

## ハード/ソフト・コラーニングシステムにおける 各種マイクロプロセッサの設計と実装

中村 浩一郎<sup>†</sup> Hoang Anh Tuan<sup>†</sup> 小柳 滋<sup>†</sup> 山崎 勝弘<sup>†</sup>

<sup>†</sup> 立命館大学大学院理工学研究科 〒525-8577 滋賀県草津市野路東 1-1-1

E-mail: <sup>†</sup> koichiro@cpsy.is.ritsumeimei.ac.jp, oyanagi@cs.ritsumeimei.ac.jp, yamazaki@se.ritsumeimei.ac.jp

あらまし 我々はプロセッサアーキテクチャの理解を基に、アセンブリプログラミング、プロセッサ設計、及びFPGA ボード上での検証など、ハード・ソフト両面を学習するハード/ソフト・コラーニングシステムを構築している。本稿では、本システムにおけるハード/ソフト協調学習方法を紹介し、コラーニングシステムにおける基本命令セットに基づいて設計したシングルサイクル、マルチサイクル、パイプライン、及びスーパースカラ各方式の教育用マイクロプロセッサのFPGA 上での実装と性能評価について述べる。また、拡張命令セットに基づくシングルサイクルプロセッサを設計し、本システム上に実装した。これらの経験を踏まえ、本システム上でプロセッサを開発するために必要となる開発支援環境について検討する。

キーワード ハード/ソフト・コラーニングシステム、マイクロプロセッサ、プロセッサアーキテクチャ

## Design and Development of Microprocessors on a Hardware/Software Co-learning System

Koichiro NAKAMURA<sup>†</sup> Hoang Anh TUAN<sup>†</sup> Shigeru OYANAGI<sup>†</sup> and Katsuhiro YAMAZAKI<sup>†</sup>

<sup>†</sup> Graduate School of Science and Engineering, Ritsumeikan University

1-1-1 Nojihigashi, Kusatsu-shi, Shiga, 525-8577 Japan

E-mail: <sup>†</sup> koichiro@cpsy.is.ritsumeimei.ac.jp, oyanagi@cs.ritsumeimei.ac.jp, yamazaki@se.ritsumeimei.ac.jp

**Abstract** The hardware/software co-learning system helps user to learn both hardware and software such as assembly programming, processor design and verification using FPGA boards by understanding processor architecture. This paper describes how to learn both hardware and software on this system. We have developed single cycle, multi cycle, pipeline and superscalar microprocessors based on a basic instruction set for this system. In addition, a single cycle microprocessor for another instruction set is described. At last we discuss about a support system for developing a processor which aims to efficiently debug both hardware and software.

**Keyword** Hardware/Software Co-learning System, Microprocessor, Processor Architecture

### 1. はじめに

ハードウェアとソフトウェアの協調設計、組み込みプロセッサの普及などにより、プロセッサアーキテクチャの理解、及びプロセッサ設計能力が重要視されており、我々はその教育システムの構築が必要であると考え。他大学での研究においても教育用プロセッサを用いたシステム[6]~[9]、プロセッサシミュレータ[10][11]、ハード/ソフト協調学習システム[12]など、さまざまなアプローチでプロセッサアーキテクチャ教育が試みられている。

我々はプロセッサアーキテクチャの理解を基に、ハード・ソフト両面の学習を体系的に行えるハード/ソフト・コラーニングシステムを考案し、システムを構築してきた[1]~[4]。ハード/ソフト・コラーニングシ

ステムは、システム利用者（学習者）に各種プロセッサアーキテクチャを理解してもらい、アーキテクチャにあったプログラム開発を行うソフトウェア学習と、ソフトウェア学習で得たプロセッサアーキテクチャの知識を基に、実際に HDL を用いてプロセッサを設計し、FPGA 上に実装するハードウェア学習から成り立っている。このように、ソフトウェアとハードウェアの境界であるプロセッサを意識した学習方法をとることで、ハードウェアとソフトウェアの協調学習を行うことを目的としている。システムを構築するにあたり、教育的に理解しやすい命令セットと各種プロセッサアーキテクチャを考案した。また、利用者が設計したプロセッサの評価を行う際に、プロセッサ外部を意識しなくても評価を可能にするため、プロセッサ外部とのやり

とりを統一化したシステムを FPGA ボード上に構築した。

本稿では、ハード/ソフト・コラーニングシステムでの協調学習法、基本命令セット MONI、プロセッサ評価システムについて述べた後、MONI による各種プロセッサ、MONI の拡張命令セットである TOHD 命令セットによるプロセッサの実装結果を示す。これらの経験を踏まえ、本システムでプロセッサ開発する際に必要となる開発支援環境について検討する。

## 2. ハード/ソフト・コラーニングシステム

### 2.1 システムの概要

ハード/ソフト・コラーニングシステムとは、プロセッサアーキテクチャの理解と、それを意識したプログラミング学習を行うためのハードウェアとソフトウェアの協調学習システムである。ソフトウェア面では、アーキテクチャが可変な命令セットシミュレータを用いて、プロセッサアーキテクチャの理解、アセンブリ言語や C 言語で設計したプログラムを各アーキテクチャで評価し、アーキテクチャに最適なプログラミングを学習する[3]。シミュレータではシングルサイクル、マルチサイクル、パイプライン、スーパースカラの4つのアーキテクチャで命令レベルシミュレーションが可能である。また、最適化コンパイラの設計を通じて各種アーキテクチャに対する最適化手法を理解する。ハードウェア学習では、シミュレータで理解したプロセッサアーキテクチャの知識を基に、HDL を用いて設計した MPU を実際に FPGA 上に実装し、プロセッサの設計と実現方法を理解することを目標とする[1]。利用者が自らプロセッサのアーキテクチャ、命令セットを考案・設計し、それを実装することによって、プロセッサアーキテクチャの更なる学習ができると考える。システムの構成を図 1 に示す。図 1 において左側は主にソフトウェア学習を示し、右側がハードウェア学習を示す。

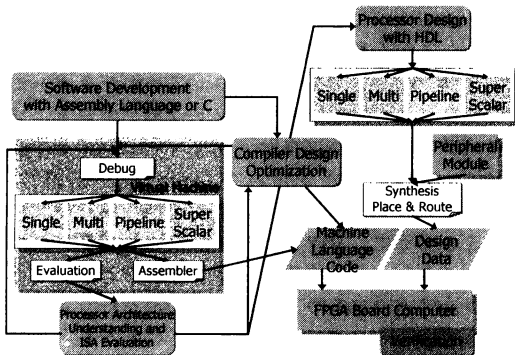


図 1: ハード/ソフト・コラーニングシステム

### 2.2 MONI 命令セット

基本命令セット MONI は、教育システムにおけるアセンブリプログラミングのしやすさを考慮して定義した MIPS のサブセットで全 43 命令から成る。図 2 に MONI 命令形式を示す。MONI 命令セットは 4 つの命令形式(R、I5、I8、J)を持ち、加減算、論理演算、シフト、セット命令に即値アドレッシングとレジスタ・アドレッシングの両方を備え、メモリアクセスはレジスタ間接アドレッシングを採用する。レジスタ間演算は 3 オペランド方式を採る。

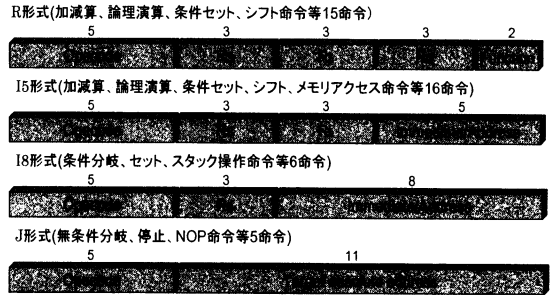


図 2: MONI 命令形式

### 2.3 FPGA ボードコンピュータ

FPGA ボードコンピュータとは、コラーニングシステムの利用者がプロセッサアーキテクチャの理解と実践を行うために使用する。本研究では Celoxica 社 RC100 ボードを使用し、ボード上には FPGA として Xilinx 社 Spartan2(xc2s200-5fg456)が搭載されている。また、外部メモリとして Intel 社製の 8MB FlashRAM を使用する。FPGA ボードコンピュータのモジュール構成を図 3 に示す。MPU が使用するデータ・命令メモリは Xilinx Spartan II FPGA 上の Block RAM を使い、それ以外のモジュールは VerilogHDL を用いて設計した。DMA コントローラは MPU を介さないモジュール間データ転送を管理し、BUSController は DMA 転送時に用いるシステムアドレスから、実際の内部アドレスに変換するアドレスデコードを行う。BoardSequencer は FPGA ボードコンピュータにおけるシステムの全体制御、DMA 転送時におけるバス・アービタの役割を担っている。また外部からのデータ転送を容易にするために、システムアドレス空間と TDI (Transfer Data Information)を用意する。システムアドレス空間には、FPGA 外部バスに接続されている Flash RAM と FPGA 内部バスに接続されている命令メモリ、データメモリ等を同一のアドレス空間に割り当てる。TDI とは DMA 転送時に用いるデータ(情報)である。TDI は 64bit のレジスタで構成され、データメモリコンフィギュレーション用、命令メモリコンフィギュレーション用、及

び MPU による演算結果書き戻し用の 3 つが用意されている。TDI レジスタには上位 24bit に転送元アドレス、中位 24bit に転送先アドレス、下位 16bit に転送量 (1 ワード (2 バイト) 単位) が格納されている。システム起動後、Board Sequencer が FlashRAM から読み出し、これを元に DMA 転送が行われる。ボードコンピュータの動作手順は、前準備としてアセンブラを用いて命令列を作成し、データと共に Flash RAM に書き込む。また、命令列とデータより TDI を作成し、Flash RAM に書き込む。次に、システムを動作させるとデータと命令をメモリに DMA 転送し、DMA 転送が完了すると MPU が動作する。その後、演算結果が Flash RAM に書き込まれる。

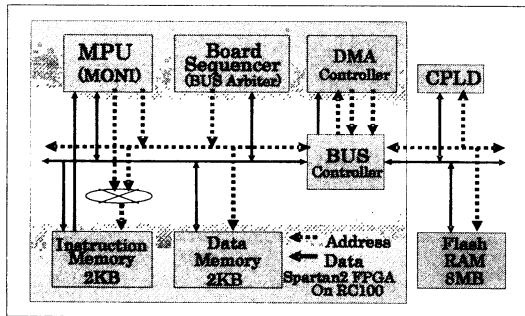


図 3: FPGA ボードコンピュータのモジュール構成

### 3. MONI 各種マイクロプロセッサの実装

#### 3.1 シングルサイクルプロセッサ(SP)

MONI プロセッサは MIPS アーキテクチャのサブセットとして、教育的に理解しやすいデータバス構築を目指して設計した。MONISP のデータバスを図 4 に示す。

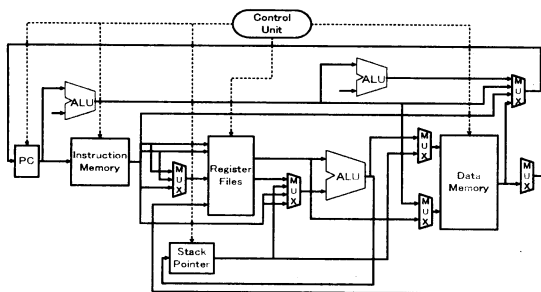


図 4: MONISP のデータバス

#### 3.2 マルチサイクルプロセッサ(MP)

MONIMP は以下の 5 つのステップで動作する。

1. 命令フェッチ
2. 命令デコード、レジスタフェッチ、分岐先アドレス計算

3. 演算実行、メモリ読み出し、分岐完了、スタックポインタ変更
4. メモリ書き込み、レジスタ書き込み
5. サブルーチンへの分岐完了

ストア、条件分岐、スタックからのポップ、jump 命令は 3 サイクルで終了し、サブルーチンへの分岐のみ 5 サイクル要する。その他の命令は 4 サイクルで終了する。また、命令・データメモリは統合することも可能だったが、シングルサイクル、パイプラインプロセッサとの比較しやすさなど教育的要素を考慮してハーバードアーキテクチャを採用した。MONIMP のデータバスを図 5 に示す。

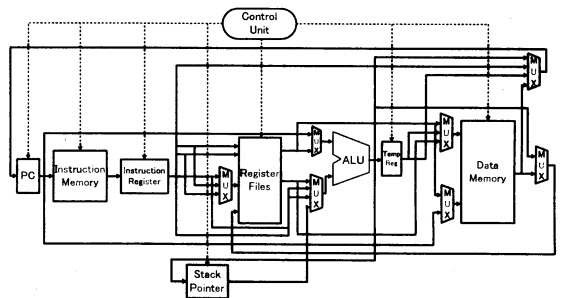


図 5: MONIMP のデータバス

#### 3.3 パイプラインプロセッサ(PP)

MONIPP は MONISP を基本とし、パイプラインレジスタを挿入して 5 段パイプラインを実現した。各ステージでは以下の処理を行う。

- IF(第 1)ステージ : 命令フェッチ
- ID(第 2)ステージ : 命令デコード、分岐判定
- EX(第 3)ステージ : 演算実行
- MEM(第 4)ステージ : メモリアクセス
- WB(第 5)ステージ : レジスタ書き込み

また、各種ハザード回避のためにハザード検出ユニットとフォワーディング判定ユニットを設計し、必要に応じてデータフォワードとパイプラインストールを行う。MONIPP のデータバスを図 6 に示す。

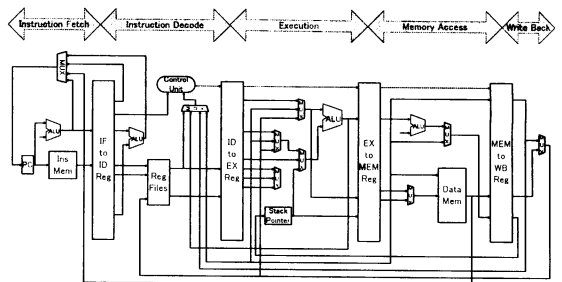


図 6: MONIPP のデータバス

### 3.4 各種プロセッサの性能

コラーニングシステムにおける基本命令セット MONI による各アーキテクチャで設計・実装したプロセッサの性能の詳細を表 1 に示す。コラーニングシステムでは Flash RAM への演算結果の書き込みを保証するために、利用者への要求動作周波数を 10MHz に設定しているが、3 種類とも要求を満たしている。

表 1：各プロセッサの回路規模と動作周波数

	Register	LUT	Number of gates	Usage rate (%)	Frequency (MHz)
Single Cycle	161	895	44,000	22	16.4
Multi Cycle	228	1,137	56,000	28	29.2
Pipeline	369	1,044	58,000	29	23.4

システム全体の回路規模と動作周波数を表 2 に示す。MONIPP を実装したシステム全体の回路規模は約 15 万システムゲートとなり、資源利用率は 75% になる。

表 2：システム全体の回路規模と動作周波数

	Register	LUT	Number of gates	Usage rate (%)	Frequency (MHz)
Board Sequencer	333	718	36,000	18	26.3
DMA Controller	197	617	34,000	17	47.9
BUS Controller	94	369	16,000	8	65.7
MPU (Pipeline)	369	1,044	58,000	29	23.4
ALL	1,204	2,616	150,000	75	23.3

### 3.5 スーパースカラプロセッサ(SSP)

MONISSP のデータバスは教育用、また FPGA の回路規模を考慮した結果、現段階では MONISP を基に 2 命令同時発行可能な構造とした。MONISSP は命令ウィンドウに格納された命令の資源競合、データ依存、手続き依存などのハザードをアービトレイターによって検出し、ハザードが検出されないか、回避可能ならば 2 命令を同時に発行する。もし、ハザードが検出されたならば 1 命令のみ発行する。命令発行ポリシーはイン・オーダー発行&イン・オーダー完了である。MONISSP のデータバスを図 7 に示す。また、MONISSP の回路規模と動作周波数を表 3 に示す。

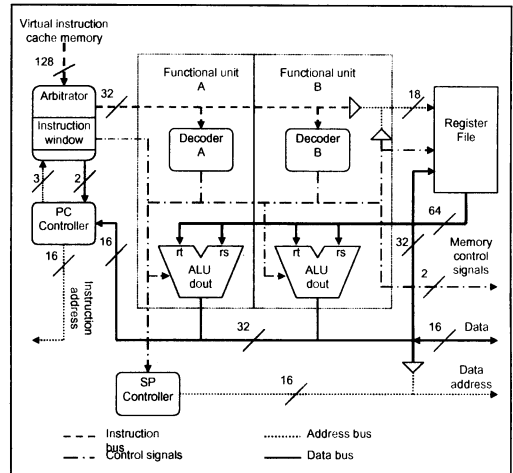


図 7：MONISSP のデータバス

表 3：MONISSP の回路規模と動作周波数

	Register	LUT	Number of gates	Usage rate (%)	Frequency (MHz)
Super Scalar	215	2,346	102,000	51	1.8

### 3.6 考察

MONI 命令セットでの各種プロセッサを実装したことにより、コラーニングシステムにおける学習の第 1 ステップである、MONI 命令セットを用いたプロセッサアーキテクチャの理解、アセンブリプログラムでのソフトウェア開発、各種アーキテクチャでのプロセッサ設計・実装という学習体系が確立できた。しかし、プロセッサの性能には課題を残す結果となった。MP は SP に比べ最高動作周波数は約 1.8 倍向上しているが、1 命令に 3~5 ステップ要するため性能は低下する。PP は最高動作周波数が約 1.4 倍向上しているため、開発ソフトウェアでのストール回数低減が重要になる。SSP は SP と比較して、CPI が 0.57~0.67 と良好な結果を出しているが、最高動作周波数が 1.8MHz と SP の約 10 分の 1 となっているため性能向上は見込めない。表 4 に SP と SSP の性能比較を示す。

表 4：MONISP と MONISSP の性能比較

	Single Cycle	Super Scalar	CPI
	Number of clocks	Number of clocks	
Bubble sort	59,732	37,973	0.635
Selection sort	30,443	20,604	0.676
Insertion sort	13,643	7,895	0.578

プロセッサ性能に関する今後の課題としては、まずシステム利用者がアーキテクチャの違いによる性能向上をより実感できるように、最高動作周波数が低くなる原因の究明を行う。その後、SSPをパイプライン化、イン・オーダー発行&アウト・オブ・オーダー完了、更にはアウト・オブ・オーダー発行&アウト・オブ・オーダー完了可能な構造にし、更なる性能向上を目指す。

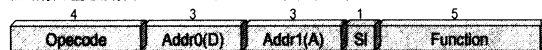
#### 4. TOHD 命令セットマイクロプロセッサ

##### 4.1 TOHD 命令セットの概要

カラーリングシステムにおける学習の第2ステップは、MONI 命令セットでの学習を踏まえて、学習者が独自に命令セット、アーキテクチャを考案し、設計・実装することを目標としている。今回考案した TOHD 命令セットは、MONI 命令セットを改良したものである。MONI 命令セットがアセンブリプログラミングのしやすさを考慮しているのに対して、TOHD 命令セットは性能向上を目標に作成した。図 8に TOHD 命令セットの構成を示す。命令形式は4つあり全54命令からなる。レジスタ間演算は2オペランド方式を採用する。

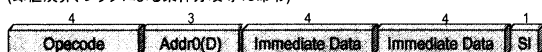
###### Register operations

(加減算、論理演算、シフト、メモリアクセス命令等40命令)

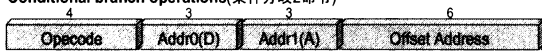


###### Immediate data/Absolute address operations

(即値演算、フラグによる条件分岐等10命令)



###### Conditional branch operations(条件分岐2命令)



###### Unconditional branch operations(無条件分岐2命令)



図 8 : TOHD 命令形式

##### 4.2 シングルサイクルプロセッサ(SP)

TOHD プロセッサでは2ビットのフラグを採用している。キャリーフラグは演算命令、コンディションフラグは分岐命令で効果を示す。TOHDSPのデータバスを図9に示す。

##### 4.3 考察

TOHDSP の回路規模と動作周波数を表5に、MONI 命令セットと TOHD 命令セットの性能比較を表6に示す。実行命令数は約4割減の効果が得られた。

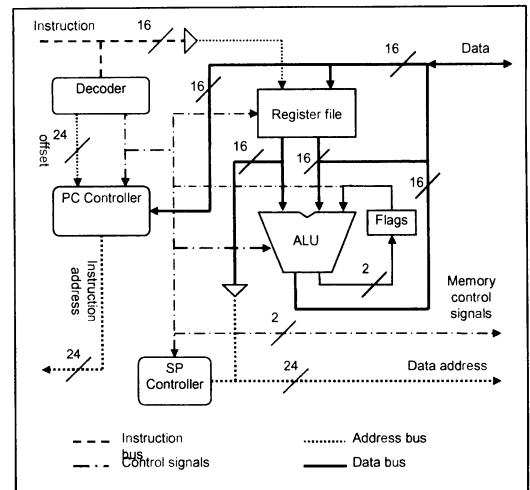


図 9 : TOHDSP のデータバス

表 5 : TOHD プロセッサの回路規模と動作周波数

	Register	LUT	Number of gates	Usage rate (%)	Frequency (MHz)
TOHD Single	183	1,007	48,000	24	17.1

表 6 : MONI 命令セットと TOHD 命令セットの性能比較

	MONI Instruction Set		TOHD Instruction Set	
	Number of clocks	Increase	Number of clocks	Increase
Bubble sort	59,732	100%	42,252	141%
Selection sort	30,443	100%	16,452	185%
Insertion sort	13,643	100%	7,585	180%

しかし、独自命令セットによるプロセッサ設計では、MONI 命令セットによる学習にはプロセッサシミュレータやアセンブラが用意しているのに比べると、開発環境の不足という問題がある。

#### 5. 今後の課題

MONI 命令セット、独自命令セットに関わらずプロセッサ設計時に以下のような問題がある。

1. ソフトウェアベースのシミュレータでは動作速度が遅い
2. CAD ツールシミュレーションでの大規模テストプログラム評価時のエラー箇所の発見
3. FPGA 外部のメモリ等デバイスはシミュレーション不可

そこで我々は、FPGA 内に実装されたプロセッサをデバッグするための回路を作成し、ホスト PC との通

信により PC 上で容易にデバッグ可能なプロセッサデバッガの設計を行う。使用する FPGA ボードは RC200 ボードを検討中である。デバッガの最低限の機能として、以下に示すものを検討中である。

1. プログラムのブレーク実行、1 命令実行、1 クロック実行
2. プロセッサ内部の信号、レジスタ・PC の値を表示・置換
3. FPGA ボード上のメモリ内容の表示・置換

以上の機能を実現するために考案したのが図 10 に示すシステム改良後の MPU 周辺ブロック図である。現在使用している FPGA ではグローバルクロックを分周させ、2 種類のクロックを使用することが可能であるため、システム改良後は MPU 専用クロックとシステム制御用クロックに使い分ける。ホスト PC からブレーク信号が入力されると、FPGA 内のブレーク回路により MPU 専用クロックの供給がストップされる。MPU の動作が停止することにより、その時点でのプロセッサ内部のレジスタの値を表示させることが可能となる。また、RC100 ボードでは命令・データメモリは FPGA 内の IP コアを利用しているが、RC200 ボードではボード上に SRAM が 2 つ装備されているのでそれらを使用する。制御用回路からメモリに対し信号を送信し、メモリ内容の表示・置換を行う。

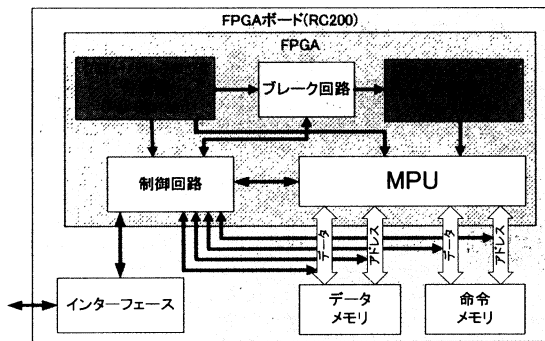


図 10：システム改良後の MPU 周辺ブロック

更に、学習者が独自命令セットによるプロセッサ設計時のソフトウェア開発支援として、汎用的なアセンブラとシミュレータの開発も並行して行う。

## 6. おわりに

本稿では、ハード/ソフト・カラーリングシステムにおける協調学習法、カラーリングシステムにおける基本命令セット MONI による各種プロセッサの実装、及び拡張命令セット TOHD によるプロセッサの実装について述べた。カラーリングシステムにおける学習の第 1 ステップである MONI 命令セットでの学習体系が確

立したことを示し、第 2 ステップである学習者による独自命令セットの考案とプロセッサ設計の方針を示した。しかし、学習者がアーキテクチャによる性能向上をより実感するには性能の面で課題が残った。今後は、課題の原因究明とプロセッサ開発支援環境の構築を目指す。

## 文 献

- [1] 中村,池田,小柳,山崎: プロセッサアーキテクチャ教育用 FPGA ボードコンピュータシステムの開発, FIT2004,LC-008,2004.
- [2] 池田,中村,大八木,Tuan,山崎,小柳: ハード/ソフト・カラーリングシステムにおける FPGA ボードコンピュータの設計,情報処理学会第 66 回全国大会論文集,5T-5,2004.
- [3] 大八木,池田,山崎,小柳: ハード/ソフト・カラーリングシステムにおけるアーキテクチャ選択可能なプロセッサシミュレータの設計,情報処理学会第 66 回全国大会論文集,5T-6,2004.
- [4] H. Anh Tuan, K. Nakamura, K. Yamazaki and S. Oyanagi : Superscalar microprocessor design in a Hardware/Software co-learning system,FIT2004,C-030,2004.
- [5] John L. Hennessy and David A. Patterson 著, 成田光彰 訳:コンピュータの構成と設計(上)(下),日経 BP 社,1999.
- [6] 田中,久我,末吉,小羽田:教育用マイクロプロセッサ KITE とその開発支援環境,情報処理学会研究報告,Vol.93, No.49(ARC-100), pp.59-66,1993.
- [7] 末吉,小羽田,野崎,田中,久我:FPGA を利用した教育用マイクロプロセッサ KITE-2 システムソフトウェア教育への対応 情報処理学会研究報告,Vol.94, No.50(ARC-106), pp.25-32,1994.
- [8] 井上,中垣,大内,末吉:教育用 RISC 型マイクロプロセッサ DLX-FPGA とそのラビッドシステムプロトタイピング,電子情報通信学会技術研究報告,Vol.95, No.25(ICD95 11-22), pp.71-78,1995.
- [9] 桜井,長沢,宮内,石川:教育用 RISC 型マイクロプロセッサ MITEC-II を用いた演習環境の開発及び MITEC-II を用いた演習の実施,情報処理学会研究報告,Vol.2001, No.101(CE-61), pp.47-54,2001.
- [10] 土江,佐々木,弘中,児島:教育研究用スーパーカラ・プロセッサ・シミュレータ Mikage の概要,情報処理学会研究報告, Vol.96, No.80(ARC-119), pp.107-112,1996.
- [11] 今井,古川,井面,白木,石川:計算機システム教育のためのビジュアルシミュレータ VisuSim,情報処理学会研究報告, Vol.2001, No.34-(CE-59), pp.77-84,2001.
- [12] 下川,西野,早川,システムソフトウェア教育支援環境「港」における FPGA を利用した演習環境の開発 電子情報通信学会技術研究報告,Vol.102, No.697(ET2002 95-119), pp.7-12,2003.
- [13] 細川,松本,中村,FPGA エミュレータを利用したカスタムプロセッサ向け高機能デバッガ,第 4 回リコンフィギュラシステム研究会論文集,2004.