

1 チップ再構成可能コンピューティングシステムの開発

安河内 真弓 下尾 浩正 山脇 彰 岩根 雅彦
九州工業大学 工学部 電気工学科

概要: 様々な応用処理に対して適切な回路構成をとることのできる再構成可能プロセッサ MRP を提案する。MRP は、1 回の回路構成で複数の機能処理回路を構成できる 1 プレーンマルチファンクション構成により実行回路の書き換え回数を削減し、大規模な回路に対しては 1 ファンクションマルチプレーン構成を採ることで対応している。MRP は、CPU コア、再構成部制御回路および複数の再構成部実行回路から成り、再構成部実行回路を動的に変更することでローディング時間を隠蔽することができる。予備実験として、1 プレーンマルチファンクションを実現する個別部品構成の試作機 486RCP を開発し、評価を行ったところ、フィボナッチ数列の第 n 項算出では最大 30 倍の性能向上が得られた。

キーワード: 再構成可能プロセッサ、1 プレーンマルチファンクション、1 ファンクションマルチプレーン、再構成部、動的ローディング、フィボナッチ数列

1 Chip Reconfigurable Computing System

Mayumi Yasukouchi, Kousei Shimoo, Akira Yamawaki and Masahiko Iwane
Department of Electrical Engineering, Faculty of Engineering,
Kyusyu Institute of Technology

Abstract: This paper presents the Micro Reconfigurable Processor (MRP) can be suitable circuit composition to the various application processing. MRP reduces the number of times of reconfiguration using Single Plane Multiple Function organization, which can constitute two or more functional processing circuits per circuit configuration. Single Function Multiple Plane organization is enable large circuits to be run on limited physical FPGA. MRP contains CPU core, control logic of reconfigurable part and some reconfigurable part, and the dynamic loading circuit reduces overhead for reconfiguration. We report the experiment model 486RCP, discrete parts composition, which developed for the purpose of realization of Single Plane Multiple Function organization. The experiment results show that 486RCP achieves speedup maximum 30 times on Calculation of Fibonacci numbers.

Key Word: Reconfigurable Processor, Single Plane Multiple Function, Single Function Multiple Plane, Reconfigurable Part, Dynamic Loading, Fibonacci numbers

1. はじめに

処理のハードウェア化では、PCI インターフェースに接続された FPGA (Field Programmable Gate Array) を用いて、ハードウェアアルゴリズムを開発している研究が多い。また、Brigham Young 大学の DISC [1] のように、単独の処理装置として、再構成部分に特徴を持たせて、通常の命令から少し

複雑な処理 (FFT) 程度の小さな処理単位を対象にして、処理に応じて動的に命令を書き換える研究もある。UCB 校の Garp [2] に代表される研究は、特徴を持った再構成可能なハードウェアをコプロセッサとして開発し、その構成について論じている。UCI の MorphoSys [3] では、データ並列処理を対象として、特徴ある再構成可能な処理ユニットとそれ

を制御する回路をコプロセッサとして提案し、CPU コアを含んでチップ化している。また、Xtensa [4] のように特定分野に適するようにチップ内構成を容易に変更できるマイクロプロセッサの設計方法の研究もある。

本研究では、再構成部制御回路に特徴を持たせて、多種多様な処理を対象とした再構成可能プロセッサを提案する。FPGA上にCPUコアと様々な応用処理に対して適切な回路構成を実現する再構成部を持った再構成可能プロセッサMRP (Micro Reconfigurable Processor)を開発する。また、マザーボード上に組み込んだMRPシステムを製作し、さまざまな応用処理に対してMRPシステムの有効性を確認することを目標とする。

本論文では、まずMRPシステムの概要を述べ、システムの核となるMRPの構成について説明した後、基本的な機能を検証するために製作した試作機である486RCPについて述べ、試作機での実験結果を示し、最後に結びとする。

2. 再構成可能コンピューティング

2.1 概要

本システムでは、対象とする処理に適した回路が再構成部に構築され、その回路データはユーザがCADによってあらかじめライブラリ化しておく。このようなシステムの概要を図1に示す。

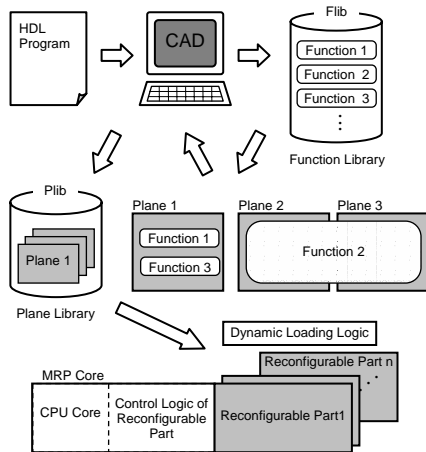


図1 システム概要

一般的な処理において、共通化またはサブルーチン化できる部分をファンクションとしてHDL (Hardware Description Language) で記述し、デバッグしたものをHDLレベルでファンクションライブラリ (Flib) に登録する。これらを再構成部にローディングする固定の大きさ (プレーン) にまとめて、論理合成、デバッグしたものをプレーンライブラリ (Plib) に登録する。再構成回数を減らすために、1回の再構成で複数のファンクションを構築する。したがって、1つのプレーンには再構成部の

容量が許す限り関連する複数のファンクションを格納する (1プレーンマルチファンクション構成)。

また、FPGA容量の制限により、1つのファンクションを一度に再構成部にローディングできない場合もある。このような場合には、Flibを利用してHDL記述を行い、論理合成時にプレーンの大きさに分割し、デバッグしたものをPlibに登録し、同時にFlibにも登録する (1ファンクションマルチプレーン構成)。1ファンクションマルチプレーン構成では、複数のプレーンを切り替えて連続的に処理を実行する必要があるため、ファンクションの実行と再構成を並列に行い (半動的ローディング)、ローディング時間の隠蔽を行う。

2.1 オンチップ再構成可能システム

このようなファンクションを実行する再構成可能プロセッサ (MRP: Micro Reconfigurable Processor) システムを図2に示す。

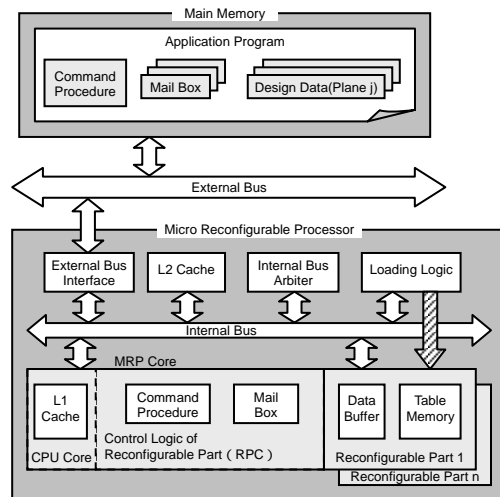


図2 オンチップ再構成可能プロセッサシステム

MRPは、CPUコア、CPUコアからの命令により再構成部の構築、ファンクション実行といった指示および制御を行う再構成部制御回路 (RPC)、 n 個のプレーンを持つ再構成部 (ファンクション実行回路) から成るMRPコアと、回路の構築を行うローディング回路などで構成される。

CPUコアは、RPCに再構成部へのプレーンのローディング、再構成部のファンクション実行を指示する命令を持ち、その命令によってRPCを駆動する。CPUコアとRPCの通信は、メモリを介して行う。RPCへの命令には、プレーンのローディング、ファンクションの実行制御といった命令をコマンドとし、それらをひとまとめでしたコマンドプロシージャ (CP: Command Procedure) を用いる。ユーザはアプリケーションに応じて、プレーンのローディング、ファンクションの実行、条件分岐などを

スケジューリングし、CPとしてメモリに格納する。

また、再構成部がファンクションを実行する際に必要となるデータもファンクション毎に Mail Box としてひとまとめにし、あらかじめメモリに格納しておく。Mail Box には、ファンクション固有のファンクション番号と各ファンクションに必要な情報（パラメータ）を格納する。パラメータには、即値、ポインタ等が含まれる。このように、Mail Box に実行するファンクションの情報を持たせることによって、アプリケーションに柔軟な構成ができる。

RPC は CPU コアから CP を受け取りそのコマンド列を実行する。RPC がローディングコマンドを実行すると、RPC によって駆動されたローディング回路が、置き換え可能な再構成部にメモリからプレーンを動的にローディングする。また、ファンクション実行コマンドを実行すると、RPC によって駆動された再構成部が Mail Box を利用してファンクションの実行を行う。

3 MRP 構成

3.1 制御部と再構成部

MRP コアとローディング回路の構成を図 3 に示す。

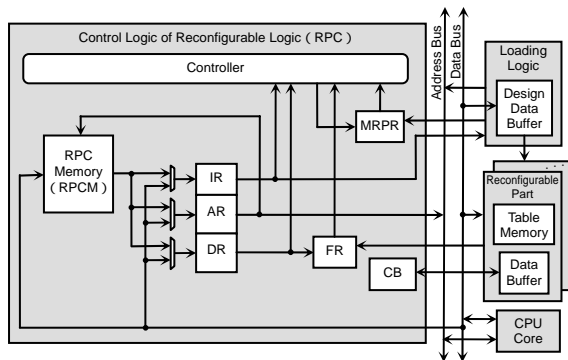


図 3 MRP コアとローディング回路

RPC は、CP や Mail Box を格納する RPC メモリ (RPCM)、コマンドコードや MRP 命令のオペコードを保持する命令レジスタ (IR)、CP や Mail Box の格納先メモリアドレスや RPCM 内アドレスを保持するアドレスレジスタ (AR)、ファンクション実行や Mail Box 実行に必要なデータを保持するデータレジスタ、CP の条件分岐のために各ファンクションがセット/リセットを行う条件フラグを保持するフラグレジスタ (FR)、再構成部の各プレーンの状態 (ローディング中、ファンクション実行中) を示すステータスレジスタ (MRPR)、各プレーン間でデータなどの通信を行うためのメモリであるコミュニケーションボックス (CB)、および、コントローラで構成されている。

3.2 再構成コマンド

CPU コアは、RPC に対して CP のローディングおよびその実行を専用命令によって指示する。それらの命令フォーマットを表 1 に示す。

表 1 MRP 命令

命令	オペランド
LDCMDP	Command Procedure Head Address, WF
EXECMD	Command Start Address, No. of Command, WF
EDTCMD	Command format, WF

WF=1 : Wait for Completion of Execution
WF=0 : Begin Next Instruction immediately

CP は、ヘッダとコマンド列から成り、ヘッダには CP のサイズなどを格納する。コマンド列は、表 2 に示すコマンドから構成される。コマンドは、コード部、アドレス部、データ部から成り、RPC によってデコード、発行される。

表 2 コマンドフォーマット

Command	Format		
	Code	Address	Data
LDP	000	Design Data Head Address	RP No.
EXE	001	Mail Box Head Address	RP No.
BRA	010	Branch Target Address	Flag No.
BRC	011	Branch Target Address	
END	100		

- (1) LDCMDP (LoaD CoManD Procedure) 命令：RPC に CP のローディングを指示する。RPC は、第 1 オペランドで指定された CP の格納先から CP を RPCM に格納する。
- (2) EXECMD (EXEcute CoMmanD) 命令：RPCM に格納されたコマンドの実行を RPC に指示する。RPC は、第 1 オペランドで指定された RPCM の内部アドレスからコマンドの実行を開始し、第 2 オペランドで指定されたコマンドの数だけ実行する。
- (3) EDTCMD (Execute DirecT CoMmanD) 命令：CPU コアが直接コマンドを実行する。EDTCMD 命令のオペコードがコマンドコードと対応付けられており、RPC は命令フェッチによって実行するコマンドを解釈する。第 1 オペランドでは表 2 に示すアドレス部、データ部を指定する。
 - (i) LDP (LoaD Plane) コマンド：RPC がローディング回路に対して再構成部の回路構築を指示する。ローディング回路は、アドレス部で指定された設計データ格納先から設計データを読み出し、データ部で指定された再構成部にプレーンをローディングする。
 - (ii) EXE (EXEcute function) コマンド：RPC が再構成部にファンクション実行を指示す

る。RPC は、アドレス部で指定された Mail Box の格納先から Mail Box を読み出し、その内容に基づいてデータ部で指定された再構成部の制御を行う。再構成部（プレーン）は RPC の制御に基づき、ファンクションの実行を行う。Mail Box は、ファンクション固有であるので、1 ファンクションマルチプレーンの場合、最初のプレーンの EXE コマンドのみ Mail Box の先頭アドレスを指定する。

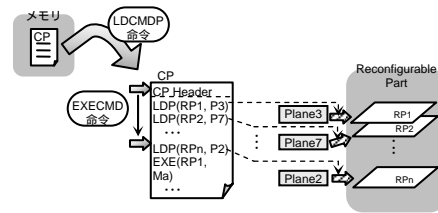
- (iii) BRA (BRAnch) コマンド：CP 内でのコマンドの条件分岐を示す。データ部で指定された FR のビットが 1 である場合、アドレス部で指定された CPM 内アドレスに分岐する。0 である場合、分岐せず次のコマンドが発行される。
- (iv) BRC (BRAnch on Condition) コマンド：CP 内でのコマンドの無条件分岐を示し、アドレス部で指定された CPM 内アドレスに無条件に分岐する。
- (v) END (command END) コマンド：CP の終わりを示す。END コマンドによって、MRP は処理を終了する。

WF (Wait Flag) は、CPU コアが RPC の命令実行終了を待つか待たないかを定めるフラグである。1 にした場合 CPU コアは命令完了を待ち、0 にした場合 CPU コアは命令の完了を待たずに独立して自身の処理を行う。

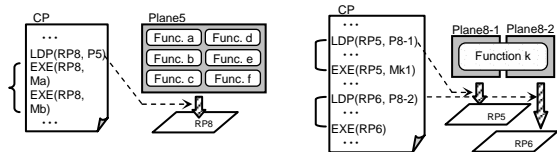
3.3 コマンド使用例

MRP において、プレーンのローディング、ファンクションの実行はすべてコマンドを通して指示される。コマンドの使用例を図 4 に示す。

MRP を用いてアプリケーションを実行するには、まず CPU コアが RPC にメモリからの CP の読み出しを指示する LDCMDP 命令を実行する(図 4 (a))。最初、再構成部のどのプレーンにも回路がローディングされていないので、CP のコマンド列の初めにはプレーンの数だけ LDP コマンドを記述する。したがって、それらの LDP コマンドを EXECMD 命令で実行する(図 4 (a))。このとき WF を 0 にしておくことで、CPU コアはすべてのプレーンのローディングの完了を待つことなく何らかの処理を行うことができ、最初のローディング時間を隠蔽することができる。すべてのプレーンのローディングが終わると、CPU コアは EXECMD 命令を用いて残りのすべてのコマンドを実行したり、EDTCMD 命令を用いて個別にコマンドを実行するなど、CPU コアと再構成部が独立あるいは協調して処理を行っていくことが可能である。



(a) 最初のコマンド実行



(b) 1プレーンマルチファンクション

(c) 1ファンクションマルチプレーン

図 4 コマンド使用例

最初のローディング以外は、プレーンの再構成とファンクション実行が連続的に行われていくが、プレーンのローディングには時間がかかるので、再構成時間をうまく隠蔽していかなければ、ボトルネックになってしまう。1 プレーンマルチファンクション構成では、図 4 (b) のように LDP コマンドの実行後複数の EXE コマンドを実行するため、この間に別のプレーンをローディングすることで再構成時間を隠蔽できる。また、1 ファンクションマルチプレーン構成では、図 4 (c) のように LDP コマンドと EXE コマンドが 1 対 1 の関係にあるので、再構成時間を隠蔽するためには、LDP コマンドを EXE コマンドのかなり前に記述する必要がある。

再構成時間を隠蔽する方法として、ファンクションの実行とプレーンの再構成を並列に行う動的ローディングの実現を考える。そのためには、コマンドを逐次に発行するのではなく、LDP コマンドと EXE コマンドを並列に発行する必要がある。したがって、LDP コマンドと EXE コマンドが連続して存在し、かつ、異なる再構成部である場合、次のコマンドも発行する。

4. 486RCP の開発

4.1 486RCP 構成

上記のようなシステムの開発に際して、回路構成回数の削減を目的とした 1 プレーンマルチファンクションの実現、および、多種多様なアプリケーションに柔軟に対応するために再構成部制御回路と再構成部のインターフェースの確立（物理的分離）を目的として、486DX2 マイクロプロセッサ、PVI486SP3 マザーボード、XILINX の XC4020E FPGA を 2 個使用した個別部品構成の試作機 486RCP を開発した。

図 5 に 486RCP のシステム構成を示す。

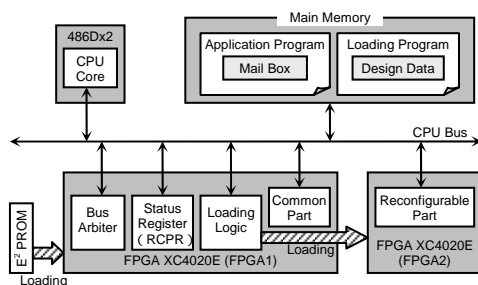


図5 486RCP システム構成

このシステムは、再構成可能プロセッサ 486RCP およびメインメモリ等から成っている。486RCP は、主に CPU コア、ローディング回路、再構成部を制御する共通部、バス調停回路、486RCP 内の状態を示すステータスレジスタ (RCPR)、および、ファンクションを実行する再構成部で構成される。

共通部には、多種多様なファンクションにおいて統一的に扱える機能 (バスインターフェースなど) を集約し、再構成部にはファンクションに固有な機能のみを実現する。共通部は、各ファンクションに共通な機構なので固定回路として実現し、再構成部はファンクションに個別の機能なので、可変回路として実現する。そのため、共通部と再構成部のインターフェースを明確化している。こうすることによって、ファンクションの作成が統一かつ容易になり、多様な処理に柔軟な構成をとることができる。

メモリにはローディング・プログラム用の設計データ (プレーン)、ならびに、アプリケーション・プログラム用の Mail Box が格納されている。

ローディング回路は、CPU からのローディング実行命令を受けると、メモリに格納されている設計データを再構成部へローディングする。共通部は、CPU からのファンクション実行命令を受けると Mail Box の内容を読み出し、再構成部の制御を行う。それに伴って、再構成部は各ファンクションの実行を行う。

RCPR は図 6 のような構成をしており、再構成部のイネーブル/ディスエーブル、ローディング回路が実行中か否か、再構成部が実行中か否かを示す。

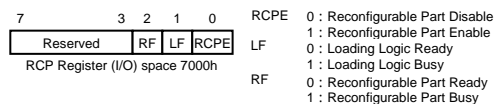


図6 RCPR の構成

実装形態は、ローディング回路、RCPR、バス調停回路、共通部を 1 つの FPGA (FPGA1) に、再構成部をもう 1 つの FPGA (FPGA2) に構成する。電源投入時、FPGA1 には E²PROM の使用により回路が構築され、FPGA2 は CPU からの命令によりローディング回路が回路構築を行う。また、

486DX2、FPGA をドータボードとして構成し、市販の PC/AT 互換機用マザーボードのプロセッサソケットに接続する。動作クロックは 16MHz である。マザーボードに接続したドータボードの写真を図 7 に示す。



図7 486RCP の写真

4.2 再構成用命令と動作

486RCP では、CP のコマンドの各機能を確認し、CP 機構への拡張性を検証することが目的であるため、MRP 命令のうち、コマンドを直接実行する EDTCMD 命令の実装を考える。したがって、EDTCMD (LDP, EXE コマンド) に対してそれぞれ LDRCP, EXERCMP 命令という 486RCP 専用の命令によって、コマンドの各機能を確認する。また、486RCP は、市販のプロセッサ (486DX2) を使用しており、共通部、ローディング回路、ステータスレジスタ (RCPR) を IO として実装しているため、LDRCP/EXERCMP 命令の代わりに in/out 命令を使用する。表 3 に CPU の命令フォーマットを示す。

表3 CPU 命令フォーマット

命令	Op	I/O Address	Data
WRCPR	OUT	7000h	Value of Status Register
RRCPR	IN	7000h	Value of Status Register
LDRCP	OUT	8004h	Design Data Head Address
EXERCMP	OUT	8008h	Mail Box Head Address

WRCPR/RRCPR 命令は、RCPR への書き込み/読み出し命令である。LDRCP (LoaD RCP) 命令は、再構成部への回路のローディングを指示する。オペランドで設計データの先頭アドレスを指定する。EXERCMP (EXEcute RCP) 命令は、再構成部へのファンクションの実行を指示する。オペランドで Mail Box の先頭アドレスを指定する。

5. 実験

5.1 実験用プログラム

検証にあたり、128 ビットまでの加減算、フィボナッチ数列第 n 項の算出の 3 つのファンクションを 1 つのプレーンとしてまとめた。そのときの Mail Box の仕様を図 8 に示す。

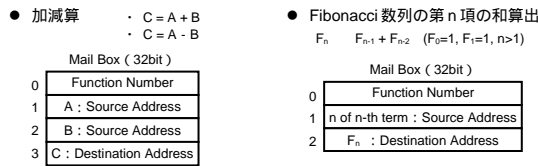


図 8 Mail Box の仕様

Mail Box は各演算とも、ファンクション番号、ソースデータの格納先アドレス、デスティネーションデータの格納先アドレスの順で構成している。

システムの検証に用いたローディング・プログラムおよびアプリケーション・プログラム(フィボナッチ数列の第 n 項算出)を図 9 に示す。

```

Loading Program
handle = _open("suite1.bit", O_RDONLY);
design_data = malloc(design_data_size);
_read = (handle, design_data, design_data_size);

LDRCP(design_data);

while(RRCPR & 0x0002 != 0);

Application Program
n = 186;
F[4]; /* 128bit = 32bit X 4 */

mail_box[0] = function_number_fibo128;
mail_box[1] = &n;
mail_box[2] = &Fn;

EXERCPR(design_data);

while(RRCPR & 0x0004 != 0);

for(i=3; i>=0, i--) printf("%x", F[i]);

```

図 9 検証プログラム例

ローディング・プログラムでは、まず作成した設計データ(加減算、フィボナッチ数列)をメモリに格納し、LDRCP 命令を発行する。ローディングの終了は RRCPR 命令の発行によって確認する。フィボナッチ数列第 n 項算出のプログラムでは、まず n、F[n] を定義し、Mail Box としてファンクション番号、n、F[n] の格納先アドレスを用意する。その後、EXERCPR 命令を実行し、アプリケーションの終了を RRCPR 命令で確認する。

5.3 結果と考察

FPGA のインプリメント結果を表 4 に示す。

表 4 FPGA のリソース状況

FPGA	CLB 使用率	ゲート規模
FPGA1	76%	約 15,200 ゲート
FPGA2	49%	約 9,800 ゲート

Xilinx 社 XC4020E : 約 20,000 ゲート

単一プロセッサとの評価を図 10 に示す。

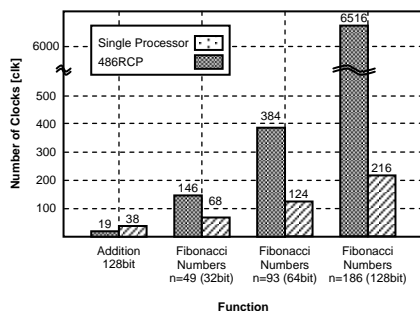


図 10 単一プロセッサとの評価

図 10 より、フィボナッチ数列の第 n 項の算出では、単一プロセッサに比べて約 2~30 倍の性能向上が見られる。しかし、128 ビットの加算では、単一プロセッサのほうが速くなっている。これは、Mail Box 読み出しのオーバーヘッドによるものである。これらの結果から、128 ビットの加減算のように単一プロセッサでも計算にあまり時間がかからないファンクションよりも、フィボナッチ数列のようにファンクションの処理がある程度大きく、単一のプロセッサでは計算時間がかかるようなファンクションに対しては本システムが有効であることが分かった。

6. むすび

本論文では、CPU コアと再構成部から成る 1 チップ再構成可能プロセッサ MRP と実行環境に適したプロセッサ構成のためのシステムを提案した。予備実験として、1 ロードマルチファンクションを実現する個別部品構成の試作機 486RCP を製作し、128 ビットの加減算、フィボナッチにおいて評価を行ったところ、フィボナッチ数列の第 n 項算出では最大 30 倍の性能向上が得られ、本システムの有効性を検証することができた。

今後は、様々なアプリケーションに対して適用、評価を行い、その結果をふまえて、1 ファンクションマルチプレーンへ対応した MRP システムを開発していく予定である。

参考文献

- [1] M.J. Wittig and B.L. Hutchings, "A Dynamic Instruction Set Computer," *Proc. IEEE Symp. FPGAs for Custom Computing Machines*, pp. 99-107, Apr. 1995.
- [2] J.R. Hauser and J.Wawrzynek, "Garp: A MIPS Processor with a Reconfigurable Co-Processor," *Proc. IEEE Symp. Field-programmable Custom Computing Machines*, Apr. 1997.
- [3] H. Singh, et al., "MorphoSys: An Integrated Reconfigurable Architecture", *Proceeding of the NATO Symposium on Concepts and Integration*, April, 1998.
- [4] Ricardo E. Gonzalez, "XTENSA: A Configurable and Extensible Processor," *IEEE Micro*, March-April 2000.
- [5] 岩根, "再構成可能ハイパーコンピューティングの開発", 設計メモ(内部資料), 1998年2月.
- [6] 下尾, 安河内, 山脇, 岩根, "オンチップ再構成可能プロセッサシステムの開発", 第 5 回システム LSI ワークショップポスタセッション, 2001年11月.