

## FPGA を用いたスーパースカラ設計教育に関する一考察

高橋 隆一 †, 大岩 元 ‡

† 広島市立大学 情報科学部

〒 731-3194 広島市安佐南区大塚東 3-4-1

‡ 慶應大学 環境情報学部

〒 252-8520 藤沢市遠藤 5322

†tryuichi@ce.hiroshima-cu.ac.jp, ‡ohiwa@sfc.keio.ac.jp

情報化社会の進展に伴い、高機能マイクロプロセッサの開発技術がこれまでになく求められる時代を迎えた。FPGA はハードウェア記述言語を用いて動作仕様を指定するだけで自由にマイクロプロセッサを実現できる万能の機能素子である。学習者は正統的周辺参加によって実際の仕事の過程に従事することで業務遂行能力を獲得するという考え方がレイヴらによって提唱されている。スーパースカラプロセッサの算術論理演算回路とその上での並列処理が可能か否かを判定する機構の HDL 記述を完成させる指導で、試行初年度は 39 人中 7 人、試行の 2 年目には 47 人中 14 人の学部 3 年生が 15 週間でスーパースカラプロセッサを稼働させることに成功した。

**FPGA, スーパースカラ, 正統的周辺参加,  
状況に埋め込まれた学習, 設計教育**

## A Note on Superscalar Processor Design Education using FPGA

**Ryuichi TAKAHASHI†, Hajime OHIWA‡**

†Faculty of Information Sciences, Hiroshima City Univ.

3-4-1 Ozukahigashi, Asaminami-ku, Hiroshima City, 731-3194 Japan

‡Shonan-Fujisawa Campus, Keio Univ.

5322 Endoh, Fujisawa City, 252-8520 Japan

Skill to develop microprocessors is widely required these days. Field-Programmable Gate Array (FPGA) is a functional device which can be used to implement microprocessors only by specifying the behavior in hardware description language (HDL). Lave et al. suggested an idea that the individual learner acquires the skill to perform by actually engaging in the process, under the attenuated conditions of legitimate peripheral participation(LPP). 21 out of 86 junior students succeeded in implementing superscalar CISC/RISC processors within 15 weeks by engaging an effort to complete the entire description of superscalar processors without having the description for the parallel issue for ALUs.

**FPGA, Superscalar, legitimate peripheral participation,  
Situated learning, design education**

## 1 はじめに

情報化社会の進展に伴い、様々なマイクロプロセッサが社会の隅々にまで広く、深く浸透するようになった。高度な機能を持つようになった携帯電話機や家庭電化製品にもマイクロプロセッサは広く用いられている。10段を越える段数をもつスーパーパイラインプロセッサ、同時実行可能な命令を陽に指定する VLIW 型プロセッサなど様々なマイクロプロセッサの開発技術が、これまでになく求められる時代を迎えた。

FPGA はユーザが手元でプログラムできる LSI であり、ASIC に代るものとして 1990 年代中頃から広く用いられるようになった。特に SRAM によって電氣的に仕様を記録する FPGA は何度でも書き換えができ、FPGA を搭載したボードを用意することで、ハードウェア教育が実施できる [3]。

FPGA を用いたマイクロプロセッサ設計教育の事例は多いが、本設計教育環境 **City-1**[5] は命令セットアーキテクチャを自由に定めさせていること、製作までを行わせて創る喜びを実感できることが特徴的であった。既存のボードを用い、課題を変えない場合は、先輩から解答を受け継いでしまい、大多数の学生が何もしないという事態を引き起こしかねない。自由な命令セットアーキテクチャで製作までをさせる **City-1** にこの危険はない。

**City-1** にも問題がないわけではなかった。日程が遅れたり、大多数の学生が命令セットアーキテクチャの工夫程度に始終するなどしていた。これらの問題点が実施 6 年目にあたる平成 13 年 (2001 年) 度に改善された。

学習者は正統的周辺参加 (LPP) という、ゆるやかな条件のもとで実際に仕事の過程に従事することによって業務遂行能力を獲得するのであり、抽象的な表象を内化するわけではないという指摘がレイヴらによってなされた [1]。正統的周辺参加という観点からするなら、ゆるやかな条件のもとで、実際の業務の課

程のどの部分をどう課題として与えるかが重要な役割を果たす。正統的周辺参加においては、観察によって製品についての第一次近似をつくる段階は「入り口」(way-in) と呼ばれ、全工程をたどる段階は「練習」(practice) と呼ばれている。

スーパースカラ設計教育を試行した平成 13 年 (2001 年) 度の **City-1** では、39 人中 7 人、試行 2 年目の平成 14 年度には 47 人中 14 人がスーパースカラプロセッサの実現に成功した。初期段階では、算術論理演算回路 (ALU) と、これを用いて 2 つの命令を並列に処理できるかを判定する回路の設計に集中した指導を行った。このことが正統的周辺参加における適切な「入り口」になっていたと思われる。

2 節では、平成 12 年 (2000 年) 度に至るまでの 5 年間の実施状況について述べる。3 節では実施 6 年目にあたる平成 13 年 (2001 年) 度からの指導について述べ、4 節で実施 6 年目と 7 年目の履修状況について述べる。5 節では正統的周辺参加について述べ、履修した学生が、どのように状況に埋め込まれたかについて考察する。

## 2 5 年間の実施状況

広島市立大学に最初の 3 年生が誕生した平成 8 年 (1996 年) 度から実施 5 年目にあたる平成 12 年 (2000 年) 度に至るまでの 5 年間では、パイプライン設計教育に焦点を置いた指導を行った。260 台のコンピュータが設計され、製作され、84 % にあたる、220 台のコンピュータが、課題のプログラムを走らせることに成功した。220 台のうち、125 台が CISC マシン、95 台が RISC マシンであった。

本学生実験では RISC を下記によって特徴付けている：

- [1.] 固定フォーマット
- [2.] レジスタ上での算術論理演算
- [3.] 1 クロック 1 命令というスループット

## 2.1 履修の目的と学生の予備知識

学部3年の15週で、1人1台、思い思いのコンピュータを設計し、製作し、課題として与えられたプログラムを、自らの機械語で書き下し、自らのコンピュータ上で走らせることによって、「コンピュータの仕組みと設計方法の基本を学ぶ」ことが目的である。

学部2年前期の「論理回路」と「論理回路演習」によって、論理回路設計のための基本的な知識 [7] はすでに有していることを前提にできた。「情報工学基礎実験」では、既成のマイコンによってアセンブラを学び、カウンタ程度の簡単な回路の設計と製作を経験済みであった。学部2年後期のコンピュータシステム I、3年前期のコンピュータシステム II では、コンピュータにおける2進数の表現方法にはじまり、入出力やパイプライン化に至るひとつおりを学んでいた。

## 2.2 シラバス

講義概要には実際の設計／製造工程に添ったトップダウンの作業日程を掲載していた。

- [1.] システム分析 (商品企画)
- [2.] 方式設計
- [3.] 命令セットアーキテクチャ
- [4.] 機能設計／モジュール分割
- [5.] データパスの設計
- [6.] データパスの検証
- [7.] 制御部の設計
- [8.] FPGA ボードの製作
- [9.] FPGA ボードの動作確認
- [10.] FPGA の実装
- [11.] システムのデバッグ
- [12.] システムの動作確認
- [13.] ホームページ作成
- [14.] 終了判定 (出荷検査)

## 2.3 パイプライン設計教育

最初の3年生が誕生した平成8年(1996年)度前期には、CISC-1と名付けた記述例を、実施2年目の平成9年(1997年)度前期にはCISC-3と名付けた記述例を提供し、パイプライン設計教育に焦点を置いた指導を行った。

### 2.3.1 CISC-1

CISC-1はVerilog-HDLの7個のモジュールからなる573行という記述量だった。CISC-1は結線論理制御方式を、あたかもマイクロプログラム制御方式であるかのように表現している点に特徴があった。結線論理制御方式とマイクロプログラム制御方式との間に本質的な相違はない [4]。CISC-1は、命令ごとに所要クロックサイクル数の異なるようなコンピュータに対する水平型の制御(1制御点1ビット)が、結線論理制御でも容易に実現できることを示していた。

### 2.3.2 CISC-3

CISC-3はVerilog-HDLの9個のモジュールからなる639行という記述量だった。このCISC-3の予約表(reservation table)を手直しすることを課題として与えた [5]。

City-1ではメモリ上のデータを算術論理演算のオペランドとして指定していることをもってCISCと見做していた。RISCはロード／ストア命令を用いて、あくまでレジスタ上で算術論理演算をするコンピュータだという認識であった。図1にCISC-3の予約表を示す。

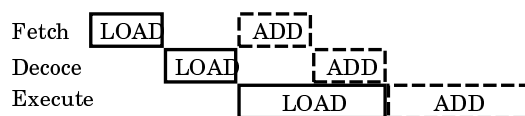


図1. CISC-3の予約表

RISCの場合は、オペランドは既にレジスタ上にあるため、足し算をするときもメモリ上のオペランドをフェッチしたりする必要は

ない。このため実行 (execution) の所要時間は命令フェッチやデコードと同じだと見做せる。**CISC-3** を RISC に手直した場合の予約表を図 2 に示す。

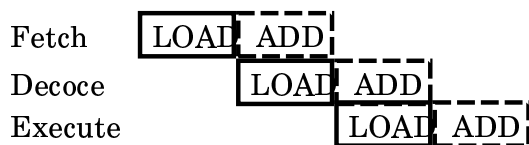


図 2. RISC に改造した場合の予約表

パイプラインを起動ことはイニシエーションと呼ばれることがある [9]。**CISC-3** を RISC に手直しすることでイニシエーションの間隔は半分になり、スループットが 2 倍に向上する。

CISC のままオペランドをフェッチするステージを追加することによってもスループットが 2 倍に向上する。**CISC-3** にオペランドフェッチのステージを追加する手直しをした場合の予約表を図 3 に示す。

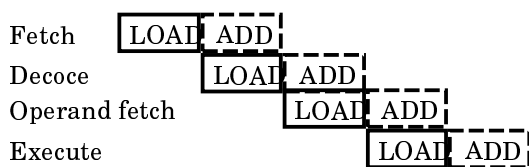


図 3. 4 段パイプラインに手直した場合

単純な命令セットアーキテクチャであっても、条件付き分岐予測の扱いにはインターロック機構の導入を要する。ハザードを回避する機構を導入することでデータ依存関係について学ぶことが期待された。

## 2.4 履修状況

実施 3 年目にあたる平成 10 年 (1998 年) 度までは日程が 2 週間程度遅れる傾向があった。全員の足並みを揃えるために、課題は小刻みに与えるという指導を行っていた。

平成 11 年 (1999 年) 度は、期限内に完成させることを強く要求し、すべての資料をできる限り早い時期に提供してしまうことで、正規の履修期間内で 47 人中 41 人が課題として

与えたプログラムを走らせることに成功したが、内容の理解が不十分に思われた。

平成 12 年 (2000 年) 度は、期限内に内容を理解できる学生を増やすために 49 人中 39 人が完成させる程度に止める必要に迫られた。

ごく一握りの優秀な学生が自らのマシン用にアセンブラを開発するなどの例が見受けられることはこの 5 年間変らなかつた。しかし、大多数の学生の工夫が命令セットアーキテクチャを云々する程度に止まっていた。これらの問題点が実施 6 年目にあたる平成 13 年 (2001 年) 度に改善された。

平成 13 年 (2001 年) 度からは、ネットワーク実験が学部 3 年前期に生まれ、本設計教育環境 **City-1** は情報工学科の学部 3 年後期に組み直された。これを機に、高度にすぎると思われていたパイプライン設計教育を、スーパースカラ設計教育に改めた。スーパースカラは、実行時に動的に命令の並列性を解析し、可能なら複数の命令を同時発行する技術であった。

## 3 スーパースカラ設計教育

実行時に動的にデータ依存関係を解析して可能なら複数の命令を同時発行するスーパースカラプロセッサは、データ依存関係を直接的に扱う点で、短期間にデータ依存関係の意味を学生に理解させる上で特に有効だと思われた。算術論理演算回路とその上での並列処理が可能か否かを判定する機構はスーパースカラプロセッサの心臓部であった。条件付き分岐予測を扱うためのインターロック機構を置く必要性はフォワーディングで回避するものとした。

### 3.1 RISC-2SS2

実施 6 年目にあたる 2001 年から、**RISC-2SS2** と名付けたスーパースカラ RISC プロセッサの記述例を提供することにした。

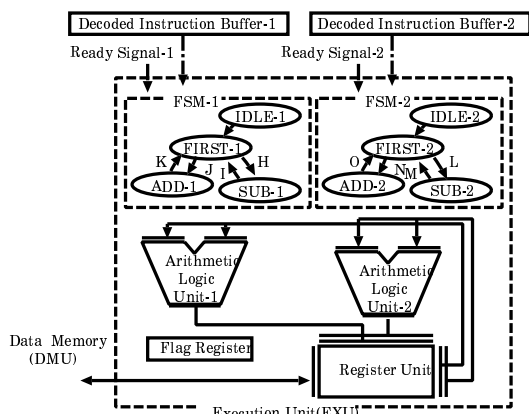


図 4. RISC-2SS2 の実行ユニット

**RISC-2SS2** は Verilog-HDL の 9 個のモジュールからなる 983 行で記述できた。2 段パイプラインの RISC マシンであり、可能なら 2 つの命令を同時発行可能なスーパースカラプロセッサであった。従来から提供している記述例 **CISC-1, CISC-3**[5] と比較して、特に大規模な記述とはならなかった。

図 4 に **RISC-2SS2** の実行ユニット (**EXU**) のブロック図を示す。

2 つの算術論理演算ユニットは個別にステートマシン **FSM-1, FSM-2** で制御されており、並列に合計 2 つの算術論理演算を処理できる。

図 5 に **RISC-2SS2** のフェッチ/デコードユニット (**FDU**) のブロック図を示す。

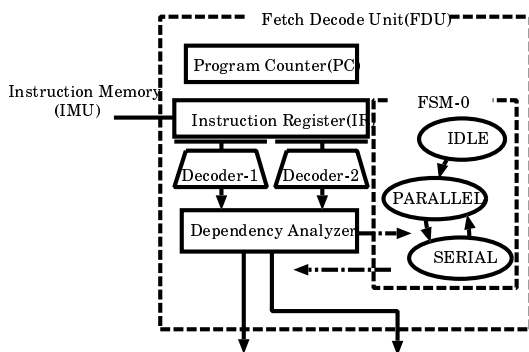


図 5. RISC-2SS2 のフェッチ/デコードユニット

スタートスイッチが押されると、ステートマシン **FSM-0** は **PARALLEL** の状態に遷移して、フェッチされた 2 つの命令の演算が同時処理可能なら、そのまま 2 つの命令のデコード結果を実行ユニットに送り込む。データ依

存関係があつて同時実行できなければ最初の命令だけを実行ユニットに送り、**SERIAL** の状態に遷移して 2 番目の命令を送り込んで **PARALLEL** の状態に戻る。同時処理可能か否かを判定させる機構が制御するのはこの **FSM-0** の状態遷移である。

### 3.2 ALU と FDU の記述の手直し

学生にはスーパースカラプロセッサの算術論理演算回路とその上での並列処理が可能か否かを判定する機構の HDL 記述を次のように手直しするよう指示した。

- 提供した記述例 **RISC-2SS2** の算術論理演算ユニットは非負数の演算しか行えないものとした。これを 2 の補数を使い、可能ならオーバーフローを検出して内部割り込みを発生させるように指示した。オーバーフローは 2 の補数表現において、符号位置への桁挙げと符号位置からの桁挙げが異なるかで検出できる。
- データ依存関係には次の 3 通りがある：
  - [1.] 真の依存 (true dependency) 先行する命令が書き込む結果を後続する命令が使う
  - [2.] 出力依存 (output dependency) 先行する命令が書き込む結果を後続する命令が書き改める
  - [3.] 逆依存 (anti-dependency) 先行する命令が読み出すデータを後続する命令が書き改める

提供した記述例 **RISC-2SS2** において 2 つの演算を同時処理可能か否かを判定させる記述では、出力依存がないかだけしか調べないものとした。これをすべてのデータ依存関係を調べるように書き改めるよう指示した。

## 4 履修の状況

実施6年目にあたる2001年度は、履修登録した学部3年生50人中39人が課題として与えたユークリッドの互除法を走らせ、39人中7人がスーパースカラプロセッサを稼働させることに成功した。39人中20人が算術論理演算回路に2の補数を用いて負の数を扱うことができるよう改造し、20人中14人がオーバーフローによる内部割り込みを発生させる手直しを行った。

表1に、実施6年目の7つのスーパースカラプロセッサの概要を示す。

表1. 実施6年目のスーパースカラ

	命令数	レジスタ数	場合の数	規模
例1	12	3	5	142
例2	16	4	70	340
例3	19	4	41	318
例4	16	4	52	350
例5	12	3	25	216
例6	16	3	19	202
例7	16	3	4	179

実施7年目にあたる2002年度は、履修登録した学部3年生54人中47人が課題として与えたユークリッドの互除法を走らせ、47人中14人がスーパースカラプロセッサを稼働させることに成功した。47人中20人が算術論理演算回路に2の補数を用いて負の数を扱うことができるよう改造し、20人中18人がオーバーフローによる内部割り込みを発生させる手直しを行った。

表2に、実施7年目の14のスーパースカラプロセッサの概要を示す。

命令数は8から27、レジスタ数は2から4と様々で、全員が独自の命令セットアーキテクチャを定めていた。同時発行可能であると判断できる場合の数は最大185に及んだ。

実施7年目の工夫はこれだけに止まらなかった。表2の「場合の数」の後に括弧で示したOOCは乱終了の恩恵を享受していること、

MPEは分岐点で両方のパスを追跡する多重パス実行を行って完全な分岐予測と等価な結果を得ていること、SPMは課題のユークリッドの互除法のための専用命令を備えていることを表している。例の番号に続く括弧内のCはCISCマシン、RはRISCマシンであることを表している。

表2. 実施7年目のスーパースカラ

	命令数	レジスタ数	場合の数	規模
例1(C)	12	4	64(OOC)	319
例2(R)	17	4	185	294
例3(R)	16	4	3	200
例4(R)	16	4	4	263
例5(R)	16	4	4	186
例6(C)	19	2	7(OOC)	209
例7(R)	23	7	121(MPE)	320
例8(C)	22	2	6(OOC)	318
例9(R)	14	4	3(SPM)	228
例10(R)	16	4	8	190
例11(R)	18	3	7	197
例12(C)	27	2	26(OOC)	232
例13(C)	16	2	8(OOC)	182
例14(R)	8	3	4	149

学生が製作したスーパースカラプロセッサの写真を図6に例示する。

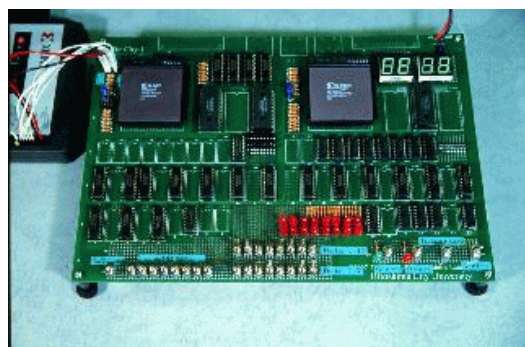


図6 A Superscalar FPGA computer example

## 5 状況に埋め込まれた学習

実施6年目と7年目の成功は、算術論理演算回路(ALU)と、これを用いて2つの命令を

並列に処理できるかを判定する回路の設計に集中した指導を行ったことが正統的周辺参加における適切な「入り口」になっているためだと思われた。

### 5.1 古典的主知説

ヴィゴツキーらに代表されるように、従来、学習は内化 (internalization) の過程であると考えられてきた。学習者が単独で問題に取り組むときに示す問題解決能力と、より経験を積んだ人に助けられたり共同であったりで取り組むときに示す問題解決能力との距離が問題にされ、やがては支援なしで業務を遂行できるようにしようとする教授的なアプローチが活発化した。レッスン (訓練) も、ときには命題として記述可能な抽象的な表象を内化するためであるとされてきた [1]。

しかし、実務から切り離された教授的な訓練プログラムでは、学習状況を処理する学習者の能力は、その技能を遂行する能力とは別のものになってしまう傾向があった。学習者は学習状況を処理する能力を身につけはするが、実際の仕事を遂行する能力は身につけない [1]。

### 5.2 正統的周辺参加

レイヴとウエンガーは、学習によって獲得される技能がきわめて相互作用的であることに注目し、学習者は正統的周辺参加 (Legitimate Peripheral Participation:LPP) という、ゆるやかな条件のもとで実際に仕事の過程に従事することによって業務遂行能力を獲得するのだと考えた [1]。問題提起は言語の学習でなされた。レイヴらは、1990年に、意味生成を個々の発話者の頭の中ではなく、社会的相互作用に位置づけた。

正統的周辺参加において、学習者は、熟達者の実践活動に参加はするものの、はじめはごく限られたものであり、最終的な製品には限られた責任しか負わない。そのとき学習者が

獲得するのは変わることのない心的表象ではないし、レッスンは抽象的な表象の集合でもない。レイヴらはリベリアの伝統的な仕立屋を調査した [2]。徒弟は、はじめはアイロン掛けやボタン付けという作業を、実際に生産工程に加わりながら学ぶ。同様に裁縫のしかたを学び、最後に布地の裁断のしかたを身につける。徒弟にとっては、正統的周辺参加が実際の生産工程とは逆に系統付けられていることが最良であるとされていた。アイロン掛けやボタン付けを通して洋服のおおまかな構造を知り、裁縫をすることで洋服を構成する布地の関係を理解し、そのように裁断することの意味を学ぶ。観察期間であり、衣服についての第一次近似をつくろうとする段階は「入り口」 (way-in) と呼ばれる。徒弟が最初から最後まで全工程をあらためてたどる段階は「練習」 (practice) と呼ばれる。

工学にとって創る喜びが命であることは言うまでもない。しかし、教育内容を系統付け、適切な指示を与えることで、より充実した指導が行える。

### 5.3 正統的周辺参加における入り口

スーパースカラ設計教育においては複数の命令を可能なら同時発行するというスーパースカラにとっては本質にあたる部分が観察に最も適した部分になっていたと考えられる。ALUは複数置かれることで複数の命令を同時に処理する資源そのものであり、2つの命令を並列に処理できるかを判定する回路はこれを使いこなすための機構そのものだった。

### 5.4 正統的周辺参加における練習

正統的周辺参加においては、徒弟が最初から最後まで全工程をあらためてたどる段階は「練習」 (practice) とされる。提供した記述例 **RISC-2SS2** の命令セットを改め、それに合わせて上記を手直しすることが設計の全工程をたどる作業になっていた。設計が完

成した後に、製作を行えば、シラバスで示した設計／製造工程に添ったトップダウンの工程をすべてたどることになる。スーパースカラの設計、製作を行った学生は全体の記述を8回程度以上書き換えていた。

本学生実験 **City-1** ではクイックソートやユークリッドの互除法などのアプリケーションを指定して、それが動けばどのようなコンピュータを創っても良いという形でしか仕様を与えていない。スーパースカラを選択した学生は、正統的周辺参加における入り口で算術論理演算回路とその上での並列処理が可能か否かを判定する機構のHDL記述を手直ししてみたのちに、自分なりの命令セットを定めて全工程をたどっていたと考えられる。

## 6 まとめ

本稿では、広島市立大学情報科学部情報工学科3年後期に置かれた設計教育環境 **City-1** の実施6年目と実施7年目に試行したスーパースカラ設計教育を紹介して、正統的周辺参加に配慮することが教育上有効だと思われることを示した。

スーパースカラ設計教育における正統的周辺参加についてより深い考察を行うことが今後の課題である。

本研究で用いているCADシステムの、超高速論理シミュレータはCADENCE社のアカデミック・プログラム、ロジックシンセサイザはSynplicity社のアカデミック・プログラム、自動配置配線プログラムのライセンスはXilinx社のユニバーシティー・プログラムによるものである。

## 参考文献

[1] Lave J. and Wenger E.:Situating Learning: Legitimate peripheral participation, Cambridge Univ. press (1991) 佐伯 胖(訳):

状況に埋め込まれた学習 - 正統的周辺参加 -, 産業図書 (1993)

- [2] 波多野 誼余夫編: 認知心理学5 学習と発達, 東京大学出版会 (1996)
- [3] 末吉 敏則: 教育へのFPGA応用例, 情報処理 Vol.35, No.6, pp.519-529(1994)
- [4] 高橋隆一, 吉村猛: ハイレベルシンセシスの動向, 信学論A, Vol.J74-A, No.2, pp.143-151(1991)
- [5] 高橋隆一, 吉田典可: システムのインテリジェント化を支えるデジタル設計教育, 信学誌 Vol.81, No.9, pp.908-912(1998)
- [6] Ryuichi TAKAHASHI and Noriyoshi YOSHIDA: "Diagonal Examples for Design Space Exploration in an Educational Environment CITY-1", Proc. 1999 International Conference on Microelectronic Systems Education pp.71-73 (1999)
- [7] 当麻喜弘: スイッチング回路理論, コロナ社 (1986)
- [8] 馬場敬信: コンピュータアーキテクチャ, オーム社 (2000)
- [9] Peter M. Kogge: The Architecture of Pipelined Computers, McGraw-Hill, New York (1981)
- [10] Mike Johnson: Superscalar Microprocessor Design, P T R Prentice-Hall (1991)
- [11] Hennessy J. L. and Patterson D. A. : Computer Architecture A Quantitative Approach 2nd edition, Morgan Kaufmann Publishers, Inc. (1996)