

## 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

末吉 敏則†‡ 田中 康一郎† 柴村 英智†

†九州工業大学情報工学部  
知能情報工学科

‡九州工業大学情報工学部  
マイクロ化総合技術センター

sueyoshi@ai.kyutech.ac.jp

あらまし

再構成可能な論理LSIとそのCADシステムを利用して、学生が自らの手で設計、実装、動作検証を行えるように配慮した教育用マイクロプロセッサKITEとその開発支援環境について述べる。KITEは基本的な計算機構成要素から構成される簡素な16ビットマイクロプロセッサであり、計算機入門教育におけるノイマン型計算機の動作原理の理解にも活用できるように、プロセッサ内部の動作や状態が容易に把握できる可観測性に配慮した仕様となっている。実装結果から、LSI実装を前提とするマイクロプロセッサの設計が論理回路設計演習の一課題として、大学等における学生実験の厳しい時間的制約の枠内で実施できることを確認できた。

和文キーワード マイクロプロセッサ, 論理設計, CAD, ハードウェア記述言語, 計算機アーキテクチャ, 教育

## KITE: An Educational Microprocessor Using Reconfigurable Logic Device

Toshinori Sueyoshi †‡ Kouichiro Tanaka † Hidetomo Shibamura †

† Department of Artificial Intelligence  
Kyushu Institute of Technology  
Iizuka, 820 Japan

‡ Center for the Microelectronic Systems  
Kyushu Institute of Technology  
Iizuka, 820 Japan

sueyoshi@ai.kyutech.ac.jp

Abstract

This paper presents an educational microprocessor called KITE and the development environment where the students can individually design, implement and verify their own microprocessor design using a reconfigurable logic device and CAD system. The KITE microprocessor features a simple 16-bit microprocessor consisting of the basic computer components. Furthermore, the operations and status within the microprocessor are highly observable from the outside providing students the view to understand the operation principles in the introductory Computer Science course. The implementation results have confirmed that LSI implementation of microprocessor design as an exercise in logic circuit design education can be performed within the critical time restriction requirement in the curriculum.

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

## 1. はじめに

半導体集積度の向上による開発期間の長期化、多品種少量生産のASIC化の進展に伴い、相対的にLSI開発工程における設計の比重が増し、CAD技術の重要性ならびにCADツールを利用したLSI設計教育の必要性が益々高まってきた。一方、現在大学等の教育機関で行われている論理設計科目では、LSI実装を前提として実用規模の論理回路を題材にCADシステムを使用した設計教育が行われているのはまだ少数である[1],[2],[3]。また、情報工学の教育において論理設計と密接な教科目である計算機アーキテクチャに至っては講義のみで行われている場合が多い。さらに、オペレーティングシステム(OS)やコンパイラ等のシステムソフトウェアに関する教育とも遊離し、一貫した計算機教育の実施が難しいと言える。実際の計算機ハードウェアを用いた実験演習が行われる場合は、市販のマイクロプロセッサを教材として利用しているのが現状である[1],[4]。しかし、市販のマイクロプロセッサでは、内部状態を直接把握できない、内部動作が複雑である、アーキテクチャに非公開部分がある等の理由から、上記教科目の教材として最適とは言い難い。

情報工学の教育の場合、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学習者自身がCADシステムを利用して当該計算機の設計演習を行うことが望ましい。さらには、当該計算機を対象とするOSやコンパイラ等を作成させることにより計算機の管理技術を学習させ、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、計算機の方式設計に反映させることにより、計算機システムに対するより効果的な教育が期待できる。

そこで、本研究では計算機の動作原理、構成原理、管理技法まで一貫した教育を行え、かつ学生自身の手で設計から動作確認まで行える簡素なアーキテクチャをもつ教育用マイクロプロセッサKITE(Kyushu Institute of Technology Educational microprocessor)の開発を行った。教育に用いる設計手法としては、スキマティックエディタを用いた回路図入力による設計と、ハードウェア記述言語を用いた機能記述/動作記述による設計の2通りが考えられる。今回は両者を学生実験という教育の観点から比較することも狙って、これらを併用して開発を行った。

以下、第2章では教育用マイクロプロセッサの開発に使用する実装デバイスとその開発支援環境について述べる。また、第3章ではKITEマイクロプロセッサの設計仕様について述べ、第4章ではそのLSI実装について言及する。そして、第5章で評価用ボードを用いた動作検証について述べる。最後に、第6章で簡単なまとめを述べる。

## 2. 開発支援環境

### 2.1 実装デバイス

マイクロプロセッサの開発を行うには、大学における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、開発支援環境は以下のような要求を満たす必要がある。

- (1) 論理設計の経験が浅い学部学生が設計するので、バグが入ることを前提にして、バグ取りや改良のために何回でも設計し直せること。
- (2) 学生が設計したマイクロプロセッサはすべてLSI化し、全員が手に取って動作確認をできること。
- (3) 多数の学部学生が参加する実験演習を実験演習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実装できること。

これらの要件を満たすには、現在のところ大学等の教育機関におけるLSI教育環境では無理がある。そこで、本研究では教育用マイクロプロセッサの実装デバイスとして書換え可能なLSIの一種であるFPGA(Field Programmable Gate Array)を採用し、前述の要求仕様を満たす教育用マイクロプロセッサの開発を行うことにした。

FPGAとは内部回路を使用者が自由にプログラム設定を行えるゲートアレイの総称であり、本研究では米国Xilinx社のLCA(Logic Cell Array)を使用している。LCAは教育用マイクロプロセッサのような数千ゲート相当の論理回路の実現に十分対応可能であり、他のFPGAに比べてゲートアレイに近い構造であるため設計の自由度が高い。さらに、LCAはバス・トランジスタ型、つまりSRAM制御方式のFPGAであるため書換え回数に制限がなく、構成データ(configuration data)を再ロードするだけで直ちに動作検証が行え、予算的/時間的制約が厳しい学生実験に適している。

### 2.2 開発フロー

論理回路システムを設計する手法としては、スキマティックエディタを利用した回路図入力による設計法と、最近急速に普及しつつあるハードウェア記述言語による設計法がある。現在はKITEの設計に際して、図1に示すようにこの両方の設計法を利用することができる。以下に、各設計法での開発の流れを簡単に述べる。

#### [1] スキマティックエディタによる設計

LCA専用CADシステムにはスキマティックエディタが用意されており、学習者は設計仕様に基づき論理回路を設計した後、これ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が用意されているため、比較的容易に論理回路を入力することができる。回路図の入力終了後、

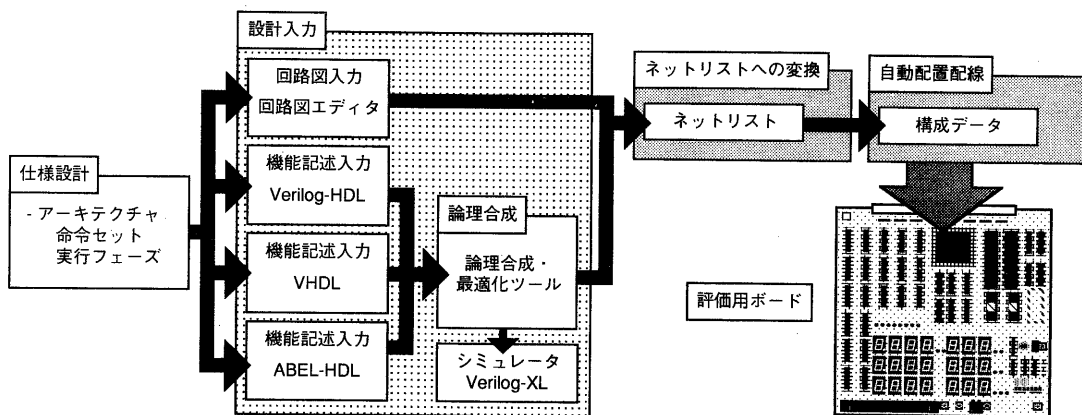


図1 開発フロー

そのデータをXNF (Xilinx Netlist Format) と称するネットリストへ変換する。この段階で、ネットの誤り等の簡単な間違いをチェックすることができる。その後、LCA専用CADシステムの自動配置配線機能を利用してLCAにダウンロード可能な構成データに変換する。この自動配置配線機能を利用することにより、学習者はLCA内の配置配線問題に起因する様々な制約から解放され、各自のアイデアを生かす論理回路の設計に専念できる。そのため、LSI設計を専門としない学生でも実際にシステムとして動作するマイクロプロセッサの設計を修得することができる。

[2] ハードウェア記述言語による設計

現在、LCA設計に利用できるハードウェア記述言語として、Verilog-HDL[6]、VHDL [7]、ABEL-HDL [8]という代表的な言語を用意している。たとえば、最も実績のあるVerilog-HDLの場合、LCA設計を対象とするために論理合成ツール (Synopsys [9], Exemplar [10]) と組み合わせて使用する。Verilog-HDLでは機能レベル記述と動作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。また、記述した論理回路はVerilog-XLという強力なシミュレータを使って内部動作を細部にわたって検証できるため、誤りの少ない回路を実現できる。シミュレーション終了後、論理合成ツールを使ってネットリスト (XNF) を生成する。その後は、スキマティックエディタによる設計の場合と同様、LCA専用CADシステムの自動配置配線機能を利用して構成データを生成する。

論理回路の設計原理よりもプロセッサの動作原理に重きをおいた教育をしたい場合、特に情報工学専攻の教育の場合には、回路図入力よりもむしろハードウェア記述言語を用いて記述した方が、より短時間で高レベルの学習効果が期待できる。つまり、学習者は計算機システム上でプログラムを作成

していたのと同じ感覚でマイクロプロセッサを設計できる。教育の目的に合わせて、回路図入力による設計法かハードウェア記述言語による設計法のいずれかを選択できる。なお、ハードウェア記述言語の場合には、論理合成ツールを利用することによって設計完了後必要に応じてゲートアレイやスタンダードセルに実装することも容易である。

2.3 開発支援環境のシステム構成例

実際に学生実験に適用する場合の開発支援環境のシステム構成例を図2に示す。相対的に対話的な操作に時間を費やすスキマティックエディタによる回路図入力、構成データのダウンロード、動作テスト、ならびにクロスソフトウェアを利用したプログラム開発は学習者それぞれに割り当てられるパーソナルコンピュータを用いて行い、KITEの実装で最もCPU負荷の重い自動配置配線処理はローカルエリアネットワークを介して高性能ワークステーション上で実行する。高性能ワークステーションを利用することで開発時間の短縮を図り、学生実験の時間的制約を満たすことが可能となる。

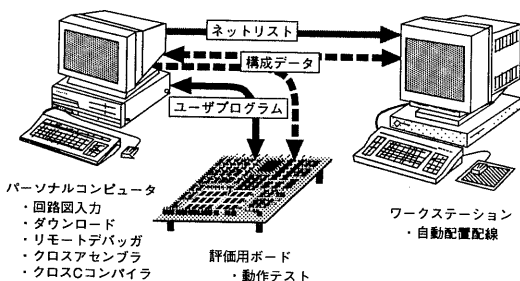
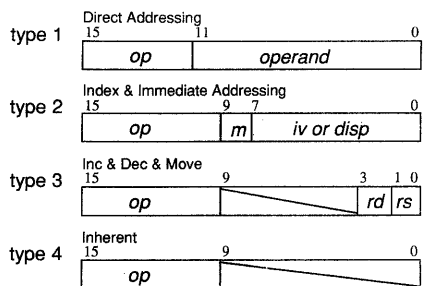


図2 開発支援環境のシステム構成例



op = operation code      m = addressing mode  
 disp = displacement      iv = immediate value  
 rd = destination register      rs = source register

図3 命令形式

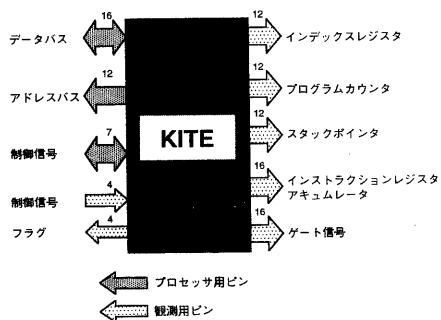


図4 入出力端子

### 3. KITEマイクロプロセッサの設計仕様

教育用マイクロプロセッサを開発するに当たり、前述の目的を達成するために設計仕様は以下に示すような要件を満たす必要がある。

- (1) プロセッサ・アーキテクチャは集積度を考慮したできるだけ簡素なもので、かつ論理回路の基本的構成要素はすべて含んでいること。
- (2) 命令セットは、教育用マイクロプロセッサとして必要と思われる最小限の命令を用意すること。
- (3) 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) 特別な測定器を用いなくても、計算機の動作を逐一確認できるように、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。

表1 命令セット

Mnemonic	Addressing Mode				Explanation	Flags C Z S V
	Direct	Index	Immediate	Inherent		
LD	0000	11000010	11000001	-	ACC ← M	-----
ST	0001	11000110	-	-	M ← ACC	-----
MV	-	-	-	111100	DREG ← SREG	-----
JPC	0100	-	-	-	If (z=1) (PC ← IR or PC ← M)	-----
JPz	0101	-	-	-	If (z=1) (PC ← IR or PC ← M)	-----
JPs	0110	-	-	-	If (s=1) (PC ← IR or PC ← M)	-----
JPo	0111	-	-	-	If (s=1) (PC ← IR or PC ← M)	-----
JP	0011	-	-	-	PC ← M	-----
CALL	0010	-	-	-	S ← PC, SP ← SP-1, PC ← IR	-----
RET	-	-	-	11100	SP ← SP+1, PC ← S	-----
ADD	-	10000010	10000001	-	ACC ← ACC+M	****
SUB	-	10000110	10000101	-	ACC ← ACC-M	****
INC	-	-	-	100010	ACC ← ACC+1 or IXR ← IXR+1	****
DEC	-	-	-	100011	ACC ← ACC-1 or IXR ← IXR-1	****
NOT	-	-	-	100111	ACC ← ~ACC	0*0*
OR	-	10010010	10010001	-	ACC ← ACC   M	0*0*
AND	-	10011010	10011001	-	ACC ← ACC & M	0*0*
EOR	-	10010110	10010101	-	ACC ← ACC ⊕ M	0*0*
LSL	-	-	-	101000	ACC ← ACC << 1	****
ASL	-	-	-	101001	ACC ← ACC << 1, C ← ACC[7]	****
LSR	-	-	-	101010	ACC ← ACC >> 1, C ← ACC[0]	****
ASR	-	-	-	101011	ACC ← ACC >> 1, C ← ACC[0]	****
ROL	-	-	-	101100	ACC ← ACC << 1, C ← ACC[7], ACC[7] ← C	****
ROR	-	-	-	101101	ACC ← ACC >> 1, C ← ACC[0], ACC[0] ← C	****
SWP	-	-	-	101111	ACC ↔ M	0*0*
POP	-	-	-	110100	S ← ACC, SP ← SP-1	-----
PUSH	-	-	-	110101	SP ← SP+1, ACC ← S	-----
IN	-	-	-	110010	ACC ← IBUF	-----
OUT	-	-	-	110011	OBUF ← ACC	-----
HALT	-	-	-	111110	-	-----
NOP	-	-	-	111111	PC ← PC+1	-----

ACC = Accumulator  
 M = Memory  
 DREG = Destination  
 SREG = Source Register

PC = Program Counter  
 IR = Instruction Register  
 SP = Stack Pointer  
 S = Stack

IBUF = In Buffer  
 OBUF = Out Buffer

本研究ではこれらの要求を考慮して、教育用マイクロプロセッサの設計を行った。今回開発した教育用マイクロプロセッサは、要求仕様ならびに実装に使用するLCAの集積度を考慮し、4Kワード(8Kバイト)の記憶空間をもつ16ビット・マイクロプロセッサである。

命令形式は16ビット固定長であり、図3に示すように4種類のみとしている。これにより命令フェッチは1回だけで済み、マイクロプロセッサのシーケンス制御部ならびに命令デコードの単純化を図ることができる。また、命令は表1に示すように、ソフトウェアの作成に不自由とならない程度の基本命令として31種類を用意し、アドレス方式としては直接、インデックス修飾、即値の3種類を指定できる。

KITEマイクロプロセッサの入出力端子としては図4に示すように、通常のアドレスバス、データバス、制御信号の他に、プロセッサ内部の動作や状態を把握するための出力端子を設けて可観測性を高めている。観測対象としては、アキュムレータ、プログラムカウンタ、インデックスレジスタ等のレジスタおよびゲート開閉信号を含んでいる。これらすべて動作検証を行う評価用ボード上に実装されているLEDに表示され、プロセッサの動作を逐一に観測できる。また、動作原理の理解ならびに動作検証を容易にするため、通常動作、命令単位動作、クロック単位動作という3つの動作モードを備えている。

表2 実行フェーズ

(a) 1バス構成

Mnemonic	Phase	0	1	2	3	4	5	6
LD	direct		MAR←(IR)	MEM←(MAR)				
	immediate		ACC←(IR)					
	index		BUF1←(IR)	ALU←(BUF1) ALU←(IXR) BUF2←(ALU)	MAR←(BUF2)	MEM←(MAR)	ACC←(MEM)	
ST	direct		MAR←(IR)	MEM←(MAR)				
	immediate		ACC←(IR)					
	index		BUF1←(IR)	ALU←(BUF1) ALU←(IXR) BUF2←(ALU)	MAR←(BUF2)	MEM←(MAR)	MEM←(ACC)	
MV			REG←(REG)					
JP	direct		PC←(IR)					
ADD SUB	immediate		BUF1←(ACC)	ALU←(BUF1) ALU←(IR) BUF2←(ALU)	ACC←(BUF2)			
	index		BUF1←(IR)	ALU←(BUF1) ALU←(IXR) BUF2←(ALU)	MAR←(BUF2)	BUF1←(ACC)	ALU←(BUF1) MEM←(MAR) ALU←(MEM) BUF2←(ALU)	ACC←(BUF2)
AND OR EOR			MEM←(PC) IR←(MEM) PC←	BUF1←(ACC)	ALU←(BUF1) ALU←(IR) BUF2←(ALU)	ACC←(BUF2)		
NOT INC DEC LSL ASL LSR ASR ROL ROR SWP			MEM←(PC) IR←(MEM) PC←	BUF1←(ACC)	ALU←(BUF1) ALU←(IR) BUF2←(ALU)	ACC←(BUF2)		
CALL			BUF1←(SP) MAR←(SP)		SP←(BUF2)	MEM←(MAR) MEM←(PC)	PC←(IR)	
RET			BUF1←(SP)		SP←(BUF2) MAR←(BUF2)	MEM←(MAR) PC←(MEM)		
PUSH			BUF1←(SP) MAR←(SP)		SP←(BUF2)	MEM←(MAR) MEM←(ACC)		
POP			BUF1←(SP)		SP←(BUF2)	MEM←(MAR) ACC←(MEM)		
IN			ACC←(BUF)		MAR←(BUF2)	ACC←(MEM)		
OUT			OBUF←(ACC)					
NOP								
HALT								

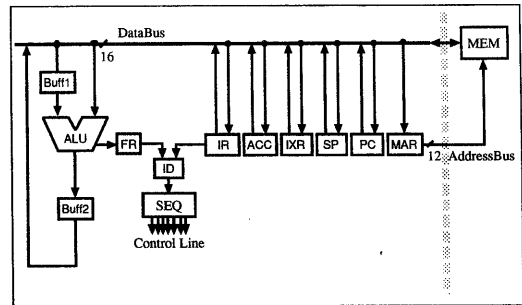
(b) 3バス構成

Mnemonic	Phase	0	1	2	3
LD	direct		MAR←(IR)	MEM←(MAR)	ACC←(MEM)
	immediate		ACC←(IR)		
	index		ALU←(IR) ALU←(IXR) MAR←(ALU)	MEM←(MAR)	ACC←(MEM)
ST	direct		MAR←(IR)	MEM←(MAR)	MEM←(ACC)
	immediate		ACC←(IR)		
	index		ALU←(IR) ALU←(IXR) MAR←(ALU)	MEM←(MAR)	MEM←(ACC)
MV			REG←(REG)		
JP	direct		PC←(IR)		
ADD SUB	immediate		ALU←(ACC)	ALU←(IR)	ACC←(ALU)
	index		ALU←(IR) ALU←(IXR) MAR←(ALU)	ALU←(ACC)	MEM←(MAR) ALU←(MEM) ACC←(ALU)
NOT INC DEC LSL ASL LSR ASR ROL ROR SWP			MEM←(PC) IR←(MEM) PC←	ALU←(ACC) ALU←(IR) ACC←(ALU)	
CALL			ALU←(SP) MAR←(SP) SP←(ALU)	MEM←(MAR) MEM←(PC)	PC←(IR)
RET			ALU←(SP) SP←(ALU) MAR←(ALU)	MEM←(MAR) PC←(MEM)	
PUSH			ALU←(SP) MAR←(SP) SP←(ALU)	MEM←(MAR) MEM←(ACC)	
POP			ALU←(SP) SP←(ALU) MAR←(ALU)	MEM←(MAR) ACC←(MEM)	
IN			ACC←(BUF)		
OUT			OBUF←(ACC)		
NOP					
HALT					

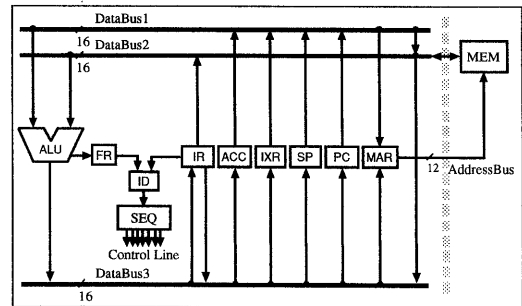
4. KITEマイクロプロセッサの実装

マイクロプロセッサを題材とした論理設計教育の実施に際してプロセッサ内部の構成方式の相違によって生じる設計時間や実装結果（ゲート規模）の違いを把握するために、計算機構成として図5に示すような1バス構成と3バス構成という2種類のKITEマイクロプロセッサを実装した。これはまた、計算機アーキテクチャの教育において構成方式によるプロセッサ動作の差異を理解するための教材としても役立つ。表2にそれぞれの実行フェーズを示す。両者を比較すると、たとえばインデックス修飾アドレッシングの加算命令では極端な実行時間の差があるが、分岐命令のように同じ実行時間を要する命令もある。これにより、ノイマン型計算機の動作原理および構成原理の理解にも大きな効果が期待できる。

マイクロプロセッサの構成要素は、記憶回路要素、組合せ回路要素、順序回路要素の3つに大別できる。記憶回路要素としてはアキュムレータ等のレジスタおよびバッファがあり、Dフリップフロップを用いて構成した。なお、プログラムカウンタはインクリメント機能や分岐命令のためのプリセット機能を必要とするため、プリセットアップカウンタとして構成している。また、組合せ回路要素としては算術論理演算器（ALU）と命令デコーダがあり、ALU内の加算器はリプルキャリー方式ならびにキャリールックアヘッド方式の2種類について実装を行った。そして、順序回路要素とし



(a) 1バス構成



(b) 3バス構成

図5 計算機内部構成

てはシーケンス制御部がある。シーケンス制御部の実現方式には布線論理方式とマイクロプログラム方式があるが、実装デバイスのLCAでは内部の論理セル(CLB)を記憶要素として用いるとゲート利用率が著しく低下するため、さらには順序回路の設計自体も教育目的の1つであるために、ここでは布線論理方式によって構成している。

今回使用する実装デバイスとしては、新しいLCA4000シリーズのうち5000ゲート相当の論理回路を実装可能なXC4005を採用している。LCA内のCLBの利用効率として90%程度を目標に設計した。1バス構成ならびに3バス構成のマイクロプロセッサの実装結果を表3に示す。この実装結果から、1バス構成のマイクロプロセッサは3バス構成と比べてフリップフロップ数やゲート数は多いことが分かる。これは、シーケンス制御部における状態数の多さやALUの入出力部分のバッファが原因である。一方、配線量に関しては3バス構成のマイクロプロセッサの方が多くなっている。これは、バス数の増加に伴ってレジスタ等の構成要素がマルチポート化するためである。表3から分かるように、これらは自動配置配線を行う際のCPU時間に大きな影響を及ぼしており、自動レイアウトにおける配線問題の重要性が認識できる。

学生実験における論理回路設計演習テーマとしてマイクロプロセッサの開発を予定しているため、開発時間は最も重要な要因といえる。論理回路の設計完了後、実装のために最も時間のかかる作業は自動配置配線である。前回、旧シリーズのLCA3000シリーズを用いた設計では、自動配置配線にNECのパーソナルコンピュータPC-9801RA21で約36時間を要し、これが学生実験のテーマとして実施するには大きな障害となっていた[11],[12]。今回の実装時間を、1バス構成の場合について利用計算機毎を表4に示す。サンマイクロシステムズ社のワークステーションSPARC Station 2を使用して最新の自動配置配線プログラム(PPR Ver 1.20)を実行すると、最大遅延に制限を設けない場合で約9分、最大遅延に制限を設ける場合でも約3時間で実装できている。本実験では最大遅延は実際ほとんど問題にならないので、SPARC Station 2における実装時間程度であれば3バス構成の場合でも学生実験の厳しい時間的制約を満たすことができ、本テーマの学生実験実施に向けて良好な結果が得られた。

## 5. 動作検証

KITEを実際に動作させるには、LCAに実装して稼働させる環境が必要となる。そこで、設計完了後のKITEの動作検証を行うために、また計算機教育の教材として利用するために評価用ボードを製作した。この評価用ボードには、KITEの動作環境として次のものを実装している。

表3 実装結果

SPARC Station 2 (SPARC 40MHz Memory 48Mbytes)	バス構成	
	1バス	3バス
<b>Input XNF Design Statistics</b>		
Number of Logic Symbol	817	783
Number of Flip Flops	134	91
Number of 3-state Buffers	132	160
Number of IO Pads	111	111
Number of Nets	1151	1152
Number of Pins	4170	4006
Equivalent "Gate Array" Gates	3809	3475
<b>Partitioned Design Utilization</b>		
Occupied CLBs	93%	98%
Packed CLBs	77%	75%
Package Pins	99%	99%
Flip Flops	26%	18%
3-state Buffers	23%	31%
3-state Buffers Output lines	28%	85%
<b>Routing Summary</b>		
Number of unrouted connections	0	0
Number of pips used	5446	6382
Number of local lines used	2344	2808
Number of double lines used	684	962
Number of long lines used	307	387
Number of global lines used	22	26
<b>CPU Times</b>	hr.min.sec	hr.min.sec
Netlist Preparation	00:00:06	00:00:06
Partition/Placement	00:02:37	00:01:41
Placement Improvement	00:03:56	00:01:17
Routing	00:01:32	00:33:06
Total	00:08:45	00:36:39

表4 実装時間(1バス構成の場合)

PPR Ver 1.10	最大遅延制限なし (Timing off)	最大遅延制限あり (Timing on)
NEC PC-9801RA21 (i386DX 20MHz) Memory 10Mbytes	<b>14:13:57</b>	<b>41:17:40</b>
NEC PC-9801FA2 (i486SX 16MHz) Memory 12Mbytes	<b>06:39:33</b>	<b>37:10:37</b>
SUN SPARC Station 2 (SPARC 40MHz) Memory 48Mbytes	<b>01:05:36</b>	<b>03:56:20</b>
PPR Ver 1.20	最大遅延制限なし (Timing off)	最大遅延制限あり (Timing on)
SUN SPARC Station 2 (SPARC 40MHz) Memory 48Mbytes	<b>00:08:45</b>	<b>03:18:12</b>

- (1) 専用CADシステムによって生成された構成データをダウンロードケーブルを介してLCAへ転送するロード機構を備えており、転送終了後直ちにLCA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は専用のEPROMに書き込んでおくことによって評価用ボードの電源投入と共にLCAへロードでき、スタンドアロンで計算機教育に利用することもできる。

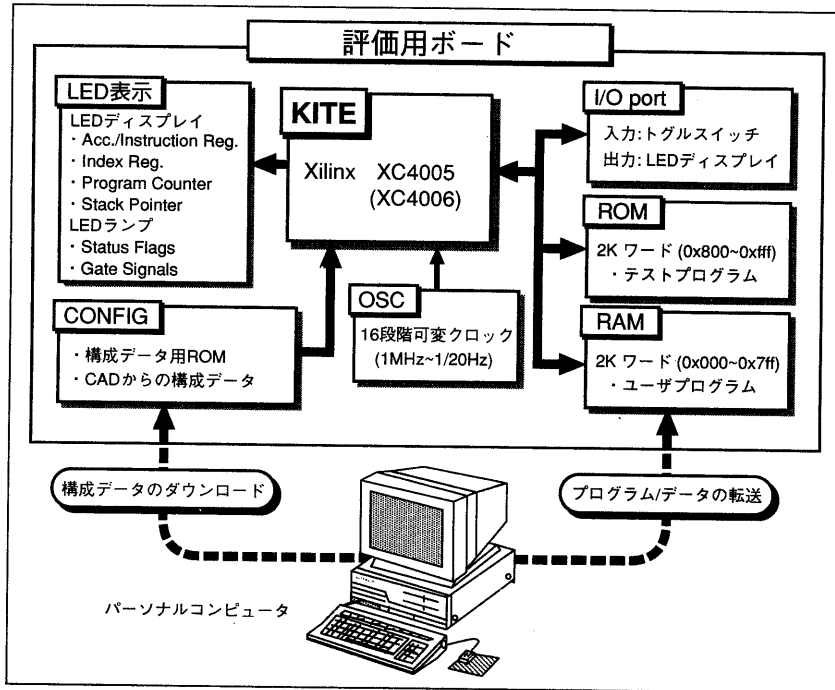


図6 評価用ボードの概要

- (2) K I T Eは4 Kワード(8 Kバイト)のメモリ空間を持つ。評価用ボードには、K I T Eの主記憶としてRAMとROMをそれぞれ2 Kワード実装している。ROM領域にはK I T Eの動作テスト用に簡単なプログラムを用意しており、構成データのダウンロード完了後、直ちに動作テストを行える。一方、RAM領域にはデュアルポートRAM構成になっており、パーソナルコンピュータからプログラム/データの転送ができる。パーソナルコンピュータ上に用意したクロスアセンブラやクロスCコンパイラ、あるいはリモートデバッガを用いることで、より詳細にK I T Eの動作テストを行うことができる。
- (3) K I T EのI/Oポートとしては、入力ポートと出力ポートがそれぞれ1個ずつある。評価用ボードにはトグルスイッチによる16ビット入力とLEDディスプレイ(7セグメントLED)の表示による16ビット出力を用意している。
- (4) K I T Eはプロセッサ内部の状態(レジスタの値、制御信号など)を観測するために、72本の信号を出力している。評価用ボードでは、これらの出力を20個のLEDランプと13個のLEDディスプレイに表示し、プロセッサ内部の状態を常に観測することができる。
- (5) K I T Eはプロセッサの動作の観測を容易にするために、通常動作、命令単位動作、クロック単位動作という3つの

動作モードを備えている。評価用ボードでは、この機能を利用してプロセッサの動作を容易に観測できるが、さらにクロック周波数として1 MHzから1/20 Hzの範囲で16通りの値を選択でき、プロセッサの動作速度も自由に調整できる。これにより、K I T Eにおけるプログラムの連続的な実行過程も視覚的に追尾することが可能である。

この評価用ボードの概要を図6に示す。また、この評価用ボードを含む実際の開発支援環境、ならびに評価用ボードの写真をそれぞれ図7、図8に示す。学生は設計したマイクロプロセッサを評価用ボード上で動作させることにより、プロセッサの動作を視覚的に捉え、容易に動作テストを行うことができる。さらに、開発デバイスとしてLCAを利用しているため、必ずしもマイクロプロセッサ全体を一度に実装して動作テストを行う必要はなく、マイクロプロセッサの構成要素ごとに実装・動作テストを行いながら段階的に設計を進めることもできる。たとえば、マイクロプロセッサの構成要素のうち、算術論理演算回路の部分抜き出して設計・実装を行い、その部分だけの動作テストを行うことができる。その際、本来はプログラムカウンタの内容を表示するLEDディスプレイに算術論理演算回路のバッファ内容等を表示するなど、端子割当ては自由に行えるので、特別な測定器がなくとも構成要素ごとに動作テストを行える。

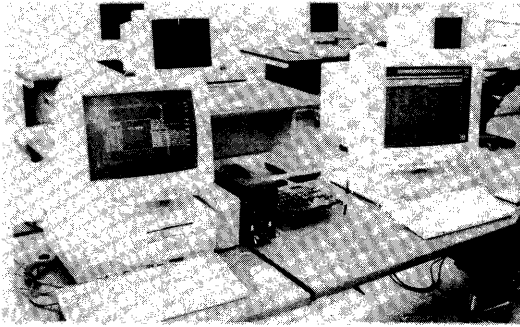


図7 実際の開発支援環境

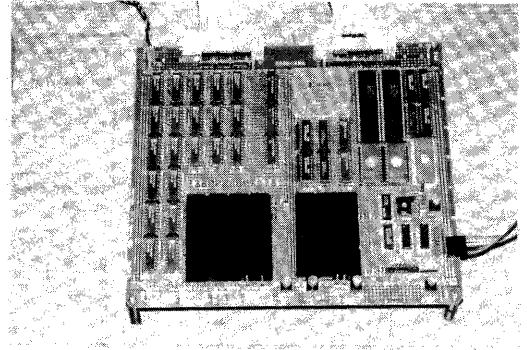


図8 評価用ボード

この評価用ボードを用いるとノイマン型計算機の動作原理を視覚的に理解することができるので、現在、計算機入門教育に活用するための準備を進めている。

## 6. おわりに

本稿では、学生実験における論理設計教育の実験演習テーマならびに計算機教育の教材として開発した教育用マイクロプロセッサ K I T E の開発支援環境、設計仕様、実装、動作検証について述べた。今回の実装は評価用ボードを利用して動作テストを行い、正しく動作することを確認している。また、マイクロプロセッサ設計が実験演習テーマとして実施可能か否かを左右する開発時間については、学生実験の厳しい時間的制約の枠内で実施できることを確認できた。K I T E マイクロプロセッサは内部の動作や状態を外部から逐一観測可能なため、計算機アーキテクチャの教育にも大きな効果が期待できる。

現在、K I T E では OS 等のシステムソフトウェアを作成する際に必要不可欠な「割り込み」や「デュアルモード」が実装されていない。そのため、K I T E に割り込み機能とデュアルモード機能を付加したマイクロプロセッサの設計を行っている。これらを利用することで、学習者に対してより効果的な計算機教育を推進できる。なお、評価用ボードやクロスソフトウェアから成る開発ツールを配布できるよう準備を進めており、教材として広く利用して頂ければ幸いである。

## 謝辞

本研究遂行にあたり、ご支援をいただいた本学有田五次郎教授に感謝の意を表します。また、日頃ご討論いただく本学マイクロ化総合技術センターの久我守弘講師、ならびに情報工学研究院生の小羽田哲宏氏と志々目幸憲氏に謝意を表す。なお、本研究の一部は、本学マイクロ化総合技術センターの

CADシステムを利用して実施したことをここに記す。

## 参考文献

- [1] 情報処理学会： 大学等における情報処理教育のための調査研究報告書 (1991)。
- [2] 末吉ほか： F P G A 専用 C A D による論理回路設計の学生実験事例，電気関係学会九州支部連合大会講演論文集，No.1025 (1991)。
- [3] 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)。
- [4] IEEE Computer Society： Special Issue on Micro Processor in Education，IEEE Micro，Vol.11，No.1 (1991)。
- [5] Xilinx, Inc.： Programmable Gate Array Data Book (1991)。
- [6] Cadence Design Systems, Inc.： Verilog-XL Reference Manual (1991)。
- [7] IEEE, Inc.： IEEE Standard VHDL Language Reference Manual，IEEE Std 1076-1987 (1988)。
- [8] Data I/O Corp.： ABEL-FPGA Design Software User Manual (1991)。
- [9] Synopsys, Inc.： Design Compiler Reference Manual (1991)。
- [10] Exemplar Logic, Inc.： CORE User Manual (1992)。
- [11] 末吉ほか： 書換え可能な L S I を用いた教育用マイクロプロセッサの開発，第 4 3 回情報処理学会全国大会講演論文集，2Q-11 (1991)。
- [12] 田中ほか： 書換え可能な L S I を用いた教育用マイクロプロセッサの設計詳細，第 4 3 回情報処理学会全国大会講演論文集，2Q-12 (1991)。