対象外のコンテキストに対してコンフィギュレーション ロードが可能である。こ

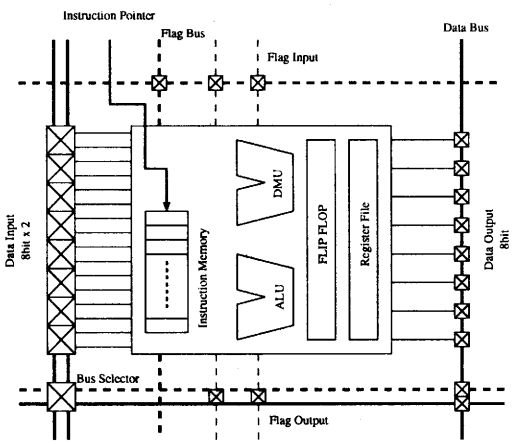

図 3　PE の構造

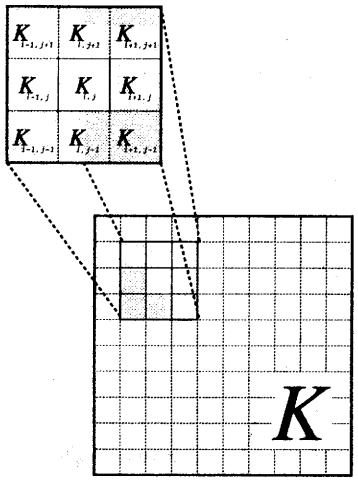

図 4　ピクセル $K_{i,j}$ とその近傍ピクセル

の特徴により、仮想ハードウェア [10] や動的適応ハードウェアの効率的な実現が可能となる。

プロセッサアレイ構成: DRP は FPGA 等の LUT によりロジックを実現する細粒度構成とは異なり、データパスを再構成するデバイスである。このため、 8bit を基本単位とした効率的なデータフローの制御および多桁演算の高速実装が可能である。LUT を書き換えるのに比べて高速な再構成が可能であり、無駄なクロックなしに、切り替えと同時に動作を行なうことができる。また、PE がチップ全体に配置されているため、並列処理を効率的に行うことが可能である。

パーシャルリコンフィギャラビリティ: DRP はコンテキスト切替えを Tile 単位で行うことができ、コンフィギュレーション データのロードは PE 単位で行なうことができる。これにより、Tile ごとに異なるコンテキスト構成を取ることができる。

メモリ拡散配置型のアーキテクチャ: DRP は各 PE が Flip Flop およびレジスタファイルを独立に持ち、Tile の両側に 8bit × 256 エントリの小規模なメモリを 8 セット、Tile の上側もしくは下側に 8bit × 8192 エントリのメモリを 4 セットずつ持つ。これらの記憶要素の分散により、パイプライン構成、データ駆動型制御、フィードバック構成を簡単且つ高速に実現することができる。

また、チップ内のメモリを異なる回路間で共有メモリとして見ることができるため、I/O 部分がボトルネックとならず、高速にデータのやりとりを行うことが可能である。

3　エッジ近傍の合成機能付き α ブレンダ

α ブレンダは二つの画像の直線補完を行う。直線補完とは任意の二つの点を合成し、一つの連続したデータに変換することである [1]。具体的には、m × n の画像 I_{mn} と J_{mn} について、合成画像の K_{mn} を生成する。K の任意のピクセル $K_{i,j}$ は次式で求めることができる。

$$K_{i,j} = \alpha I_{i,j} + (1-\alpha) J_{i,j}$$

$I_{i,j}$, $J_{i,j}$ はそれぞれ、画像 I, J の任意のピクセルであり、係数 α にもとづき画像 K を生成する。

エッジ近傍の合成処理 (アンチエイリアシング) は α ブレンダによって生成された画像に含まれるジャギー (図形の縁がピクセルによってギザギザになっている箇所) を除去する作業である。ジャギーは連続的なアナログ値をピクセルという離散的な値にマッピングする際に生じる虚象であり、ピクセルデータの補間をすることで除去できる [5]。

具体的には、任意のピクセル $K_{i,j}$ とそれを囲む八つのピクセルに対して、予め定めたウエイト ω を掛け、近傍の情報も反映するようにピクセルの値を下記のように加工する。

$$K_{i,j(new)} = \frac{1}{10}\begin{pmatrix} \omega K_{i-1,j+1} + \omega K_{i,j+1} + \omega K_{i+1,j+1} \\ +\omega K_{i-1,j} + \omega 2 K_{i,j} + \omega K_{i+1,j} \\ +\omega K_{i-1,j-1} + \omega K_{i,j-1} + \omega K_{i+1,j-1} \end{pmatrix}$$

4　DRP 上での設計と実装

41　仕　　様

エッジ近傍合成機能付き α ブレンダの設計制約は下記の通りである。

- 係数 α の値は可変である
- 2 系統の 24bit (RGB) 画像データをデータストリームとして、DRP 外部からシリアルに入力する
- 3 × 3 の画素を網羅するスライディング・ウィンドウを実現し、それによるデータのフィードバックを新たに設計するアドレスジェネレータで管理する

41.1　スライディング・ウィンドウの実現

任意のピクセル $K_{i,j}$ についてエッジ近傍処理を行う場合、それを取り囲む 8 ピクセルの画素データが必要となる。それらのピクセルのデータを効率的に扱うために、図 5 で示すスライディング・ウィンドウを実現する必要がある。スライディング・ウィンドウを実現するにあたって、次の制約を考慮しなくては

ならない。

- メモリは一次元配列として存在し、一度に 1 書き込み、または 2 読み出ししかできないため (2 port memory)、それを考慮した設計にする
- 大量に生成される処理データを効率的にメモリに格納し、後の段階で発生するデータのフィードバックを効率的に管理する

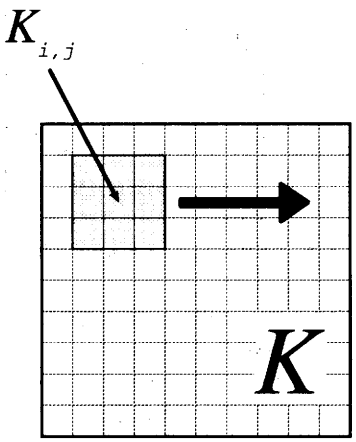

図 5　スライディング・ウィンドウ

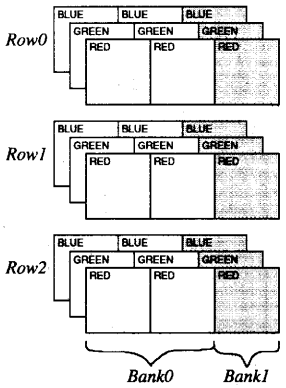

図 6　スライディング・ウィンドウを実現するためのメモリ展開

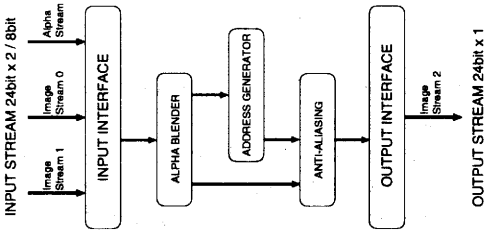

図 7　エッジ近傍合成機能付き α ブレンダの基本構成

4.1.2　アドレスジェネレータ

α ブレンディングで生成されるデータの退避先アドレスを生成し、スライディング・ウィンドウを実現するのに必要な 9 ピクセル分のデータのメモリ番地を出力する機構として、アドレスジェネレータを設計する。処理内部全体に点在する複数のカウンタと連携することで機能する。

4.1.3　メ　モ　リ

スライディング・ウィンドウを実現するにあたって DRP 上の VMEM (図 2 参照) を用いる。DRP に装備されている VMEM は 2 ポートメモリで、一度に 1 書き込み、ないしは 2 読み出しができる。α ブレンダ処理部からデータが出力され次第、アドレスジェネレータに従って VMEM へデータを退避する。エッジ近傍合成処理では、9 ピクセル分の画素データを同時に用意することが必要条件となるが、図 6 で示すように *Bank*0, *Bank*1 に二つに分けて、2 クロック (2 コンテキスト) で読み出す。

4.2　エッジ近傍合成機能付き α ブレンダの演算機構の基本構成

実装したアプリケーションの基本構成を図 7 に示す。大きく分けて α ブレンダの演算処理部、アドレスジェネレータ、エッジ近傍合成の演算処理部から構成される。処理の流れは次の通りである。

入力されたストリームデータは 8bit 単位に分けられ、α ブレンダ処理部に入力される。α ブレンダ処理部が出力したデータはアドレスジェネレータによりメモリに一時的に退避される。エッジ近傍合成の処理部が必要なデータを必要に応じてアドレスジェネレータ経由でメモリから読み出し、それを元にエッジ近傍合成処理を行う。

4.3　各コンテキストの概要

DRP ではコンテキストという単位でハードウェアを切替えることで複数の機能を同じチップ上で実現している。エッジ近傍合成機能付き α ブレンダを DRP に実装するにあたって、それぞれのコンテキストの処理を図 8 に示した。

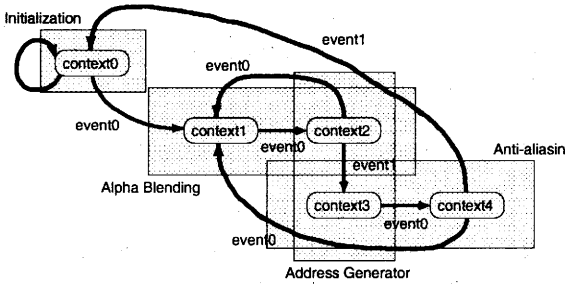

図 8　コンテキスト分割図

各コンテキストの概要は以下の通りである。

Context0: Idle ステージ。レジスタやメモリの初期化を行う。アプリケーションの実行時の初めに 1 回のみ実行される。

Context1: α ブレンディング機構 (前処理)。RGB データの内、赤 (R) と 緑 (G) のデータ (2 系統) を取り込み、α ブレンディングを行う。処理結果と共に Context2 に遷移する。

Context2: α ブレンディング機構 (後処理)。RGB データの内、青 (B) のデータ (2 系統) を取り込み、α ブレンディングを

表 1 使用リソース

Context	Alu	Dmu	Rfu	Pe	Mul	Vmem	Hmem
0	2	2	31	31	6	0	0
1	9	7	10	10	3	0	0
2	24	7	19	24	0	9	0
3	34	21	11	34	0	9	0
4	16	17	8	17	3	9	0
MAX	**34**	**21**	**31**	**34**	**6**	**9**	**0**

表 2 α ブレンダの動作実行時間 (256 × 256 ピクセル)

Processor	Clock	Clocks (recorded)	Exec time[ns]	Normalized against DRP
DRP	38 MHz	262,144	6,898	1.0
DSP (TI C6713)	225MHz	27,350,179	121,556	17.6
Pentium4	2.53GHz	52,615,440	20,796	3.0
AthlonXP	1.47GHz	31,869,691	21,680	3.1

行う。Context1 からの処理結果とこのコンテキストの処理結果をまとめ、アドレスジェネレータに従ってメモリ (VMEM) に書き込む。スライディング・ウィンドウを実現するだけのデータが揃ったならば、次のコンテキストに遷移する。それ以外の場合は、Context1 に遷移する。

Context3: スライディング・ウィンドウ処理部。アドレスジェネレータによりアドレスを割り出し、必要なデータをメモリ (VMEM) から読み出す。読み出したデータと共に Context4 に遷移する。

Context4: エッジ近傍合成処理部。Context3 から受け取ったデータと自らメモリより読み出したデータを合わせて、エッジ近傍合成処理を行う。ストリームデータがまだ外部より入力されているのであれば、Context1 へ遷移する。それ以外は、Context0 へ遷移し、システムがリセットされる。

この設計では立上り時間として 2(2m + 3) クロック掛かり、その後 4 クロックに一回結果が出力される。

5 評価・考察

前節で示した エッジ近傍合成機能付き α ブレンダについて、演算器レベルの RTL で記述し、DRP 設計環境の自動配置配線ツール RAP を用いて合成した。以下にその評価結果を示す。

5.1 使用リソース

表 1 は、各コンテキスト毎に必要な演算リソースの量をまとめたものである。自動配置配線ツールで回路を展開した後、クリティカルパス (26,337ps or 26ns) を求めた。それより、DRP が正常に動作する周波数を計算した。結果は表 2 の通りである。

処理全体に使用するリソースは全 PE 数 (512) の 7% にあたり、1 Tile 内に抑えることができた。面積効率の面では効率良く使えているといえ、その分消費電力の観点から見ても低く抑えることができている。しかしながら、乗算器を多用する エッジ近傍合成機能付き α ブレンダにおいてさらなる並列処理を実現するためには、乗算器の数を現在の 8 よりも増やす必要がある。

5.2 他のアーキテクチャとの比較

C 言語で記述したソフトウェアを CPU、DSP で実行し、DRP の結果と比較した。評価に用いたハードウェアは以下の通りである。

CPU: Pentium 4 (2.53 GHz), Athlon XP (1.47 GHz)

- OS: RedHat Linux 2.4.21
- コンパイラ: gcc 2.95 (-O3 最適化オプション使用)

DSP: Texas Instruments C6713 (評価ボード: TMS320C6713)

- OS: Windows 2000
- 開発環境: TI Code Composer Studio Ver. 2.20.05
- コンパイラ: 附属コンパイラ (no -ms, -o3, -pm -op0 最適化オプション使用)

Pentium 4 (2.53 GHz)、Athlon XP (1.47 GHz) での実装には、gcc コンパイラを用い、コンパイラが提供する -O3 オプションを用いた。DSP の参考機種として、Texas Instruments 社の C6713 を用いた。C6713 は VLIW アーキテクチャの浮動小数点 DSP であり、225 MHz の動作周波数で 1350 MFLOPS を実現する。コンパイルは附属のコンパイラを用いて、コンパイラが提供するすべての最適化オプションを有効にして行った。

5.3 実行速度

表 2 はそれぞれのアーキテクチャでの実行結果である。最新の高速な CPU と DRP でそれぞれ エッジ近傍合成機能付き α ブレンダを実行した場合、DRP の方が Pentium 4、Athlon XP に対して 3 倍、DSP に対しては 17 倍高速であることが図 2 より分かる。これは エッジ近傍合成機能付き α ブレンダが潜在的に並列度が高いことと、DRP の並列処理能力がうまく噛みあって生まれる相乗効果によるものである。

Pentium 4、Athlon XP、DSP に対しては、コンパイラが提供する最適化技術を可能な限り用いたものの、逐次処理をしている点でエッジ近傍合成機能付き α ブレンダの並列性を活かしきれていない。特に同時に複数の演算処理を行い、2 クロックでエッジ近傍処理用のデータを全てメモリから用意できる DRP は、メモリアクセスが頻繁に発生するアプリケーション

において、CPU/DSP を遥かに上回るパフォーマンスを発揮することができた。

5.4 更なる高速化に向けて

本稿では、ストリーム処理を念頭に置いたため、DRP 外部より画像データを適宜入力する設計にした。これ以外の実装方法として、データをその都度入力するのではなく、外部より直接 DRP 内部のメモリに書き込む方法が考えられる。仮に外部より直接 DRP 内部のメモリに書き込むことができたとすれば、α ブレンダ処理機構として 2 コンテキスト分を 1 コンテキストに削減し、更なる並列性を抽出することができる。この I/O を高速化する手法を用いれば、3 クロックに一回処理結果を出力できる エッジ近傍合成機能付き α ブレンダが実装できることになる。

6 結　　論

本稿では、NEC が開発した DRP 上にエッジ近傍合成機能付き α ブレンダを実装し、その評価を行った。その結果、当該実装例のように並列性が高く、大量な計算量を処理する必要があるアプリケーションにおいて、現行の PC や DSP を遥かに上回るパフォーマンスを得ることができた。プロセッサをアレイ状に配置し、メモリが拡散して配置されている DRP は並列性が高いアプリケーションにおいて、その有効性が確認された。

謝　　辞

DRP およびその開発環境を提供して頂き、本研究に対する多くのアドバイスをしてくださった NEC エレクトロニクスおよび NEC ラボラトリーズのみなさまに深く感謝致します。また、Nokia の後藤様にも感謝致します。

文　　献

[1] 岡崎 彰夫. **はじめての画像処理技術**. 工業調査会, 2000.

[2] 佐藤 友美. "アイピーフレックスのリコンフィギャラブル・プロセッサ・デバイス". 第 1 回 リコンフィギャラブルシステム研究会予稿集, 2003 年 9 月.

[3] 山田、出口、金子、天野. "マルチコンテキストリコンフィギャラブルデバイス上でのデータドリブンアプリケーションの実装". FPGA/PLD Conf. Exhibit. ユーザプレゼンテーション予稿, 2003 年 1 月.

[4] 出口、山田、天野. "DRP でのウェーブレットフィルタの実装". 信学技報, VLD2002-142, CPSY2002-95, 2003 年 1 月.

[5] Foley and et al. *Computer Graphics: Principles and Practice, Second Edition in C.* Addison-Wesley, Boston, 1997.

[6] Fujii, T., Furuta, K., Motomura, M., Nomura, M., Mizuno, M., Anjo, K., Wakabayashi, K., Hirota, Y., Nakazawa, Y., Ito, H. and Yamashina, M. A Dynamically Reconfigurable Logic Engine with a Multi-Context Multi-Mode Unified-Cell Architecture. In Proc. of Intl. Solid-State Circuits Conf., pages 360–361.

[7] M. Motomura. A Dynamically Reconfigurable Processor Architecture. Microprocessor Forum, Oct. 2002.

[8] QuickSilver Technology. *http://www.quicksilvertech.com/.*

[9] V. Baumgarte, F. May, A. Nückel, M. Vorbach, and M. Weinhardt. PACT XPP - A Self-Reconfigurable Data Processing Architecture. In Proc. of International Conference on Engineering of Reconfigurable Systems and Algorithms(ERSA'01), 2001.

[10] X.-P. Ling, H. Amano. WASMII: A Data Driven Computer on a Virtual Hardware. In Proc. of the IEEE Symposium on FPGAs for Custom Computing Machines(FCCM'93), pages 33–42.