

## 教育用マイクロプロセッサ KITE による 設計教育の事例報告

末吉 敏則 †‡ 田中 康一郎 † 久我 守弘 †

† 九州工業大学 情報工学部 知能情報工学科

‡ 九州工業大学 マイクロ化総合技術センター

福岡県飯塚市川津 680-4

sueyoshi@ai.kyutech.ac.jp

あらまし 教育用マイクロプロセッサ KITE は、書換え可能な FPGA を活用して学習者が自らの手で設計、実装、動作確認まで行うことができる情報工学向け教材である。本稿では、KITE マイクロプロセッサを利用した設計教育事例として、学部3年生 84 人を対象に行った部分設計によるマイクロプロセッサの開発実験と、仕様書のみを与えて自由に設計する本格的な学生実験の実施結果について報告する。前者では最終的に約 75% の学生が完成させ、学生実験の適当な課題であることが確認できた。一方、後者では許容できる期間内に全員が完成させ、回路図入力による開発のみならず、ハードウェア記述言語による開発も学生実験の課題として実施できることを確認した。

和文キーワード マイクロプロセッサ, FPGA, CAD, 論理設計, ハードウェア記述言語, 教育

## Examples of Design Education Using an Educational Microprocessor KITE

Toshinori Sueyoshi †‡ Koichiro Tanaka † Morihiro Kuga †

† Department of Artificial Intelligence  
Kyushu Institute of Technology

‡ Center for Microelectronic Systems  
Kyushu Institute of Technology

680-4 Kawazu, Iizuka-shi, Fukuoka, 820 Japan

### Abstract

Educational microprocessor KITE is a teaching material for LSI design education of students using FPGA. This paper reports on the examples of designing KITE microprocessor parts performed by 84 students in our department. Results have confirmed that 75 percent of the students have finally completed the design. Furthermore, this paper reports on the results of designing the whole KITE microprocessor by other students who were only given the specifications and allowed to perform the experiments using schematic editor or HDL. The results reveals that the students were able to complete the overall KITE microprocessor design which confirm that design education can also be performed using schematic editor or HDL.

英文 key words microprocessor, FPGA, CAD, logic design, hardware description language, education

## 1 はじめに

集積度向上による設計期間の長期化や ASIC 化の進展による開発チップ数の増加に伴い、相対的に LSI 開発工程における設計の比重が増し、CAD システムを利用した LSI 設計教育の重要性が高まりつつある。最近では、LSI 技術の進歩が次第にシステムレベルでの設計力に依存する傾向にあり、大学における LSI 設計教育の必要性が大学のみならず産業界からも言われるようになってきた。これに対し、LSI 設計教育を実際に実施するには予算的制約や時間的制約などの問題を解決する必要がある、実用規模の LSI 実装を前提とした設計教育を行っている教育機関はまだ少ない [1],[2]。

また、情報工学分野に限らないが、現在のハードウェア教育では各々の要素技術に関する実験結果の解析を行うアナリシス型の実験が主流を占めており、それらの要素技術を応用・統合化して一つのまとまった「もの」として作り上げるようなシンセシス型の実験は少ない。各々の要素技術が高度化し費用がかさむという事情もあるが、やはりハードウェア教育においてはバーチャルではなく、実際に「もの」作りの面白さや感動を体験できるような実験が必要である。

そこで、我々は LSI 設計教育と情報工学分野のシンセシス型実験を実現するために書換え可能な FPGA(Field Programmable Gate Array) に着目し、学習者が自らの手で設計から、実装、動作確認まで行える教育用マイクロプロセッサ KITE<sup>1</sup>を開発し、その開発支援環境の整備を行ってきた [3], [4], [5]。KITE はノイマン型コンピュータの動作原理 [6] から、設計理論、構成方式、管理技法まで一貫した教育を意図した簡素な 16 ビットマイクロプロセッサであり、情報工学教育や集積回路工学(論理回路設計)教育に利用できる。

本稿では、まず第 2 章では KITE マイクロプロセッサの概要について述べる。次いで第 3 章で、本年度前学期に学部 3 年生を対象に実施した、教育用マイクロプロセッサ KITE を利用した設計教育事例を紹介する。また、第 4 章では回路図エディタあるいはハードウェア記述言語を用いて、各自のアイデアを盛り込んで KITE マイクロプロセッサ全体を設計する本格的な学生実験を紹介する。そして、第 5 章では学生実験の実施結果に基づき開発期間を短縮するための改善策について述べ、最後に第 6 章で簡単なまとめを述べる。

## 2 教育用マイクロプロセッサ KITE の概要

FPGA を利用した教育用マイクロプロセッサ KITE の基本仕様について述べる [7]。

<sup>1</sup>カイトと発音する。たこ揚げの鳳のこと。

## 2.1 KITE マイクロプロセッサ

KITE マイクロプロセッサは利用予定の FPGA(Xilinx 社 LCA XC4005-PG156) のゲート数やピン数を考慮して、16 ビットマイクロプロセッサとしている。アドレス空間にはメモリ空間と I/O 空間があり、メモリ空間として 4K ワード(アドレス幅 12 ビット)、I/O 空間として 256 ワード(アドレス幅 8 ビット)を用意している。

レジスタ構成としては、短い語長や FPGA のゲート利用率を考慮してアキュムレータ方式を採用している。よって、KITE マイクロプロセッサ内には 16 ビットレジスタ 2 個(アキュムレータ、命令レジスタ)、12 ビットレジスタ 4 個(プログラムカウンタ、スタックポインタ、インデックスレジスタ、アドレスレジスタ)の計 6 個がある。その他に、演算結果の状態を保持する 4 ビットのフラグがある。

シーケンス制御方式は、布線論理(ワイヤドロジック)方式としている。これは、(1) 論理回路の設計教育が大きな目的の 1 つである、(2) 計算機のデータバス系とコントロール系の両方を設計できる、(3) マイクロプログラム制御メモリは FPGA のゲート利用率を低下させる、などの理由からである。

KITE マイクロプロセッサの構成例として、1 バス構成と 3 バス構成という代表的な内部構成を図 1 に示す。

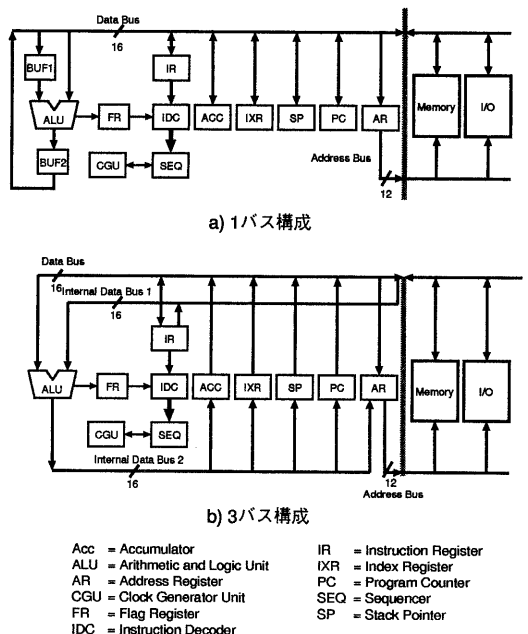


図 1: KITE マイクロプロセッサの内部構成

表 1: 命令セット

データ転送命令	算術論理演算命令	シフト命令
LD Load	ADD Add	LSL Logical Shift Left
ST Store	SUB Subtract	LSR Logical Shift Right
MV Move	INC Increment	ASL Arithmetic Shift Left
	DEC Decrement	ASR Arithmetic Shift Right
	OR Inclusive OR	ROL Rotate Left
	EOR Exclusive OR	ROR Rotate Right
	AND AND	SWP Byte Swep
	NOT NOT	
分岐命令	スタック命令	入出力命令
JP Jump always	PUSH Push down	IN Input
JPC Jump on Carry	POP Pop up	OUT Output
JPV Jump on Overflow		システム制御命令
JPZ Jump on Zero		NOP No Operation
JPS Jump on Sign		HALT Halt
CALL Call Subroutine		
RET Return from Subroutine		

## 2.2 命令セット

命令形式には 5 種類があり、すべて 16 ビット固定長である。命令セットとしては、表 1 に示すようなソフトウェアの作成に不自由とされない程度の基本的な命令を用意している。つまり、データ転送命令が 3 種類、算術論理演算命令が 8 種類、シフト命令が 7 種類、分岐命令が 7 種類、スタック命令が 2 種類、入出力命令が 2 種類、システム制御命令が 2 種類の計 31 種類である。また、アドレッシング方式は、直接、インデックス修飾、レジスタ直接、含意、即値の 5 種類がある。

## 2.3 端子機能

端子は図 2 に示すように、マイクロプロセッサの基本動作に不可欠な入出力端子と、マイクロプロセッサ内部の動作や状態の観測用端子に大別できる。

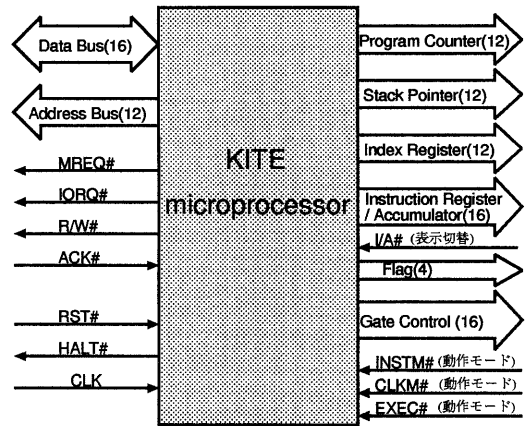
基本動作に必要な入出力端子としては、データバス (16 ビット)、アドレスバス (12 ビット) 制御信号 (7 ビット) がある。一方、観測用端子としては、レジスタ群 (プログラムカウンタ、スタックポインタ、インデックスポインタ、命令レジスタ/アキュムレータ、フラグ) の内容、各種ゲート閉閉制御信号、動作モード信号 (通常、命令単位、クロック単位) がある。このように多数の観測用端子を設けて可観測性を高めているので、学習者はマイクロプロセッサ内の状態や動作を容易に把握することができる。

## 3 学生実験 I (部分設計)

ここでは、本学情報工学部知能情報工学科の学部 3 年生 84 人を対象に実施した、教育用マイクロプロセッサ KITE を利用した設計教育を紹介する。

### 3.1 学生実験 I の概要

教育用マイクロプロセッサ KITE を利用した設計教育は現在、カリキュラムの都合で暫定的な試行期間にある。そこで、本年度 (1993 年度) は学部 3 年次の学生を対象とする前学期学生実験の 4 テーマのうち、従来の論理回



入出力端子：35ピン

観測用端子：76ピン

図 2: 入出力端子

路実験の代替テーマとして「回路図入力による KITE マイクロプロセッサの開発」を実施した。

前学期学生実験には実験テーマが 4 つあり、実験日は週あたり 2 回 (午後半日) である。このため、3 年生 84 人を 4 つのグループに分けて、1 グループ (20~24 人) ごとに 3 週間ずつ実施した。結局、1 回あたり半日の計 6 回 (3 週間) でマイクロプロセッサの開発を行うことになり、実験期間としてはかなり短い。実際の開発では、学生は基本的に 2 人一組で設計演習を遂行する。なお、2 人一組とすると総計は本来 42 組であるが、一人で設計したい旨の要望 (2 人) を受入れたので実際は総計 43 組となった。

学生実験環境としては、図 3 に示すような開発環境を用意した。人数が多かつ実験期間が短いため、パーソナルコンピュータで FPGA の自動配置配線処理まで行っているのは実験時間内に納まるほど短時間に FPGA 実装を行うことが難しい。そこで、負荷は軽いインタラクティブな操作が主体となる回路図入力やデバッグについては各組でパーソナルコンピュータを利用する一方、負荷の重い自動配置配線処理にはワークステーションを共同利用している。このようにワークステーションとパーソナルコンピュータで役割分担をすることで、本実験のような多人数教育でも円滑に行うことができ、経費削減にも役立つ。

実験期間が 3 週間 (計 6 回) とかなり短いため、回路図エディタを含む CAD システムの操作法の習得からマイクロプロセッサの動作や機能の把握までを行ってから、マイクロプロセッサ全体をすべて設計するのは難しい。そこで、学生には KITE マイクロプロセッサの仕様書に

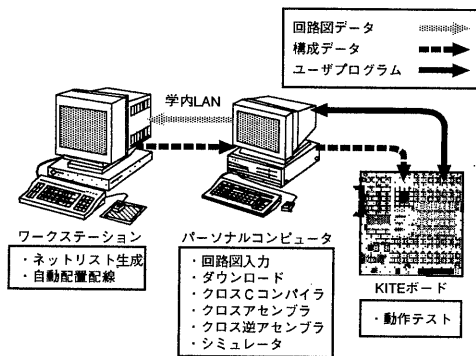


図 3: 学生実験環境の構成

加え、1バス構成マイクロプロセッサの階層設計を行う際のフレームワーク（動作確認用プリント基板の仕様に合わせてFPGAのピン割付等を定義した枠組で、階層設計における最上位図面）をライブラリとして提供し、その基本構成要素である算術論理演算回路（ALU）、命令デコーダ、ならびにシーケンサの設計を主として行わせることにした。

実験のスケジュールを図4に示す。初回ではまず、シーケンサの基本回路であるジョンソン・カウンタを題材に実験指導書に従って設計し、動作確認用プリント基板であるKITEマイクロプロセッサ・ボード[8]に実装して動作確認を行う。これにより、回路図エディタを含むFPGA開発用CADシステムの操作が一通りできるようになり、KITEマイクロプロセッサ・ボードを利用した動作確認法も習得できる。次に、2回程度の実験期間で、マイクロプロセッサのALUを設計し、動作確認を行う。最後に、残りの3回程度の実験期間で、マイクロプロセッサの制御回路（命令デコーダやシーケンサ）を設計し、先に自分達が開発したALUと予め配布している1バス構成用フレームワークを統合してマイクロプロセッサ全体を完成させ、動作確認を行う。動作確認にはROMに用意されている簡単なテストプログラムを利用できるが、図3に示すクロスソフトウェア環境[9]を利用できるのでアセンブラ言語やC言語でプログラムを記述して詳細な動作確認も行える。

### 3.2 学生実験 I の結果

学生実験の実施結果を表2に示す。実験期間が短いにも拘らず期間内にマイクロプロセッサを完成させた学生は23組で、全体の54%にあたる。また、マイクロプロセッサの設計は完了しているが、デバッグ中であるのは13組（全体の30%）である。デバッグ中のほとんどはマイクロプロセッサが全く動作しないわけではなく、数種類の命令が仕様と異なる動作をする程度である。なお、

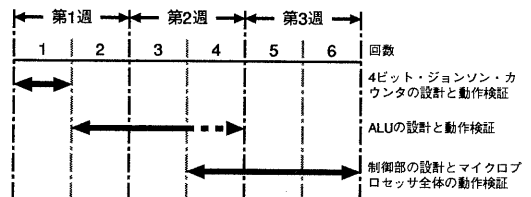


図 4: 設計演習のスケジュール

実験期間内に完成まで至らなかった学生は、実験終了後も空き時間（主として夏休み）を利用して引続きマイクロプロセッサの開発に挑戦し、最終的には32組（全体の約75%）の学生がマイクロプロセッサを完成させた。なお、表2のDグループの場合、夏休み後に実験期間が終了して直ちに学期末試験期間になるため、最終完成率が他グループよりも若干低下している。

また、本実験のように学生が設計した論理回路を実際のデバイス（FPGA）に実装する場合、学生が設計したマイクロプロセッサの回路規模やデバイスへの実装時間が重要となる。完成に至ったマイクロプロセッサすべての実装結果を表3に、それらの回路規模をゲートアレイ換算ゲート数で集計したグラフを図5に示す。学生が設計したマイクロプロセッサの回路規模は、図5から分かるように4,000ゲート以上の6%を除くと、我々が先に開発した設計サンプル[5]と比べて同等か多少大きい程度のもので多く、サンプルよりむしろ小規模にできたものも少なくない。一方、実装時間に関しては、専用CADツールをワークステーション（Sun microsystems社 SPARC Station 10 Model41 や SPARC Station 2）を使用することで、平均8分弱（最大20分程度）で実装できており、学生実験において論理回路設計に集中できる開発環境を用意することができた。

### 4 学生実験 II（全設計）

来年度（1994年度）から新カリキュラムの学生実験が始まるのを契機に、各自のアイデアを盛り込んでKITEマイクロプロセッサ全体を設計する本格的な実験を導入する予定である。そこで今回、KITEマイクロプロセッサの仕様書のみを与えて、内部構成を自由に設計した場合の開発結果について調べる学生実験を試行的に実施した。開発期間には特に制限を設けず、CADツールの操作法修得から各自のマイクロプロセッサが完成するまで挑戦してもらい、実装に要するゲート数（回路規模）、実装時間、開発期間を調べた。また、設計入力手法としては現在採用している回路図エディタに加え、ハードウェア記述言語（VHDL[10]）による方法も採用して、2種類の学生実験を実施した。

表 2: 学生実験 I の実施結果

グループ	プロセッサ 期間内完成	プロセッサ デバッグ中	シーケンサ 製作中	ALU 製作中	プロセッサ 完成(最終)
A	7 (70%)	1 (10%)	2 (20%)	0 (0%)	8 (80%)
B	5 (42%)	6 (50%)	1 (8%)	0 (0%)	10 (83%)
C	6 (60%)	2 (20%)	2 (20%)	0 (0%)	8 (80%)
D	5 (46%)	4 (36%)	2 (18%)	0 (0%)	6 (54%)
全体	23 (54%)	13 (30%)	7 (16%)	0 (0%)	32 (75%)

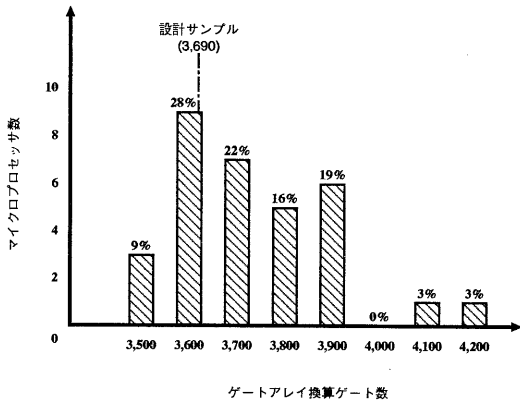


図 5: 回路規模一覧

#### 4.1 学生実験 II (回路図) の結果

回路図入力による KITE マイクロプロセッサの開発には 9 人の学生が挑戦し、全員が完成させた。実装結果を表 4 に、開発期間を表 5 に示す。

バス構成にも 3 種類あるように、各自の工夫が凝らされた KITE マイクロプロセッサが開発され、いずれも現在使用している実装デバイスの許容範囲内で実装を終えることができた。同じバス構成でもシーケンサ、ALU などの設計の違いにより CPI (Clock cycle Per Instruction) や回路規模に差が生じているのが分かる。

開発期間については、自動配置配線に利用できるワークステーションが 2 台であったため負荷が集中し、その待ち時間が開発期間に影響を及ぼしたが、平均開発日数は 25 日程度、平均開発時間は 230 時間程度であった。最も短期間 (19 日) で完成させた学生 C の場合、構成は最も基本的な 1 バス構成であり、CPI の値も設計サンプルと同じ標準的なものであった。同じ 1 バス構成でも CPI の値が小さい学生 A や B の場合、高速化の工夫を行っている結果として開発時間が幾分長くなっている。一方、最も長い開発日数 (30 日) を費やした学生 H の場合、ゲート数が最も少なく、かつ最も高速であり、工夫した跡が窺える。

この実験結果から、実際の学生実験では 2 人一組で開発を行うため、設計および回路図入力の作業分担により開発期間の短縮が見込め、自動配置配線処理用ワークステーションの増強により来年度からの設計教育を実施できることが確認できた。

#### 4.2 学生実験 II (VHDL) の結果

過去に VHDL を使用したことのない学生 5 人がそれぞれ KITE マイクロプロセッサを設計、実装した。ただし、KITE の仕様を完全に満たすものは、論理合成ツールを利用すると現在の実装デバイス XC4005 には実装できないため [5]、演算命令を加算、インクリメント、デクリメントに限定して設計した。実装結果を表 6 に、開発期間を表 7 に示す。

実装結果において回路規模に差が出たのはシーケンサとデコーダで、学生 V と Y はこれらの回路が小規模なために全体の回路規模も小さくなっている。また、開発期間には VHDL を学習する時間も含まれているが、平均開発日数は 20 日程度、平均開発時間は 152 時間程度であった。最も短期間 (17 日) で完成させた学生 V の場合、回路図の場合と同様に、構成は最も基本的な 1 バス構成であり、CPI も標準的な値となっている。一方、最も長い開発日数 (25 日) を費やした学生 Z の場合、最も高速となっており、やはり工夫した跡が窺える。

この実験結果から、ハードウェア記述言語の使用経験の無い初心者でも、プログラミングや OS の操作法についての素養があれば、VHDL を用いて比較的短期間にマイクロプロセッサを開発できることが分かった。VHDL による設計では回路のふるまい (機能レベル) を記述できるので、マイクロプロセッサの動作原理に重きをおく教育の場合には短期間で高い学習効果を期待でき、計算機工学分野の学生実験に適していると思慮される。

#### 5 開発期間短縮のための改善策

書換え可能な FPGA を利用する KITE マイクロプロセッサの大きな利点の一つは、プロセッサ全体を必ずしも一度に実装する必要はなく、階層設計の過程で構成要素ごとに実装し動作確認を行えることである。

しかし、前述の学生実験の実施結果から、この利点を生かしてマイクロプロセッサを開発する際のデバッグに

表 3: 学生実験 I の実装結果

A グループ

学生	a01	a02	a03	a04	a05	a06	a07	a08
<b>Input XNF Design Statistics</b>								
Number of Logic Symbols	796	832	895	815	764	884	840	773
Number of Flip Flops	133	134	133	134	133	133	133	134
Number of 3-State Buffers	116	116	116	116	116	116	116	116
Equivalent "Gate Array" Gates	3612	3929	3783	3776	3578	3732	3818	3765
<b>Partitioned Design Utilization</b>								
Occupied CLBs [Utility (%) ]	100	98	98	100	95	99	100	98
Packed CLBs [Utility (%) ]	83	83	84	86	76	87	86	83
Flip Flops [Utility (%) ]	21	21	21	21	21	21	21	21
3-State Buffers [Utility (%) ]	23	23	23	23	23	23	23	23
<b>Routing Summary</b>								
Number of Total lines used	4350	4134	3507	3781	3628	4368	4171	3661
<b>Cpu Times</b>								
Total (min:sec)	09:56	07:55	04:00	11:52	04:42	20:30	08:37	04:22

B グループ

学生	b01	b02	b03	b04	b05	b06	b07	b08	b09	b10
<b>Input XNF Design Statistics</b>										
Number of Logic Symbols	821	772	870	822	803	896	875	800	864	1021
Number of Flip Flops	133	134	133	138	138	133	138	133	134	134
Number of 3-State Buffers	116	116	116	116	116	116	116	116	116	116
Equivalent "Gate Array" Gates	3628	3669	3907	3807	3639	3866	3933	3655	3804	4206
<b>Partitioned Design Utilization</b>										
Occupied CLBs [Utility (%) ]	95	95	97	100	95	95	93	92	97	94
Packed CLBs [Utility (%) ]	82	76	81	78	77	88	78	80	85	79
Flip Flops [Utility (%) ]	21	21	21	22	22	21	22	21	21	21
3-State Buffers [Utility (%) ]	23	23	23	23	23	23	23	23	23	23
<b>Routing Summary</b>										
Number of Total lines used	3374	4097	3721	4202	3173	3260	3166	3294	3290	3363
<b>Cpu Times</b>										
Total (min:sec)	06:13	09:34	04:50	15:25	06:38	06:50	09:51	03:52	06:40	06:33

C グループ

学生	c01	c02	c03	c04	c05	c06	c07	c08
<b>Input XNF Design Statistics</b>								
Number of Logic Symbols	777	886	1045	769	775	879	822	900
Number of Flip Flops	133	133	133	133	133	133	133	133
Number of 3-State Buffers	116	116	116	116	116	116	116	116
Equivalent "Gate Array" Gates	3750	3974	4143	3542	3699	3784	3907	3914
<b>Partitioned Design Utilization</b>								
Occupied CLBs [Utility (%) ]	95	96	96	90	96	100	97	95
Packed CLBs [Utility (%) ]	80	87	80	72	79	85	83	84
Flip Flops [Utility (%) ]	21	21	21	21	21	21	21	21
3-State Buffers [Utility (%) ]	23	23	23	23	23	23	23	23
<b>Routing Summary</b>								
Number of Total lines used	3424	3581	3614	3444	3797	4251	3906	3306
<b>Cpu Times</b>								
Total (min:sec)	06:28	07:16	05:13	06:25	08:00	14:58	08:31	03:32

D グループ

学生	d01	d02	d03	d04	d05	d06
<b>Input XNF Design Statistics</b>						
Number of Logic Symbols	872	829	814	873	834	833
Number of Flip Flops	133	133	133	134	134	133
Number of 3-State Buffers	116	116	116	116	116	116
Equivalent "Gate Array" Gates	3784	3569	3654	3896	3688	3645
<b>Partitioned Design Utilization</b>						
Occupied CLBs [Utility (%) ]	98	94	93	97	97	97
Packed CLBs [Utility (%) ]	89	75	74	82	82	82
Flip Flops [Utility (%) ]	21	21	21	21	21	21
3-State Buffers [Utility (%) ]	23	23	23	23	23	23
<b>Routing Summary</b>						
Number of Total lines used	3972	3438	3591	3932	3915	4228
<b>Cpu Times</b>						
Total (min:sec)	05:08	03:49	07:54	06:20	13:15	10:33

表 4: 学生実験 II (回路図) の実装結果

学生	A	B	C	D	E	F	G	H	I
バス構成	1バス			2バス		3バス			
<b>Input XNF Design Statistics</b>									
Number of Logic Symbols	977	938	826	804	739	1184	733	799	953
Number of Flip Flops	135	148	134	135	118	101	102	103	101
Number of 3-State Buffers	116	116	116	116	100	100	116	100	100
Equivalent "Gate Array" Gates	4052	3962	3629	3584	3318	4271	3252	3218	3712
<b>Partitioned Design Utilization</b>									
Occupied CLBs [Utility (%)]	99	100	99	95	90	99	88	89	96
Packed CLBs [Utility (%)]	92	92	85	83	73	89	75	72	89
Flip Flops [Utility (%)]	21	24	21	21	19	16	20	16	16
3-State Buffers [Utility (%)]	23	23	23	23	19	19	23	19	19
<b>Routing Summary</b>									
Number of Total lines used	4260	4291	4029	3876	3413	3690	3326	3130	3503
<b>Cpu Times</b>									
Total (min:sec)	07:21	10:19	06:59	05:50	06:36	05:07	03:41	05:59	04:02
<b>KITE Performance</b>									
Clock cycle Per Instruction	5.51	5.86	6.16	7.16	5.12	4.84	4.89	4.84	4.84

表 5: 学生実験 II (回路図) の開発期間

学生	A	B	C	D	E	F	G	H	I
バス構成	1バス			2バス		3バス			
方式/機能設計	8 (1)	4 (1)	3 (1)	5 (1)	40 (5)	40 (5)	6 (1)	28 (4)	5 (1)
論理設計	210 (14)	87 (11)	100 (9)	82 (8)	116 (14)	64 (8)	120 (15)	112 (15)	95 (12)
算術論理演算回路	70	15	20	24	18	16	24	18	20
レジスタ・セット	60	22	10	8	10	8	16	7	5
制御回路	60	45	60	49	80	20	64	62	50
全体構成	20	5	10	1	8	20	16	25	20
デバッグ	110 (7)	23 (3)	55 (5)	142 (15)	67 (8)	50 (5)	21 (2)	50 (8)	70 (7)
算術論理演算回路	25	10	10	8	5	8	1	10	30
レジスタ・セット	70	3	5	0	2	2	0	10	10
制御回路	15	10	40	134	60	40	20	30	30
動作検証	36 (3)	110 (13)	40 (4)	24 (3)	14 (2)	60 (7)	20 (3)	20 (3)	40 (3)
合計	364 (25)	224 (28)	198 (19)	253 (27)	237 (29)	214 (25)	167 (21)	210 (30)	210 (23)

単位は時間、ただし、括弧内は日数

要する主な作業は、構成要素ごとに動作確認を行うためのデバッグ用回路の設計時間であることが明らかになった。マイクロプロセッサの開発では階層ごとに回路図を入力していくが、一つまたは複数の階層における回路図をデバッグするにはデバッグ専用の回路図が必要になる。つまり、設計した構成要素をデバッグするために新たに回路を設計しなければならなくなり、デバッグ時間の長期化につながる。また、そのデバッグ用回路の善し悪しでデバッグ効率が大きく左右され、開発期間のバラツキの原因となっていることも分かった。

そこで、我々はデバッグ効率を高め、デバッグ時間の見積りができない現状をできるだけ改善するために、汎用性のあるデバッグ用回路図(デバッグ用フレームワークと呼ぶ)を用意することにした。現在、そのようなデバッグ用フレームワークには、ALU用、レジスタ・セット用、クロック回路用、命令デコーダ/シーケンサ用などを整備している。これらは、設計する構成要素の入出力信号のみならず内部状態を観測するための信号線も定義しているため、各自の回路設計に依存する部分が必要に応じて僅かに変更するだけでデバッグ用の回路図として使用できる。デバッグ用フレームワークを利用することで、学生は前述の実験で大きな比率を占めるデバッグ

用回路図の設計時間を削減することができ、開発期間のバラツキの大きな原因となっているデバッグの効率化を期待できる。

## 6 おわりに

本稿では、教育用マイクロプロセッサ KITE を利用した設計教育事例として、本年度学部3年生84人を対象に行った部分設計によるマイクロプロセッサ開発実験と、仕様書のみを与えて自由に設計する本格的な学生実験(回路図、ハードウェア記述言語)の計3種類について実施結果を報告した。前者の学生実験では、最終的には約75%の学生が完成させ、実験期間を考慮すれば学生実験の適当な課題であることが確認できた。また、後者の学生実験では全員が許容できる期間内に完成させ、回路図入力による開発のみならず、ハードウェア記述言語による KITE マイクロプロセッサの開発も大学等の教育機関における学生実験の課題として十分に実施できることを確認できた。

これらの設計教育事例から、学生の論理回路設計や計算機システムに対する理解や学習意欲を高めるのに非常に大きい効果が認められた。自分が設計したマイクロプロセッサが苦勞の末に実際に動く「もの」として完成させ

表 6: 学生実験 II(VHDL) の実装結果

学生	V	W	X	Y	Z
バス構成	1バス			3バス	
<b>Input XNF Design Statistics</b>					
Number of Logic Symbols	759	838	881	805	977
Number of Flip Flops	125	146	148	97	132
Number of 3-State Buffers	116	152	144	124	96
Equivalent "Gate Array" Gates	3725	4057	4346	3711	4432
<b>Partitioned Design Utilization</b>					
Occupied CLBs [Utility (%)]	90	97	97	92	100
Packed CLBs [Utility (%)]	66	76	82	75	91
Flip Flops [Utility (%)]	20	23	24	15	21
3-State Buffers [Utility (%)]	23	30	28	24	19
<b>Routing Summary</b>					
Number of Total lines used	2702	3883	3661	3339	4082
<b>Cpu Times</b>					
Total (min:sec)	02:50	06:50	05:25	04:20	11:26
<b>KITE Performance</b>					
Clock cycle Per Instruction	5.88	5.88	5.78	5.04	4.92

表 7: 学生実験 II(VHDL) の開発期間

学生	V	W	X	Y	Z
バス構成	1バス			3バス	
方式/機能設計	13 (3)	12 (3)	12 (3)	18 (3)	12 (2)
論理設計	21 (4)	23 (4)	32 (5)	79 (10)	120 (13)
算術論理演算回路	3	4	2	4	12
レジスタ・セット	2	3	2	3	3
制御回路	12	12	23	52	80
全体構成	4	4	5	20	25
デバッグ	17 (3)	16 (3)	40 (7)	22 (3)	24 (4)
算術論理演算回路	2	2	2	1	2
レジスタ・セット	1	3	2	1	1
制御回路	14	11	36	20	21
動作検証	60 (7)	80 (8)	65 (8)	48 (6)	48 (6)
合計	111 (17)	131 (18)	149 (23)	167 (22)	204 (25)

単位は時間, ただし, 括弧内は日数

ると、従来のハードウェア実験では経験したことのない満足感と喜びを覚え、その後学生にも大きな自信となっている。

なお、KITE マイクロプロセッサの設計データやクロスソフトウェアはソースコードを含めてすべて公開しており、マイクロプロセッサを題材とする設計教育を実施される際の参考になれば幸いである。

#### 謝辞

実験期間が短いにも拘らず、KITE マイクロプロセッサの設計に果敢に挑戦してくれた本学情報工学部知能情報工学科の学生諸氏に感謝する。

#### 参考文献

- [1] 庄野克房: 大学における LSI 設計教育 -2 ビットマイクロコンピュータ-, 電子情報通信学会誌, Vol. 75, No.5, pp. 530 - 533, 1992.
- [2] 神原, 安浦: 計算機教育用マイクロプロセッサの開発とその応用 -集積回路技術を利用した情報工学実験-, 情報処理学会誌, Vol. 33, No.2, pp. 118 - 127, 1992.

- [3] 末吉, 田中, 船越, 松尾, 有田: 書換え可能な LSI を用いた教育用マイクロプロセッサの開発, 情報処理学会第 43 回全国大会論文集, 2Q-11, 1991.
- [4] 末吉, 田中, 柴村: 再構成可能な論理 LSI を用いた教育用マイクロプロセッサ:KITE, 情報処理学会研究報告, 92-ARC-96-15, 1992.
- [5] 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, 情報処理学会研究報告, 93-ARC-100-8, 1993.
- [6] KITE マイクロプロセッサ・システム入門 -コンピュータの動作原理- Version 1.0, 1993.
- [7] KITE マイクロプロセッサ・リファレンス・マニュアル Version 1.0, 1993.
- [8] KITE マイクロプロセッサ・ボード取扱説明書 Version 1.0, 1993.
- [9] KITE マイクロプロセッサ・クロスソフトウェア・マニュアル Version 1.0, 1993.
- [10] IEEE: *IEEE Standard VHDL Language Reference Manual*, 1988.