Vol. 35 No. 6　　情　報　処　理　　June 1994

解　説

FPGA—その現状，将来とインパクト

3. 教育へのFPGA応用例†

末吉　敏則††

1. はじめに

半導体集積技術の発達による1チップ当たりの集積度向上にともなって開発工程が長期化するとともに，多品種少量生産のASIC化の進展にともなって開発チップ数も増加してきており，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．また，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，最近では大学における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，大学でもさまざまな方法によるLSI設計教育が検討されている[1]が，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り，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[2),3)]．

一方，計算機工学教育に目を向けると，密接な関係にある論理回路，計算機アーキテクチャ，オペレーティングシステム（OS），コンパイラなどの教科目は内容の高度化・複雑化にともなって遊離した関係に陥りがちで，一貫した計算機教育の実施が難しくなってきている．計算機工学教育の場合，入門教育では学習者が計算機の内部状態を把握することにより動作原理を理解でき，専門教育においては論理回路や計算機アーキテクチャの教科目と連携して計算機の製作実験を行えることが望ましい．さらには，当該計算機を対象とするOSやコンパイラを作成させて計算機の管理・利用技術を学習させ，システムソフトウェアの実現に必要なハードウェア機構を認識させることにより，計算機システムについての効果的な教育が期待できる．

そこで本稿では上記の状況をかんがみ，最近急速に脚光を浴びているFPGA(Field Programmable Gate Array)を効果的に用いた情報処理・計算機工学教育について紹介する．具体的には，FPGAが実現した教育への応用例として，ノイマン型計算機の入門教育から，論理回路，計算機アーキテクチャ，システムソフトウェアまでの一貫した計算機工学教育を意図して開発された教育用マイクロプロセッサKITE[4)~10)]と，それを利用した設計教育事例を紹介する．教育用マイクロプロセッサKITEは，FPGAを利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教材である．ASIC時代の計算機工学教育として，LSI実装を前提とするマイクロプロセッサの設計が大学や高専などの教育機関における厳しい予算的・時間的制約の枠内で実施できることを例示する．

以下，**2.**ではFPGAが実現する教育への応用を説明する．また，**3.**では教育用マイクロプロセッサKITEの基本仕様について述べ，**4.**ではその開発工程と設計サンプルを説明する．そして，**5.**でKITEマイクロプロセッサの開発支援環境について言及し，**6.**で設計教育事例を紹介する．最後に，**7.**で簡単なまとめを述べる．

2. FPGAが実現する教育への応用

情報処理・計算機工学分野に限らないが，現在のハードウェア教育ではおのおの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．おのおのの要素技術が高度化し費用がかさむという事情

† Application of FPGA to Education for Computer Science by Toshinori SUEYOSHI (Department of Artificial Intelligence, Kyushu Institute of Technology).

†† 九州工業大学情報工学部知能情報工学科

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」作りの感動を体験できるようなシンセシス型の実験が必要である．

計算機工学分野におけるシンセシス型実験としては，たとえばLSIを利用したマイクロプロセッサの開発などが候補にあがる．マイクロプロセッサはディジタル回路の基本的要素をすべて含み，関連する計算機基礎科目も多く，さらに設計の際に各自のアイデアを盛り込むこともできるのでかっこうの題材である．しかし，マイクロプロセッサの開発を学生実験の厳しい予算的・時間的制約の枠内で実施するためには，開発支援環境としては次のような要求を満たすことが望ましい．

（1） 多数の学生が参加する実験演習をカリキュラム時間内に消化するため，マイクロプロセッサの設計を短期間で行え，さらに設計完了後は学習意欲を損なわない程度に短い期間で LSI 実装をできること．

（2） 学生が設計したマイクロプロセッサはできればすべてLSI化し，全員が手に取って動作確認をできること．

（3） 機能設計や論理設計の経験が浅い学部学生が設計するので，デバッグや改良のために何回でも設計し直せること．

米国や欧州では大学におけるシステム設計側からの LSI 設計教育が進められ，MOSIS や EUROCHIP などの試作サービスにより学生向けの LSI 設計教育環境が整備されている．日本でも大学の LSI 設計教育研究環境を整備すべく，MOSIS や EUROCHIP のような LSI 試作サービスの実現に向けて関係者の努力が払われており，ようやく環境整備が緒につき始めたところである[11),12)]．しかし，現段階では上述のいずれの要求も満たすことは難しい．

一方，使用者側でプログラム可能で，かつ構造がゲートアレイに近くて設計の自由度が高い FPGA が近年急速に高性能化し，数千から数万ゲート相当の論理回路の実現に対応可能となってきた．このため，FPGA がハードウェア教育のための強力で重宝な道具として注目を集め，いくつかの大学で教育への応用が試みられるようになってきた[13),14)]．米国でも MOSIS を利用した大学教育の援助を行っている NSF (National Science Foundation) が，集積回路システムの高度化に対応してシステムレベル設計力の強化に重点をおく人材育成に力を入れるため，FPGA を重視している[11)]．

FPGA にもいくつかの種類があるが，とりわけ教育へ大きなインパクトを与えているのは書換え可能な FPGA である．この種の代表的な FPGA としては，米 Xilinx 社の LCA (Logic Cell Array)[15)] がある．これはデバイス内部にあるコンフィグレーション用 SRAM に構成データをロードすることで任意の論理回路を実現でき，書込み回数にも制限がない．そのため，(1)設計完了後その場で短時間に LSI 化でき，(2)学習者は自分が設計したマイクロプロセッサを手に取って動作確認できる．また，(3)デバッグや改良のために何度でも設計のやり直しができるので，ディジタル回路設計の経験の浅い学部学生のような初心者の教育にも適している．つまり，FPGA を利用することで，前述の設計支援環境に対する要求をすべて満足できる．

3. 教育用マイクロプロセッサ KITE[7)]

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．

3.1 教育用マイクロプロセッサの要件

教育を目的とする KITE マイクロプロセッサの基本仕様は，以下の要件を満たすことが望ましい．

（1） マイクロプロセッサの基本アーキテクチャは，現在の FPGA の集積度を考慮してできるだけ簡素なものとし，かつ論理回路の基本的構成要素をすべて含むこと．

（2） 命令セットは，教育用として必要最小限の命令を用意すること．

（3） 特別な測定器を用いなくても，マイクロプロセッサ内部の動作や状態，すなわちデータの流れや各種レジスタの値などを外部から把握できるよう，可観測性が高いこと．

（4） 計算機の動作を逐一観測できるように，命令単位ならびにクロック単位でプログラムの実行を制御できること．

（5） システムソフトウェアの教育において実験演習による概念の定着を図れるように，メモリ空間，データ長，命令セットに配慮し，実験対象

のモデル計算機としても利用できること.

3.2 基本構成

上記要件を考慮して基本仕様の検討を行い，次のように基本構成を決定した.

最初の検討事項は，KITE マイクロプロセッサの語長である．マイクロプロセッサの語長としては，4ビット，8ビット，16ビット，32ビットなどの設計選択肢がある．前述の要件（5）におけるシステムソフトウェアの教育を考慮すると現在のメインフレームやワークステーションなどで採用されている32ビット語長のほうが望ましいが，回路規模が増大するのみならず要件（3）の可観測性を満たすのに必要な FPGA のピン数も相応に増加するため，FPGA による実装という制約下では最適ではない．一方，4ビット語長や8ビット語長のマイクロプロセッサでは回路規模が小さくなると期待できるが，要件（5）の観点からは適当な選択とは言い難い．また，4ビットあるいは8ビットマイクロプロセッサが 16 ビットマイクロプロセッサよりも語長が短いので簡素な構成になるとも一概には言えない．つまり，命令語の長さを語長と一致させれば命令語の構成は簡単となり，命令語のフェッチや解読も簡単になるため，むしろ 16 ビットマイクロプロセッサのほうが設計は容易である．以上の理由から，KITE マイクロプロセッサは利用予定の FPGA (米 Xilinx 社 LCA XC4005-PG156) のゲート数やピン数を考慮し，16ビットマイクロプロセッサとする．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）を用意する.

2番目の検討事項はレジスタ構成であり，アキュムレータ方式，汎用レジスタ方式，スタック方式という設計選択肢がある．KITE マイクロプロセッサは語長が 16 ビットと短いこと，ならびに FPGA 内に多数のレジスタを実装するとゲート利用率が著しく低下することを考慮して，レジスタ構成としてはアキュムレータ方式を採用する．よって，レジスタとしては16ビットレジスタ2個，12ビットレジスタ4個の計6個がある．具体的には，16ビットレジスタとして命令レジスタとアキュムレータがあり，12ビットレジスタとして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，アドレスレジスタがある．そのほかに，演算結果の状態を保持する4ビットのフラグがある.

最後の検討事項はシーケンス制御方式であり，布線論理方式とマイクロプログラミング方式という設計選択肢がある．以下に述べる三つの理由から，シーケンス制御方式としては布線論理方式を採用する．第一に，論理回路の設計教育が大きな目的の一つである．第二に，計算機のデータパス系とコントロール系の両方の設計を行うことができ，アーキテクチャ教育の観点からも望ましい．第三に，マイクロプログラム方式ではマイクロプログラム格納用の制御メモリを FPGA の内か外に設ける必要があり，どちらにも問題がある．つまり，現在の FPGA 内にマイクロプログラム用の制御メモリを実装するのは容量やゲート利用率の点で難しい一方，制御メモリを外部に設けるとしてもそのインタフェースに入出力端子を多数占有してしまい，**3.4** で説明する可観測性を損なうことになる.

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
分岐命令	EOR Exclusive OR	ROR Rotate Right
JP Jump always	AND AND	SWP Byte Swap
JPC Jump on Carry	NOT NOT	入出力命令
JPV Jump on Overflow		IN Input
JPZ Jump on Zero		OUT Output
JPS Jump on Sign	スタック命令	システム制御命令
CALL Call Subroutine	PUSH Push down	NOP No Operation
RET Return from Subroutine	POP Pop up	HALT Halt

3.3 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なら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．

3.4 端子機能

端子は**図-1** 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，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．

基本動作に必要となる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，インデックスレジスタ，命令レジスタ／アキュムレータ，フラグ）の内容，各種のゲート開閉制御信号，動作モード信号がある．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．

4. 開発工程と設計サンプル[5),6)]

KITE マイクロプロセッサの開発工程と，設計サンプルの実装結果と実装時間について述べる．

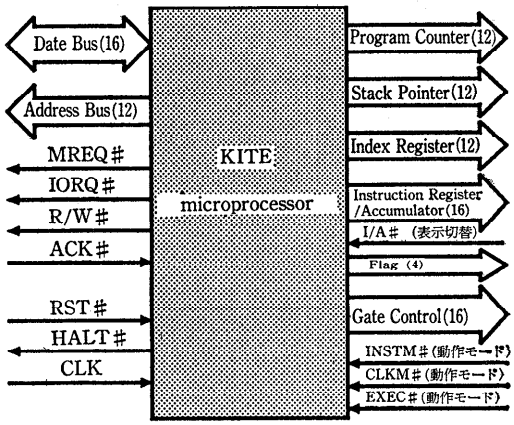
4.1 開発工程

KITE マイクロプロセッサの開発方法としては，**図-2** に示すように回路図入力による設計とハードウェア記述言語による設計を選択できる．

［1］ 回路図入力による設計

方式設計／機能設計の後，決定した設計仕様に基づき論理設計を行い，回路図エディタを用いて回路図入力を行う．回路図入力の際には豊富なマクロライブラリを利用でき，比較的容易に論理回路を入力することができる．

回路図の入力後，その図面データを XNF (Xilinx Netlist Format) と称するネットリストへ変換する．この段階で，ネット名の重複や不一致などの回路図エディタへの入力の誤りを検知することができる．この処理は複数階層（図面）でも行えるため，構成要素ごとに検証することが可能である．ネットリスト生成後，専用 CAD システムの



入出力端子：35ピン　　観測用端子：76ピン

図-1 端子機能

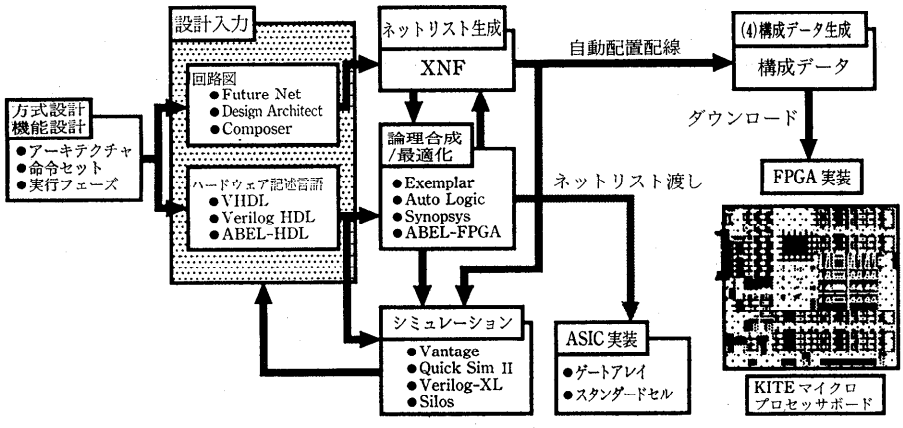


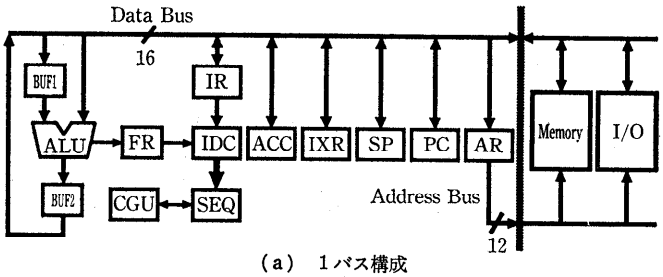
図-2 開発工程

自動配置配線機能を利用して，FPGAの構成データを生成する．この構成データをFPGAへダウンロードすることによって，FPGA内に設計者の意図した回路を実現でき，ただちに動作検証を行うことができる．

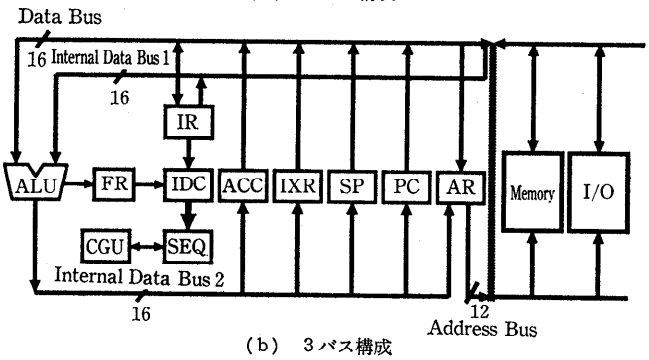
[2] ハードウェア記述言語による設計

方式設計／機能設計の後，決定した設計仕様に基づきビヘイビア（動作，機能）レベル記述あるいは構造レベル記述で設計を行う．ビヘイビアレベル記述や構造レベル記述の混在を許すので，設計の自由度が大きく設計時間の短縮に効果がある．ここではハードウェア記述言語として，VHDL[16]，Verilog HDL[17]，ABEL-HDL[18]という代表的な言語の開発環境を整えている．

ハードウェア記述言語による記述後，そのテキストデータを論理合成ツールを介してXNFに変換する．論理合成ツールを使用することで，設計者はシステム設計に集中することができる．



(a) 1バス構成



(b) 3バス構成

Acc : Accumulator
ALU : Arithmetic and Logic Unit
AR : Address Register
CGU : Clock Generator Unit
FR : Flag Register
IDC : Instruction Decoder
IR : Instruction Register
IXR : Index Register
PC : Program Counter
SEQ : Sequencer
SP : Stack Pointer

図-3 内部構成

それ以降は，回路図入力による設計と同様の工程をたどる．なお，各ハードウェア記述言語にはそれぞれ機能／論理シミュレータが用意されており，システムの動作を機能レベルや論理レベルで詳細に検証することが可能である．

上記2通りの開発方法を用意することで，実際に製作実験を行う際には目的に応じた設計教育を行うことができる．たとえば，論理回路の設計理論よりもマイクロプロセッサの動作原理に重きを

表-2 設計サンプルの実装結果

CADツール出力情報 XC-DS502-SN2 V1.41 Partition, Place, and Route Ver. 1.30	回路図入力		ABEL-HDL		VHDL		
バス構成	1バス	3バス	1バス	3バス	1バス*	3バス*	1バス
換算ゲート数	3690	3463	4118	3744	3725	3711	6519
FPGA内利用（利用率）							
使用論理ブロック数：	186 (94%)	186 (94%)	190 (96%)	186 (94%)	177 (90%)	182 (92%)	386 (96%)
理想論理ブロック数：	148 (75%)	147 (75%)	166 (84%)	154 (78%)	131 (66%)	147 (75%)	367 (91%)
フリップフロップ数：	134 (21%)	101 (16%)	135 (21%)	102 (16%)	125 (20%)	97 (15%)	131 (11%)
3ステートバッファ数：	116 (23%)	100 (19%)	116 (23%)	100 (19%)	116 (23%)	124 (24%)	124 (12%)
配線摘要 総配線数：	3987	3432	3930	3560	2702	3329	5662
CPU時間 (Sun SS10/41)	00：07：06	00：04：48	00：09：31	00：05：47	00：03：09	00：04：20	00：09：08
クロック数／命令	6.16	4.84	6.16	4.84	5.88	5.04	6.16
実装デバイス (FPGA)	XC4005-PG156						XC4010-PG191

* 演算命令を限定したサブセット版

おいた教育を行いたい場合は，回路図入力よりもハードウェア記述言語による設計のほうが，短時間で高レベルの学習効果を期待できる．それとは逆に，マイクロプロセッサの動作原理よりも論理回路の講義と連携して論理シンボルを用いた設計理論に重きをおいた教育をしたい場合は，ハードウェア記述言語よりも回路図入力による設計のほうが講義との整合性がよく回路を視覚的に把握できるため，大きな学習効果を期待できる．

4.2 設計サンプルの実装結果

KITE マイクロプロセッサを利用した設計教育の実施に先立ち，プロセッサ内部の構成方式の相違や設計手法の相違が実装結果ならびに実装時間へ及ぼす影響を事前に把握するため，内部構成として**図-3** に示す基本的な1バス構成と高速な3バス構成の2種類の設計サンプルを開発した．回路図入力ならびに2種類のハードウェア記述言語(ABEL-HDL, VHDL) による実装結果を**表-2** に示す．実装結果から，回路図入力の場合にはどちらのバス構成でも仕様に準拠した KITE マイクロプロセッサが実装できることを確認できた．

一方，ハードウェア記述言語の場合，ABEL-HDL ではシーケンサを除き基本的に構造レベル記述で設計したので，回路図入力による実装結果とほぼ同等のゲート規模で実装できることを確認できた．

しかし，VHDL ではビヘイビアレベル記述で設計を試みたため，論理合成ツールが出力する論理回路が大きくなり，完全な仕様のものはゲート規模の点で実装不可能であった．そこで，VHDL では表-2 に示すように演算命令を加算，インクリメント，デクリメントに限定したサブセット版が実装できることを確認した．なお，VHDL でも仕様に完全準拠の KITE マイクロプロセッサは 10,000 ゲート相当の論理回路を実装可能な FPGA (XC 4010-PG 191) では実装でき，設計時間の短縮や保守管理の容易さなどを考慮に入れるならば設計教育でも積極的に利用する価値がある．また，今後の論理合成／最適化ツールの性能向上により，回路図入力による実装結果と同程度になることが期待される．

4.3 設計サンプルの実装時間

FPGA を利用した設計教育を学生実験として円滑に実施できるか否かを左右する重要な要因の一つに，設計入力後の FPGA への実装時間がある．

回路図入力による開発工程において，最も時間を要する処理は自動配置配線である．2種類の設計サンプルの各種ホストコンピュータによる実装時間を**表-3** に示す．表-3 の結果から，ワークステーションによる実装時間程度ならば学生実験の時間的制約を十分に満たすことが確認できた．

ハードウェア記述言語を用いた開発工程において，時間を要する処理は論理合成／最適化と自動配置配線である．自動配置配線時間については前述の回路図入力による開発工程の場合と同様である．つまり，ワークステーションを利用した場合は表-2に示すCPU時間から分かるように，FPGA での理想論理ブロック数が 80% 程度までならばバス構成や開発方法の違いによる実装時間の差は許容できる範囲にある．一方，論理合成／最適化の処理時間については，KITE マイクロプロセッサ程度の回路規模の場合，初回コンパイルに10分程度の時間を要する．しかし，階層設計を行えばその後は変更を行った階層のみを再コンパイルすれば良く，実際に開発を行う際には問題とはならない．

5. 開発支援環境[8),9)]

設計したマイクロプロセッサを実際に動作させるために開発した，KITE マイクロプロセッサ・ボードとクロスソフトウェア環境について述べる．

5.1 KITE マイクロプロセッサ・ボード

KITE マイクロプロセッサ・ボードの写真を

表-3 設計サンプルの実装時間

CAD ツール	ホストコンピュータ	1バス構成	3バス構成
XC-DS 502-NC 2 V1.30 (Partition, Place, and Route Ver. 1.30)	NEC PC-9801 RA 21 (i 386 DX 20 MHz) Memory 10 Mbytes	01 : 44 : 40	01 : 30 : 16
	NEC PC-9801 FA (i 486 SX 16 MHz) Memory 12 Mbytes	01 : 04 : 18	00 : 44 : 59
XC-DS 502-SN 2 V1.41 (Partition, Place, and Route Ver. 1.30)	SUN SPARC Station 2 (SPARC 40 MHz) Memory 64 Mbytes	00 : 10 : 04	00 : 07 : 09
	SUN SPARC Station 10 (SUPER SPARC 40 MHz) Memory 128 Mbytes	00 : 07 : 06	00 : 04 : 48

図-4 に示す．このボードは FPGA 以外に，図-5 に示す機能を実装している．以下，図中の番号に従ってそれらの機能を説明する．

（1） KITE マイクロプロセッサの観測用端子の状態を表示するために，28 個の 7 セグメント LED ディスプレイと 25 個の LED ランプを用意している．

（2） 専用 CAD システムから構成データをダウンロード・ケーブルを介して FPGA に転送するロード機構を備えており，転送終了後ただちに FPGA はマイクロプロセッサとして動作を開始できる．また，動作検証済みの構成データ（1 バス構成，3 バス構成）を EPROM に書き込んでおり，コンピュータ入門教育用のワンボード・マイコンとしても利用できる[10]．

（3） メモリ空間に ROM と RAM をそれぞれ 2K ワード実装している．ROM 領域には簡単なテスト・プログラムを用意しており，マイクロプロセッサの実装後，ただちに動作検証をすることができる．一方，RAM 領域はデュアルポート構成でホストコンピュータから読み書きが行え，プログラムやデータを転送できる．

（4） 簡易の入出力装置として I/O 空間に，16 ビットの入力ポート（トグル・スイッチ）と出力ポート（LED ディスプレイ）を実装している．

（5） KITE マイクロプロセッサの動作を容易に把握できるようにクロック周波数を 1 MHz から 0.1 Hz までの範囲で（16 段階）変更でき，また高速な外部クロックの入力も可能である．さらに，動作モード選択機能により命令単位やクロック単位でもプログラム実行を制御できる．

上記機能によって，学習者はマイクロプロセッサの内部状態や動作を視覚的に把握することが可能となり，計算機の動作原理の理解や実装時のデバッグに役立つ．また，書換え可能な FPGA の採用でマイクロ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検証を行うことも可能である．

5.2 クロスソフトウェア環境

クロスソフトウェア環境では，アセンブラ，逆アセンブラ，Cコンパイラ，シミュレータを用意しており，ワークステーションやパソコン上で利用できる．シミュレータはアセンブラなどから出力されたオブジェクトコードに基づいてKITEマイクロプロセッサのシミュレーションを行えるが，そのオブジェクトコードを専用インタフェースを介して KITE マイクロプロセッサ・ボードの RAM 領域に転送できるので，設計したKITEマイクロプロセッサの詳細な動作検証に利用される．

また，このクロスソフトウェア環境はマイクロプロセッサ製作実験でのデバッグに役立つだけでなく，自らが設計したマイクロプロセッサを用いてアセンブリ言語やC言語によるプログラミング演習を行ったり，さらにはKITEマイクロプロセッサ用の言語処理系の作成演習にも利用できる．

6. 設計教育事例

九州工業大学情報工学部知能情報工学科では，1993年度より学部 3 年生を対象に教育用マイクロ

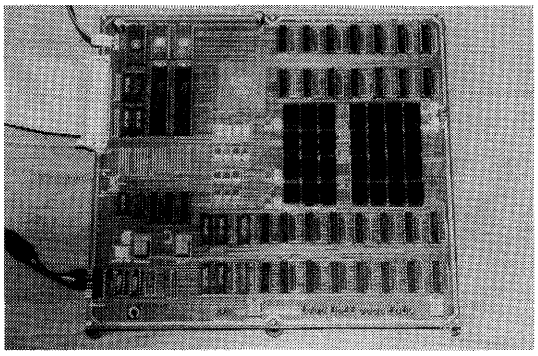


図-4 KITE マイクロプロセッサ・ボード

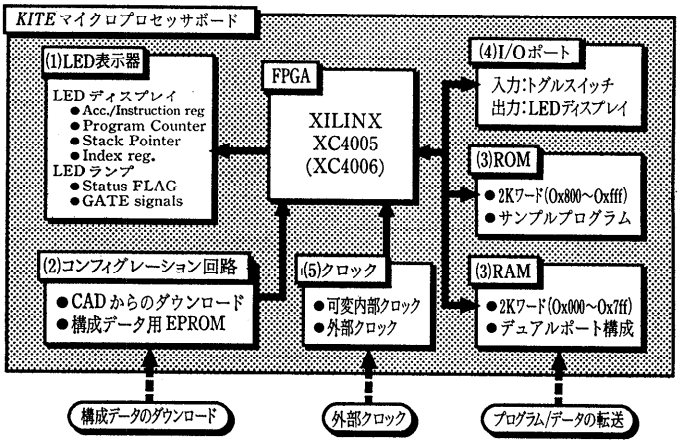


図-5 KITE マイクロプロセッサ・ボードの機能

プロセッサKITEを利用した設計教育を試行的に実施するとともに，1994年度からの本格的導入に向けてマイクロプロセッサを一人ですべて設計する実験を試みた[19]．学生実験風景を**図-6**に示す．本稿では，紙面の都合から後者について紹介する．

6.1 学生実験の概要

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計画である．そこで今回，複数の学生にKITEマイクロプロセッサの仕様書のみを与えて，各自が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予備実験を行った．開発期間には特に制限を設けず，CAD操作法の修得からマイクロプロセッサの完成まで挑戦してもらい，実装結果，実装時間，開発期間を調べた．

開発方法としては現在学生実験で利用している回路図入力による設計に加え，ハードウェア記述言語(VHDL)による設計も採用した．また，開発環境は**図-7**のように，負荷は軽いが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パソコンを利用する一方，負荷の重い自動配置配線にはワークステーションを利用した．

6.2 回路図入力による開発結果

回路図入力によるKITEマイクロプロセッサの開発には，9人の学生が挑戦した．実装結果を

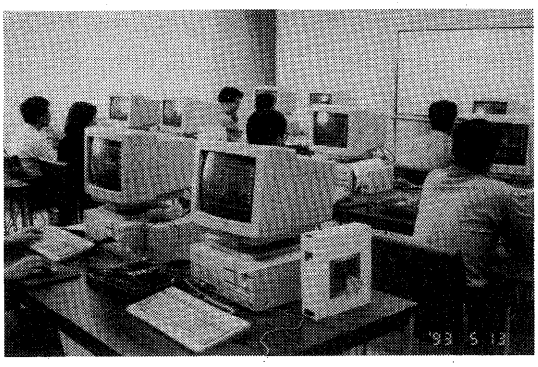


図-6 学生実験風景

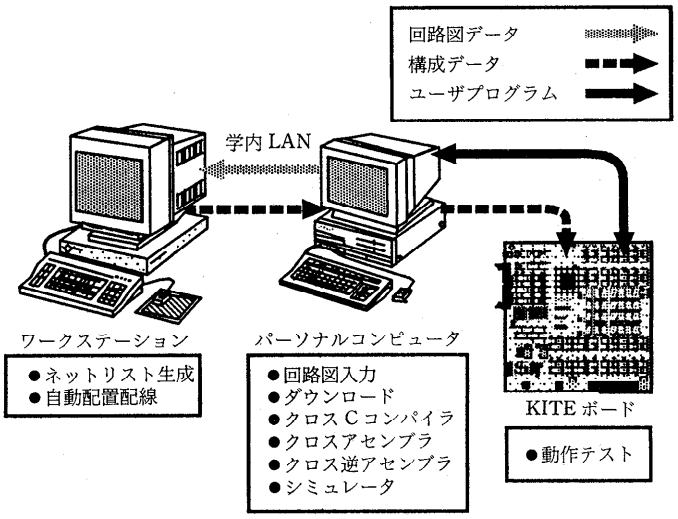


図-7 学生実験の開発環境

表-4 回路図入力による実装結果

CAD ツール出力情報 XC-DS502-SN2 V1.41 Partition, Place, and Route Ver. 1.30	学生 A	B	C	D	E	F	G	H	I
バス構成	1バス				2バス	3バス			
換算ゲート数	4052	3962	3629	3584	3318	4271	3252	3218	3712
FPGA 内利用 (利用率)									
使用論理ブロック数:	195(99%)	196(100%)	195(99%)	188(95%)	174(88%)	195(99%)	173(88%)	176(89%)	190(96%)
理想論理ブロック数:	182(92%)	181(92%)	168(85%)	163(83%)	145(73%)	176(89%)	148(75%)	142(72%)	176(89%)
フリップフロップ数:	135(21%)	148(24%)	134(21%)	135(21%)	118(19%)	101(16%)	102(16%)	103(16%)	101(16%)
3ステートバッファ数:	116(23%)	116(23%)	116(23%)	116(23%)	100(19%)	100(19%)	116(23%)	100(19%)	100(19%)
配線摘要 総配線数:	4260	4286	4029	3876	3288	3690	3326	3130	3503
CPU 時間 (Sun SS10/41)	00:07:21	00:10:19	00:06:59	00:05:50	00:03:45	00:05:07	00:03:41	00:05:59	00:04:02
クロック数/命令	5.51	5.86	6.16	7.16	5.12	4.84	4.89	4.84	4.84

表-5 回路図入力による開発期間

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	65	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	15	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，( )内は日数

表-6 ハードウェア記述言語による実装結果

CAD ツール出力情報 XC-DS502-SN2 V1.41 Partition, Place, and Route Ver. 1.30	学生				
	V	W	X	Y	Z
バス構成	1 バス			3 バス	
換算ゲート数	3725	4057	4140	3711	4432
FPGA 内利用 (利用率)					
使用論理ブロック数:	177 (90%)	192 (97%)	191 (97%)	182 (92%)	196(100%)
理想論理ブロック数:	131 (66%)	150 (76%)	161 (82%)	147 (75%)	179(91%)
フリップフロップ数:	125 (20%)	146 (23%)	148 (24%)	97 (15%)	132(21%)
3ステートバッファ数:	116 (23%)	152 (30%)	144 (28%)	124 (24%)	96(19%)
配線摘要 総配線数:	2702	3883	3661	3329	4082
CPU 時間 (Sun SS10/41)	00 : 03 : 09	00 : 06 : 50	00 : 05 : 25	00 : 04 : 20	00 : 11 : 26
クロック数／命令	5.88	5.88	5.78	5.04	4.92

表-4 に，開発期間を**表-5** に示す．

表-4 から分かるように全員が完成させ，いずれも現在使用している FPGA の許容範囲内で実装できている．バス構成にも3種類があり，同じバス構成でもシーケンサや ALU などの設計の違いによって命令当たりのクロック数や回路規模に差が生じているように，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ている．

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て開発

表-7 ハードウェア記述言語による開発期間

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	2	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	10	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，( )内は日数

期間に影響を及ぼしたが，平均開発時間は230時間程度，平均開発日数は25日程度であった．実際の学生実験では二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．

6.3 ハードウェア記述言語による開発結果

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には，過去にVHDLを使用したことのない学生5人が挑戦した．なお，VHDL用論理合成ツールを利用する場合，仕様を完全に満たすものは自動合成される論理回路のゲート規模の点で現在使用しているFPGAには実装できないため，**4.2**で述べたサブセット版を実装した．実装結果を**表-6**に，開発期間を**表-7**に示す．

全員が完成させているが，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あった．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．

開発期間にはVHDLの学習時間も含まれるが，平均開発時間は152時間程度，平均開発日数は20日程度であった．つまり，プログラミングやOSの操作法についての素養があれば，ハードウェア記述言語の使用経験がない初心者でも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．

7. むすび

本稿では，FPGAが実現した教育への応用例として教育用マイクロプロセッサKITEを紹介し，設計教育事例について報告した．FPGAを利用することによって，大学においてもASIC時代にふさわしい設計教育として，LSI実装を前提とするマイクロプロセッサの設計が学生実験の課題として実施できることを示した．書換え可能なFPGAは，階層設計の過程を通じて計算機構成要素の段階的開発が行える上，デバッグや改良のために何回でも設計のやり直しができるので，論理回路設計に経験が浅い学部学生の教育にも重宝である．

設計教育事例で紹介した実験で最大の収穫は，学生の論理回路や計算機システムに対する理解と学習意欲を高めるのに非常に大きい効果が認められたことである．自分のアイデアを盛り込んだマイクロプロセッサを苦労の末に実際に動く「もの」として完成させると，従来のハードウェア実験で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表れている．また，一部の学生はさらにKITEマイクロプロセッサ用Cコンパイラの作成を試み，完成させている．

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，FPGAを利用した設計教育を実施される際の参考になれば幸いである．

参考文献

1) 電子情報通信学会九州支部：VLSIシステムの設計をどのように教育するか？ 電子情報通信学会創立75周年記念シンポジウム会議録，pp. 26-49 (1992).
2) 神原弘之，安浦寛人：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，Vol. 33, No. 2, pp. 118-127 (Feb. 1992).
3) 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ー，電子情報通信学会誌，Vol. 75, No. 5, pp. 530-533 (1992).
4) 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).
5) 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).
6) 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).
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) 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).
11) 広瀬全孝：我が国における半導体技術開発の方向，第1回LSI設計教育・研究連絡協議会資料，p. 18 (1993).
12) 生駒俊明，浅田邦博：大学および大学院におけるVLSI教育の現状とあるべき姿，平成5年度大学電気工学教育研究集会分科会予稿集，pp. 49-52 (1993).
13) 末吉，船越，有田：FPGA専用CADによる論理回路設計教育の学生実験事例，電気関係学会九州支部連合大会講演論文集，No. 1025 (1991).
14) Dunlop, J., Girma, D., Lysaght, P.: Alternative Approach to ASIC Design Methodology Based on Reconfigurable Logic Device, IEE Proc. G, Circuits Device Syst., Vol. 139, No. 2, pp. 217-

221 (1992).
15) Xilinx, Inc.: Programmable Gate Array Databook (1992).
16) IEEE: IEEE Standard VHDL Language Reference Manual (1988).
17) Cadence Design Systems, Inc.: Verilog-XL Reference Manual (1991).
18) Data I/O, Corp.: ABEL Design Software User Manual (1990).
19) 末吉，田中，久我：教育用マイクロプロセッサ KITE を利用した設計教育の事例報告，情報処理学会研究報告，93-ARC-103-12 (1993).

(平成5年9月21日受付)

末吉 敏則（正会員）

1953 年生．1976 年九州大学工学部情報工学科卒業．1978 年同大学院工学研究科情報工学専攻修士課程修了．同年九州大学工学部情報工学科助手．同大学院総合理工学研究科助教授を経て，1989 年より九州工業大学情報工学部知能情報工学科助教授．現在，同大学マイクロ化総合技術センター助教授を兼務．工学博士．計算機アーキテクチャ，システムソフトウェア，計算機ネットワーク，LSI 設計などの研究に従事．著書「並列処理マシン」(共著)．電子情報通信学会会員．