

## 教育用パイプライン処理マイクロプロセッサPICO<sup>2</sup>の開発

西村 克信      額田 多政      天野 英晴

慶應義塾大学大学院 理工学研究科 計算機科学専攻

〒 223-8522 横浜市港北区日吉 3-14-1

{nisimura,nukata,hunga}@am.ics.keio.ac.jp

あらまし 大学における計算機教育において、実際に教育用マイクロプロセッサを設計し、FPGA 上に実装する実験・演習は、学生の計算機アーキテクチャに対する理解を深める点で重要であり、各地で試みられている。我々は、学生に命令セットを開放し、自由に命令セットを設計させる一方で制限された資源と時間内に実験を終えるための枠組みとしてPICOを提案し、実験用ボードを実装し、実際に学生実験に用いてきた。本稿では、さらに進んだ計算機アーキテクチャ教育を行なうための、パイプライン化された教育用マイクロプロセッサPICO<sup>2</sup>とこれを実装するための新しい実験用ボードを開発した。また、実際に学生実験に用いた例を報告する。

キーワード、教育用マイクロプロセッサ、パイプライン処理、FPGA、アーキテクチャ教育

## PICO<sup>2</sup> — The Framework for Education of Pipelined Architecture

Katsunobu Nishimura      Kazumasa Nukata      Hideharu Amano

Dept. of Computer Science, Keio University

Abstract Design and implementation of a model microprocessor in student experimental lab. are important for computer architecture education. A lot of educational microprocessors and experimental kits have been proposed and developed. We have also proposed and tried an educational microprocessor framework PICO. Here, we develop a framework for education of recent pipelined architecture called PICO<sup>2</sup>, and a new experimental kit. Experimental results using this framework are also reported.

key words Educational microprocessor, Pipelined architecture, FPGA, Architecture education

## 1 はじめに

大学における計算機アーキテクチャの教育において、教育用のマイクロプロセッサの設計演習、実験は、学生に広く実的な知識と経験を与えると共に、創造力を刺激し、計算機に対する興味を高める点で非常に有効である [1]。従来、教育用マイクロプロセッサは、ペーパマシンとして実現され、シミュレータ上でのみ動作するものが多かった。しかし近年、設計用ツールの普及、書き換え可能なゲートアレイ (FPGA) の集積度の向上などにより、実際に教育用プロセッサを製作する実験についての報告が活発になりつつある。これらの例として九州大学における KUE-CHIP2[2]、QP-DLX[3]、九州工業大学の KITE[4]、KITE2[5]、DLX-FPGA[6]、広島市立大学の安佐 [7]、City-1[8, 9]、東大理学部情報工学科の実験 [10]、明治大学のスタックマシン [11] などが挙げられる。

我々も早い時期から教育用マイクロプロセッサを用いた演習、実験の方法について試行を重ねてきた。最初に、データパスのみを指定し命令セット等をオープンにした教育用マイクロプロセッサ PICO による学生実験を確立した [12, 13]。この実験では、学生は与えられたデータパスに基づき自由な命令セットを持ったプロセッサをハードウェア記述言語 PARTHENON/SFL[14, 15] を用いて設計する。そして、シミュレータ上で動作を確認した後、論理合成/配置配線を行ない、書き換え可能なゲートアレイ (FPGA) 上で実際のハードウェア上でその動作を確認する。

この試みは、一応計算機アーキテクチャの基本を理解した学生に対して、学生実験の枠組の中で行なうものである。しかも PARTHENON/SFL は、教科書の上でしか存在しないモデルコンピュータをシミュレータ上で簡単に実現することができる。これは、アーキテクチャの初等教育における講義や演習に利用することにより、学生の学習意欲を増大し、学習効果を高めることのできる可能性を持っている。このことを利用し、計算機アーキテクチャの授業の最初に PARTHENON/SFL を導入し、モデル計算機を設計しながら計算機アーキテクチャを理解していく演習つき授業に関して、検討を行ない試行した [16, 17]。この結果、様々な問題点が明らかになったものの、アーキテクチャ教育の初等段階に関しての教育システムをひとつと構築することができた [18]。

ここまでの段階で我々の試みは、計算機アーキテクチャの基礎を学ぶためのもので、設計するマイクロプロセッサは、古典的な 3パス方式の RISC プロセッサであった。しかし、アーキテクチャ教育の次の段階では、マイクロプロセッサのパイプライン処理に関する知識を修得する必要があると思われる。そこで、これまでに開発してきた教育用 16bit マイクロプロセッサのパイプライン化について検討を行なった [19, 20]。

表 1: カリキュラム例

授業名	計算機構成
時期	2 年次後期 (12 週)
内容	4bit Adder, 4bit ALU 4bit アキュムレータマシン 16bit RISC プロセッサ (PICO-16)
授業名	コンピュータアーキテクチャ
時期	3 年次前期 (12 週)
内容	割り込み機能 (PICO-16) キャッシュ (PICO-16) パイプライン化 (PICO <sup>2</sup> )
授業名	情報工学実験第二
時期	3 年次後期 (3 週)
内容	PICO <sup>2</sup> を FPGA に実装
授業名	VLSI デザイン
時期	4 年次前期 (12 週)、4 年次後期 (12 週)
内容	Verilog-HDL によるプロセッサの設計

本稿では、このパイプライン化された教育用マイクロプロセッサ PICO<sup>2</sup> を FPGA 上に実装するための、実験用ボードを製作した。また、実際に学部 3 年生に対する学生実験で使用した経験について報告する。

## 2 計算機アーキテクチャ教育

### 2.1 カリキュラム

現在、慶應義塾大学理工学部情報工学科において、表 1 に示したカリキュラムを用いて、学部学生に対する計算機アーキテクチャ教育を行なっている。このカリキュラムの特徴は、計算機アーキテクチャを学ぶために教科書上の架空の計算機を用いるのではなく、実際にハードウェア記述言語により記述された計算機をシミュレータ上で動かして学ぶ点にある。しかも、学習の初期の段階からこの方法を導入することによって、学生が計算機の動作を理解するのを助けることができる。

具体的には、2 年次後期に開講される授業において、各週の前半で動作原理等を説明し、後半は学生が実際にそれをシミュレータ上で動作させる。また各週とも簡単な課題を出し、学生にはこれを達成させその結果を提出させる。内容は、4ビットの加算器から始まり、ALU、アキュムレータマシンと徐々にステップアップしてゆき、最終的には 16ビットの簡単な RISC 型マイクロプロセッサ PICO-16 の動作確認まで行なう。授業の初めの頃は、ハードウェア記述言語で記述されたハードウェアを動作させ、理解させることに主眼をおき、後半に移るにしたがって積極的に学

生に記述させる。また、同時にマイクロプロセッサによるプログラミング等も行なう。

同様に、学年が進むにしたがって、割り込みやキャッシュ、プロセッサのパイプライン化などのより高度な計算機アーキテクチャ教育が行なわれる。さらに3年次で開講される実験において、学生が実際に記述したパイプラインプロセッサ PICO<sup>2</sup> を FPGA 上に実装し、実際のハードウェア上で動作確認を行なうことができる。

## 2.2 ハードウェア記述言語 SFL

このカリキュラムでは、ハードウェア記述言語として NTT で開発された PARTHENON/SFL[14, 15] を用いる。SFL は、C 言語に似た簡潔で理解しやすい文法を持ち、C 言語を既に学習している学生にとって、習得に対する抵抗が少ない。また、テキストベースの対話型シミュレータ SECONDS が用意されており、SFL で記述したものをその場でデバッグ、動作確認することができる。さらに論理合成ツールも用意されており、ハードウェアの記述から合成までのすべてを、統一された環境で行なうことができる。

これらの環境は、基本的に大学等教育機関で自由に用いることができるようになっており、しかも軽量で移植が容易であるため、100 人を越える学生の演習にも充分対応が可能である。PARTHENON/SFL の問題は現場で実際に用いられることが少ない点にあるが、これを補うため、慶應義塾大学情報工学科では4年生の授業で比較的人数(30-40人)のクラスで VLSI 設計論の授業を用意しており、ここで Verilog-HDL/Cadence のシミュレータ/Synopsys の環境の実習を行う。この授業の経験によると、一度 SFL を習得した学生の Verilog-HDL への移行はきわめて容易である。

## 2.3 教育用マイクロプロセッサ PICO-16

前述のカリキュラムで用いられている教育用マイクロプロセッサ PICO[12, 13] は、4ビットから16ビットまでの様々なサイズをとることのできるマイクロプロセッサのフレームワークのことである。学生はその意欲と時間に応じて、命令セットを自由に付加して自分達自身のプロセッサを構築していくことができる。

実験教育用であるため、命令セットは様々なバラエティを許すが、基本方針は以下の通りである。

- 単一命令長、Load/Store アーキテクチャとする
- レジスタ数は可能な限り多く取る
- 命令長の制約によりオペランドは2つまで
- 命令は即値をとる命令群とそれ以外から構成される
- 即値をとる命令は命令長の半分を即値領域に用いる
- データバスは予め決められたものを用いる

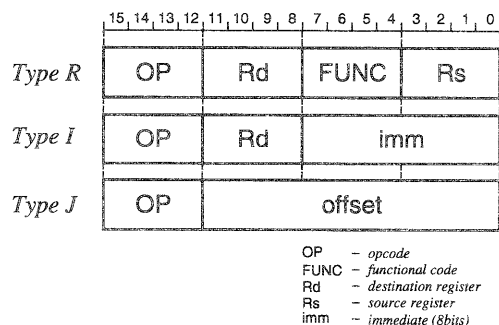


図 1: 命令フォーマット例

上記の指針に沿って設計した16ビット版 PICO である、PICO-16 の命令フォーマット例および命令セット例を図1および図2に示す。

PICO-16 の命令セットには、ロード/ストア、算術論理演算、比較、制御の4つのタイプがある。すべての命令は16ビット長で、4ビットの命令操作コードを有している。命令形式は、I (Immediate) 型命令、R (Register-register) 型命令、J (Jump) 型命令の3種類である。

I型命令は、ロード/ストア、即値演算などに使用する。R型命令は、算術論理演算などに使用し、命令操作コードの他に4ビットのファンクションコードを有している。J型命令は、分岐命令および条件分岐命令に使用する。命令長の制限によりレジスタ数はR0からR15までの16とした。またR0は、値が常に0となる。

2オペランド命令を用いていることを除き、ほぼMIPS系のアーキテクチャ[21, 22]に準じた命令セットとなっている。したがって、これを教材として用いている場合は、理解が容易になる。

## 2.4 PICO-16における問題点

1994年度より、PICO-16およびこれを実装するための実験用ボードを用いて、慶應義塾大学ならびに東京工科大学の学部3年生を対象に、PICO-16を用いた学生実験を行ってきた。その結果、いくつかの問題点が明らかになった。

**パイプライン化の必要性** 今までのPICO-16は古典的な3バス方式のRISCプロセッサであり、現在の計算機アーキテクチャを理解する上で基本となるパイプラインによる動作を考慮していない。最近では計算機アーキテクチャの教育は、古典的な3バス方式のデータバスを深くは教えずに、早い段階でパイプライン方式を導入する方法が確立しつつある[21]。このため学部3年の実験においてもパイプライン方式のマイクロプロセッサを設計することが望ましい。

IR<15:12>	xx00	xx01	xx10	xx11
00xx	ADDI	SUBI	LHI	
01xx	ANDI	ORI	XORI	
10xx	J	JAL	BEQZ	func 1
11xx	JR	JALR	BNEZ	func 2

IR<7:4>	xx00	xx01	xx10	xx11
00xx	SRLI	SLLI	SRAI	
01xx	RORI	ROLI		
10xx	SRL	SLL	SRA	
11xx	ROR	ROL		

IR<7:4>	xx00	xx01	xx10	xx11
00xx	ADD	SUB		LD
01xx	AND	OR	XOR	ST
10xx	SGT	SLT	SEQ	MOV
11xx	SLE	SGE	SNE	HALT

図 2: PICO の命令セット例

**FPGA 容量の不足** 実験用ボードは、設計当時としてはかなり大きめの FPGA を用いた (XC4010:10000 ゲート相当) が、現状の PICO-16 でも、すべての命令を実装すると使用率が 90% を越えてしまい、配置配線に多大な時間を要してしまう。また学生の中には、マルチプレクサやレジスタを意識せずに記述することにより、回路規模が大きくなり過ぎ FPGA に入りきらなくなってしまうことがあった。このため、せっかく記述した命令を削除することになり、学生のやる気を大いに削いでしまった。さらにこのボードでは、メモリとのパスが 1 セットしか用意されていないため、パイプライン構成の PICO を搭載することは困難であった。

**観測機能の問題** シミュレータ上ではマイクロプロセッサの内部が詳細に観測できるのに対し、実験用ボードではその機能が存在せず、FPGA から直接出力されるバスおよびプログラムカウンタの状態を表示する LED が付加されているだけであった。そのため、マイクロプロセッサの内部状態を観測し、その上で動くプログラムのデバッグを行なうようなことが難しかった。ともすれば、学生にとってはランダムに点灯する飾りにすぎなかったかもしれない。

**入出力装置の問題** 実験用ボード上には、入出力用として 8 ビット分のスイッチと LED しか用意されていないかった。このため、実装した PICO が物理的に I/O を制御を行う能力が不足し、結果として動作させるプログラムはソーティングや素数の計算など、メモリ上のデータに対する操作が主となった。この場合、学生はモニタを介してメモリの内容を確認するだけなので、シミュレーションとの区別がつきにくく、自分の設計したマイクロプロセッサを実際にハードウェア上で動作させたという実感を得ることが難しかった。

以上の問題点を改善するため、我々は、PICO のパイプライン化および新しい実験用ボードの作成および評価を行なった。本稿ではこれらについて報告を行う。

## 3 PICO のパイプライン化

### 3.1 パイプラインステージ

PICO-16 の設計方針および命令セットはそのまま、パイプライン化した教育用マイクロプロセッサ PICO<sup>2</sup> — PICO with Pipelined Control を図 3 に示す。

PICO<sup>2</sup> のパイプラインの動作は、文献 [23] に書かれているパイプライン化手法とほぼ同じになるが、PICO<sup>2</sup> ではメモリアクセスにディスプレイメントを用いないため、MEM ステージと EX ステージをひとつにまとめることができる。その結果、以下の 4 ステージで動作する。

#### IF — Instruction Fetch ステージ

命令を *Instruction Memory* からフェッチし、IR (*Instruction Register*) に格納する。

#### ID — Instruction Decode ステージ

命令のデコードを行なう。また、レジスタの値を読み出すと共に、PC をインクリメントする。分岐命令に関してはここで、飛び先を計算する。

#### EX — EXecution ステージ

ALU を用いる演算を行なう。また、ロード/ストア命令に関しては、このステージで *Data Memory* に対する読み書きを行なう。

#### WB — Write Back ステージ

計算結果をレジスタに格納する。

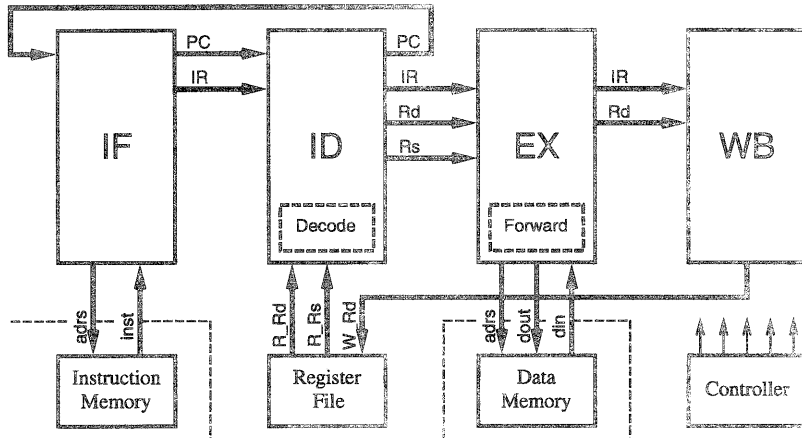


図 3: PICO<sup>2</sup> のブロック図

構造ハザードを回避するため、命令を格納するための *Instruction Memory* とデータを格納するための *Data Memory* は、それぞれの個別に用意する必要がある。

### 3.2 PICO<sup>2</sup> 実験用ボード

前節に述べたように、従来の実験用ボードは FPGA の容量不足のために PICO<sup>2</sup> に対応することができない。そこで、さらに大容量の FPGA を導入すると共に、従来のボードの使用経験に基づき改良を加え、新たな実験用ボードを製作した。この実験用ボードの概略を図 4 に、その外観を図 5 に示す。

ボード上には前回のもと同様に、FPGA (PICO<sup>2</sup>) のコンフィグレーションおよびメモリへの読み書きのために、制御用 CPU (モトローラ社 MCF5204) が搭載されている。命令格納用メモリ (*Instruction Memory*) およびデータ格納用メモリ (*Data Memory*) は、PICO<sup>2</sup> および制御用 CPU のどちらからでも自由に読み書きできるよう工夫されている。

また、ボード上には簡単なキーボードと液晶表示器が搭載されている。これらは制御用 CPU により管理され、制御用プログラムやターゲットとなるプロセッサ PICO<sup>2</sup> の制御や状態の表示などを行なう。

#### 3.2.1 観測機能

教育用マイクロプロセッサの実験システムで重要な項目のひとつは、プロセッサの内部状態の観測機能である。外部から内部状態が参照できない場合、実際にどのような処理が行なわれているのかわからず、結局のところシミュレー

タに頼ってしまうことになる。ここではシミュレータ上で実現されるペーパーマシンと変わらず、実際にハードウェア上で動作させる意味が無くなくなってしまふ。

そこで今回設計した PICO<sup>2</sup> では、プロセッサの内部状態をくまなく観察できるようにした。このため PICO<sup>2</sup> は、独立したプロセッサとして動作するだけでなく、制御用 CPU からは一般の周辺 LSI と同じようにみえるよう工夫されている。

この機能を実現するため、PICO<sup>2</sup> 側に特別な制御信号線を設け、外部からプロセッサ内にある任意のパイプラインレジスタへの読み書きができるようにした。ただしハードウェア言語によるこれらの記述は、かなり技術を要する。したがって、学生にはあらかじめこれらを記述した枠組みを与えることにより、実験をスムーズに行なう工夫が必要となる。また、新しく製作した実験用ボードでは、制御用 CPU が PICO<sup>2</sup> から読み出した内部状態を液晶表示器に表示することにより、高度な観測機能を実現している。

#### 3.2.2 入出力装置

前節に述べた従来のボードの問題点を改善するため、今回のボードには入出力装置として、簡単なキーボードと液晶表示器を備えている。これらは制御用 CPU で用いられるとともに、PICO<sup>2</sup> のアドレス空間上に仮想的にマッピングされ、PICO<sup>2</sup> からはメモリの特番地に読み書きを行うことによりアクセスできる。これにより、キーボードから値を取り込み、液晶に表示する実験が可能になり、実験のバラエティを広げることができる。

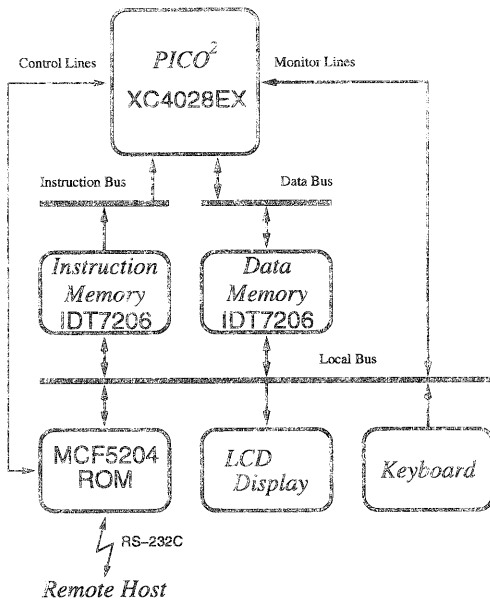


図 4: PICO<sup>2</sup> 実験用ボードの概略

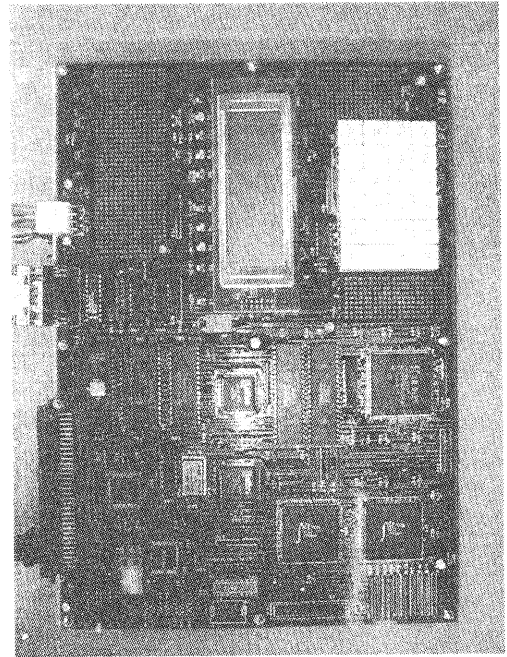


図 5: PICO<sup>2</sup> 実験用ボードの外観

### 3.2.3 コンフィギュレーションデータの転送

ターゲットプロセッサ (PICO<sup>2</sup>) となる FPGA へのコンフィギュレーションデータの転送は、前回のボードと同様にホストコンピュータから RS232C を通して行なうものとした。学生は、あらかじめ用意したコンフィギュレーションデータをホストコンピュータ上のターミナルソフトから転送することによって、FPGA のコンフィギュレーションを行なうことができる。

他に、コンフィギュレーションデータを ROM に書き込んでボード上に載せる方法などが考えられる。この方式のほうが、よりハードウェアを意識できると思われるが、今回は実験の容易性などを考え採用しなかった。

### 3.2.4 PICO<sup>2</sup> の実行

PICO<sup>2</sup> を実行するためのプログラムコードは、コンフィギュレーションデータの転送と同様に RS232C を通して書き込む方法と、ボード上のキーボードから入力する方法が用意されている。また、PICO<sup>2</sup> へのクロックの供給は、RS232C からシミュレータ (seconds) と同様のコマンド (forward) で行なう方法と、キーボードから直接入力する方法が用意されている。

## 3.3 ハードウェア量の評価

文献 [7] 同様、PICO でもステップアップ方式でパイプラインプロセッサを構築していくことを想定している。そこで以下の段階を想定して、それぞれのハードウェア量の評価を行った。

**step1** ADD, LHI, LD, ST, J 命令等の簡単な命令のみを実装したパイプライン化された PICO を学生に提示する。このパイプラインは、フォワードリング、ストール機能を持たない基本的なものである。学生はこのベース (PICO<sup>2</sup>/s1) をシミュレーションすることによりパイプラインの動作の概略を学習する。

**step2.1** 上記の PICO<sup>2</sup>/s1 に命令を付け加えた PICO<sup>2</sup>/s2 を設計する。付け加える命令は、演習用の課題 (例えばソーティング、加算の繰り返しによるかけ算等) が達成できるものを揃えればよく、学生が自分で好きなように考える。ここで、最も問題となるのは分岐命令で、これについてはヒントを与える必要がある。初期段階では ALU により飛び先を計算させてもよいし、飛び先計算専用加算器を持たせても良い。また、分岐は遅延分岐を原則とし、ストールはさせない。

**step2.2** 上記の PICO<sup>2</sup>/s2 をシミュレーションし、課題の

表 2: PICO<sup>2</sup> のゲート換算数

Model	gates	CLBs	FFs	Delay
PICO <sup>2</sup> /s1	8411	350	171	43.051
PICO <sup>2</sup> /s2	9797	607	171	49.779
PICO <sup>2</sup> /s3	9997	622	171	63.156
PICO-16	4974	360	67	41.490

表 3: 各ステージのゲート換算数

Module	gates	CLBs	FFs	Delay
IF	619	40	32	32.326
ID	1662	113	76	29.115
EX	2351	345	24	40.352
WB	2	1	0	3.662
REGFILE	4442	55	0	42.959
others	921	68	39	—
total	9997	622	171	63.156

プログラムを実行し、フォワーディングの問題、遅延スロットの問題等を認識する。プログラムが正しく動作するように NOP 命令を挿入する。

**step2.3** PICO<sup>2</sup>/s2 を論理合成し、ゲート数、最大動作周波数を評価し、パイプライン化しない PICO と比較する。さらに、FPGA 上に実装し、実際に動作を確認する。

**step3** PICO<sup>2</sup>/s2 にフォワーディング機能等の高度な機能を付け加えた PICO<sup>2</sup>/s3 を設計する。

基本となる命令を実装した PICO<sup>2</sup>/s1、拡張を行なった PICO<sup>2</sup>/s2、フォワーディングを行なった PICO<sup>2</sup>/s3 の、ゲート換算数、CLB 数、FF 数、および最大遅延 (ns) を表 2 に示す。参考までに、パイプライン化する前のプロセッサ (PICO-16) の値も示しておく。なお、論理合成には UNIX 版 PARTHENON 2.3.0.6 を、配置配線には XILINX 社 Alliance シリーズ 1.5 を使用した。また、FPGA には XILINX 社 XC4028EX を想定した。

また同様に、各パイプラインステージにおけるゲート換算数等を表 3 に示す。このとき、REGFILE (Register File) の CLB 数が他に比べて小さくなっているのは、FPGA 内のプリミティブ (RAM) を用いたためである。

## 4 学生実験における実施例

今年度後期より、慶應義塾大学理工学部情報工学科 3 年生の情報工学実験第二において、PICO<sup>2</sup> を用いた実験を行なっている。対象となる学生は、既に授業において、

PARTHENON/SFL の使い方からパイプライン化された PICO<sup>2</sup> までを一通り学習している [18]。

情報工学科では様々なテーマの実験を行う関係上、実験時間は 1 回 3 時限の枠が 3 回の制約があり、全てを 1 から設計するには不足した。そこで、学生に対しては、LD, ST, BEQZ および ALU 命令などを実装したパイプライン構造を持つ PICO<sup>2</sup> の SFL 記述を与え、これに以下の問題を解くプログラムを考えさせ、必要な命令を追加させた。なお、与えられた記述には、フォワーディング機能などのハザード回避機能はいっさい含まれていない。実装に必要な命令は、学生に自由に選択させた。

- メモリ上に置かれたデータを昇順に並び替え、最大値を液晶表示器に表示せよ
- 与えられた 2 つの数の最大公約数を計算し、結果を液晶表示器に表示せよ
- エラトステネスのふるいを用いて、与えられた数の中で最大の素数を見つけて、液晶表示器に表示せよ

学生は 3-4 人のグループを作り、プログラム担当、命令追加担当、フォワーディング担当等作業の分担を決めながら、共同作業を行った。多くの学生は、与えられた記述に対して、2 ないし 3 つほどの命令を追加することで、課題を解くことができた。またほとんどの学生が、レジスタのフォワーディング機能、コントロールハザードを回避するための早期分岐機能を追加することができた。1 回目と 2 回目は設計と SECONDS によるシミュレーションを行い、最終回である 3 回目に実験用ボードを用いて、SECONDS ではシミュレーション仕切れない大きなデータに対して処理を行い、実際に液晶に表示した。今までの所、ほとんどのグループで、液晶の表示の最終段階まで成功することができた。このことにより、学生に実機で動いたという実感を与えることができた。

## 5 まとめ

パイプライン化された教育用マイクロプロセッサ PICO<sup>2</sup> と、この学生実験用のボードを紹介した。現在、この実験は慶應義塾大学理工学部情報工学科の 3 年生に実施しているが、来年より東京工科大学工学部情報工学科でも実施する予定である。まだこの実験は 3 回目を実施している段階で、キーボード等の入力装置を用いておらず、液晶の利用も簡単なものにとどまっている。さらに改良し、学生の興味を喚起し、実際の動作時に感動を味合わせることができると期待している。また、今回の実験では、テーマの達成に重点を置いたため、動作周波数等の評価は行なわれない。これらの点も評価し、より質のよい CPU チップの実装という方向に結び付けることも課題のひとつである。

## 謝 辞

PICO<sup>2</sup> 実験用ボードの作成に協力して頂いた、東京工科大学工学部情報工学科の松永俊雄教授ならびに埴敏博講師に感謝致します。また、慶應義塾大学理工学部情報工学科の実験 TA の皆様、ならびに熱心に取り組んでくれた学生諸君に感謝します。

## 参考文献

- [1] 柴山, 新實. 大学における計算機アーキテクチャの教育方法に関する考察. 情処研報, Vol. 93, No. 49, pp. 27 - 34, June 1993. 93-ARC-100-4.
- [2] 越智 ほか. 計算機工学・集積回路工学教育用マイクロプロセッサ KUE-CHIP2. 信学技報, Vol. 92, No. 290, pp. 3 - 10, October 1992. CPSY92-46.
- [3] 岩井沢 ほか. 計算機工学一貫教育用マイクロプロセッサ QP-DLX の開発. 情処研報, Vol. 93, No. 49, pp. 35 - 42, June 1993. 93-ARC-100-5.
- [4] 田中, 小羽田, 久我, 末吉. 教育用マイクロプロセッサ KITE とその開発支援環境. 情処研報, Vol. 93, No. 49, pp. 59 - 66, June 1993. 93-ARC-100-8.
- [5] 末吉 ほか. FPGA を利用した教育用マイクロプロセッサ KITE2 - システムソフトウェア教育への応用 -. 情処研報, Vol. 94, No. 50, pp. 25 - 32, June 1994. 94-ARC-106-4.
- [6] 末吉, 井上, 奥村, 久我. 教育用 32 ビット RISC マイクロプロセッサ DLX-FPGA と教材ボードの開発. 第 3 回 FPGA/PLD Design Conference and Exhibit 論文集, pp. 579 - 588, July 1995.
- [7] H.Ochi. ASAVer.1: An FPGA-Based Education Board for Computer Architecture System Design. *Proc. of ASP-DAC 97*, pp. 157 - 165, January 1997.
- [8] 高橋 ほか. マイクロコンピュータ設計教育環境 City-1. 情処研報, Vol. 97, No. 17, pp. 41 - 48, February 1997. 97-DA-83-6.
- [9] 高橋, 吉田. 完全なインターロックを行なうパイプライン CISC/RISC の設計教育. 情処研報, Vol. 97, No. 103, pp. 97 - 104, October 1997. 97-DA-85-15.
- [10] 松本. プロセッサ作成学生実験-チップ、ボードからコンパイラ、アプリケーションまで. マイクロエレクトロニクス研究開発機構第 12 回ワークショッププロシーディング, pp. 1 - 12, December 1993.
- [11] 鈴木, 井口, 山田. FPGA を用いた実習用仮想マイクロプロセッサの試作. 第 3 回 FPGA/PLD Design Conference and Exhibit 論文集, pp. 609 - 617, July 1995.
- [12] 西村, 渡辺, 工藤, 天野. FPGA を用いたマイクロプロセッサ開発実験. 第 5 回バルテノン研究会 資料集, pp. 65 - 74, November 1994.
- [13] 西村, 工藤, 天野. 教育用 16 ビットマイクロプロセッサ PICO-16. 第 3 回 FPGA/PLD Design Conference and Exhibit 論文集, pp. 589 - 596, July 1995.
- [14] Nakamura, Oguri, Nagoya, and Yukishita. High-level Synthesis Design at NTT Systems Labs. *IEICE Trans. Inf & Syst.*, Vol. E76-D, No. 9, pp. 1047 - 1054, September 1993.
- [15] 中村, 小野. ULSI の効果的な設計法. オーム社, 1994. ISBN4-274-03437-2.
- [16] 天野, 西村. 計算機入門教育におけるバルテノンの利用. 第 7 回バルテノン研究会 資料集, pp. 73 - 82, November 1995.
- [17] 天野. バルテノンの計算機初等教育への利用 - 実施報告 -. 第 10 回バルテノン研究会 資料集, pp. 51 - 56, April 1997.
- [18] 天野, 西村. バルテノンのアーキテクチャ初等教育への利用報告. 第 14 回バルテノン研究会 資料集, pp. 9 - 18, April 1999.
- [19] 金森, 西村, 天野. 教育用マイクロプロセッサ・パイプライン化 PICO. 第 11 回バルテノン研究会 資料集, pp. 59 - 66, December 1997.
- [20] 西村, 額田, 天野. 教育用パイプライン処理マイクロプロセッサ PiPICO. 信学技報, Vol. 98, No. 449, pp. 23 - 30, December 1998. CPSY98-141.
- [21] J.L.Hennessy and D.A.Patterson. *Computer Architecture - A Quantitative Approach - 2nd edition*. Morgan Kaufmann Publishers, Inc., 1996.
- [22] D.A.Patterson, J.L.Hennessy, 成田光彰 訳. コンピュータの構成と設計 第 2 版 [上]. 日経 BP 社, 1999. ISBN4-8222-8056-X.
- [23] D.A.Patterson, J.L.Hennessy, 成田光彰 訳. コンピュータの構成と設計 第 2 版 [下]. 日経 BP 社, 1999. ISBN4-8222-8057-8.