

教育用マイクロプロセッサ KITE による 設計教育事例 (第2報)

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

教育用マイクロプロセッサ KITE は、書換え可能な FPGA を用いて学習者が自らの手で設計、実装、動作確認を行うことができる教材である。また、計算機システムの入門教育から専門教育までの一貫した情報工学教育を実施できる教材としても KITE マイクロプロセッサを活用し、教育事例に基づいて入門教育や設計教育等に関する基礎データの蓄積を行っている。本稿では、教育用マイクロプロセッサ KITE の現況について報告すると共に各種の設計教育事例について紹介し、これらの教育事例に基づき ASIC 時代にふさわしい実践的で体系的な情報工学向け教育カリキュラムを提示する。

Design Education Examples Using KITE Microprocessor

The Second Report

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

The KITE Microprocessor is a teaching material using reconfigurable FPGAs in which students can design and implement a microprocessor, and can verify its operations by themselves. Furthermore, KITE can be utilized to provide an effective computer science education from introductory until advanced courses in information science. This fact was based on our accumulated data from the offered introductory and microprocessor design courses. This paper reports on the current status of KITE, introduce various development case studies, and propose a curriculum in computer science which is practical, systematic and suitable for the ASIC era.

1 はじめに

大学や高専における学生実験テーマとして、ASICによるマイクロプロセッサの開発実験が注目されている。マイクロプロセッサはデジタル回路の基本的要素をすべて含み、関連する基礎科目も多く、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恰好の題材である。

マイクロプロセッサ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、実験環境として以下の要件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に行え、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。

一方、使用者側でプログラム可能で、構造がゲートアレイに近いFPGA (Field Programmable Gate Array) が高性能化し、ハードウェア教育のための強力な重宝な道具となってきた。SRAM方式のFPGAの場合、構成データ次第で任意の論理回路を実現でき、書き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) 学生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、デジタル回路設計の経験が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を利用することで上記要件をすべて満足できる。

そこで、我々はSRAM方式のFPGAを利用し、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育用マイクロプロセッサ KITEを開発し、情報工学向け教材の開発や設計教育に関する基礎データを蓄積してきた [1], [2], [3]。本稿では、計算機システムの入門教育から専門教育までの一貫した計算機工学教育を実施できる教材として開発した教育用マイクロプロセッサ KITE の現況を報告すると共に各種の設計教育事例について紹介し、これらの教育事例に基づいて情報工学向けカリキュラムを提示する。

2 KITE マイクロプロセッサ

2.1 開発目的

KITE マイクロプロセッサは、図1に示すようにノイマン型計算機の動作原理、設計理論、構成方式、管理技法までの一貫した情報工学教育に利用できる教材の開発を目的としている。

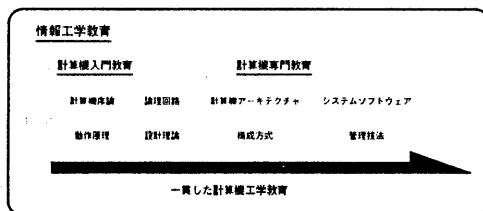


図1: 教育用マイクロプロセッサ KITE の開発目的

情報工学教育の場合、入門教育では学生がノイマン型計算機の内部動作や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生自身がCADシステムを使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オペレーティングシステム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法を学習させ、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て計算機のシステム設計(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、書換え可能なFPGAを利用して、学生が自らの手で設計、実装、動作確認まで行え、かつノイマン型計算機の動作原理から管理技法まで一貫した計算機工学教育を実施できる教育用マイクロプロセッサ KITE の開発を行い、実際に情報工学教育への試行実験を行ってきた。

2.2 基本仕様

現在、教育用マイクロプロセッサ KITE には KITE-1 と KITE-2 という2種類がある [1], [3]。KITE-1 ならびに KITE-2 マイクロプロセッサの概要を表1に示す。KITE-1 は、FPGA による実装という制約下で回路規模等を考慮した、簡素な1チップ構成の16ビットマイクロプロセッサである。一方、KITE-2 は KITE-1 マイクロプロセッサを機能拡張し、システムソフトウェア教育、特にオペレーティングシステムの実装に不可欠な機能を備えたマイクロプロセッサである。

KITE-2 マイクロプロセッサの命令セットを表2に示す。KITE-2 の命令セットでは割込みおよびデュアルモードの実装に伴い、ソフトウェア割込み命令

表 1: KITE マイクロプロセッサの基本仕様

| | |
|-------------------------|------------------------|
| KITE-1 マイクロプロセッサ | |
| • | 1チップ構成の16ビットマイクロプロセッサ |
| • | メモリ空間は4Kワード |
| • | I/O空間は256ワード |
| • | レジスタ構成はアキュムレータ方式 |
| • | 布線論理方式のシーケンス制御 |
| • | 3種類の動作モード |
| KITE-2 マイクロプロセッサ | |
| • | KITE-1 マイクロプロセッサの機能拡張版 |
| • | 割込み機能 |
| • | デュアルモード機能(スーパーバイサ/ユーザ) |
| • | メモリ管理機能(メモリ空間64Kワード) |

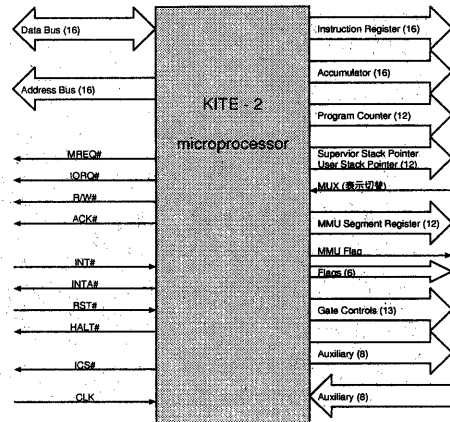


図 2: KITE-2 マイクロプロセッサの端子機能
 入出力端子:42ピン 観測用端子:117ピン

(TRAP), 割込みからの復帰命令 (RTI), 特権転送命令 (PMV) の3命令を KITE-1 の命令セットに追加し, さらに従来の入出力命令 (IN, OUT) と停止命令 (HALT) を特権命令としている。また, アドレッシングモードについても, コンパイラ作成におけるコード生成の容易さを考慮して, インデックス修飾アドレッシングを持つ命令に対してスタックポインタ相対アドレッシングを追加している。

表 2: KITE-2 マイクロプロセッサの命令セット

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

* Privileged Instruction

教育用マイクロプロセッサ KITE の特徴の一つとして, 高い可観測性が挙げられる。KITE-2 マイクロプロセッサの端子機能を図 2 に示す。端子機能は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, プロセッサ内部におけるデータの流れや各種レジスタ値の観測用端子に大別できる。KITE-2 マイクロプロセッサでは 117 本の観測用端子を設けて, KITE-1 マイクロプロセッサと同様に, プロセッサ内部の動作や状態を把握するための可観測性を高くしている。

3 開発支援環境

KITE マイクロプロセッサの開発方法としては, 回路図による設計法とハードウェア記述言語による設計法を選択できる。後者では, 現在のところ VHDL, Verilog HDL, ABEL-HDL 等のハードウェア記述言語を利用可能である。

3.1 KITE マイクロプロセッサボード

KITE-1 および KITE-2 マイクロプロセッサの設計教育に利用できる KITE マイクロプロセッサボード PLUS+ (10,000 ゲート相当の米 Xilinx 社製 XC4010-PG191 搭載) の写真を図 3 に示す。写真からも分かるように, KITE マイクロプロセッサボードはプロセッサの観測用端子の状態を表示するために多数の LED ディスプレイと LED ランプを用意している。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になり, 開発実験における実装時のデバッグにも役立つ。また, 入門教育用ワンボード・マイコンとして利用する際には, ノイマン型計算機の動作原理を効果的に理解できる。

なお, KITE マイクロプロセッサボードは, ホスト IF ボードの SCSI/RS-232 インタフェースを介して, ワークステーションやパーソナルコンピュータ等のホスト計算機と接続できる。これにより, ホスト計算機は KITE マイクロプロセッサのメモリを読み書きでき, 次項で説明するクロスソフトウェア環境を利用して作成したプログラムを KITE マイクロプロセッサボード上のメモリにロードして実行したり, 実行結果をホスト計算機から読み出すことができる。

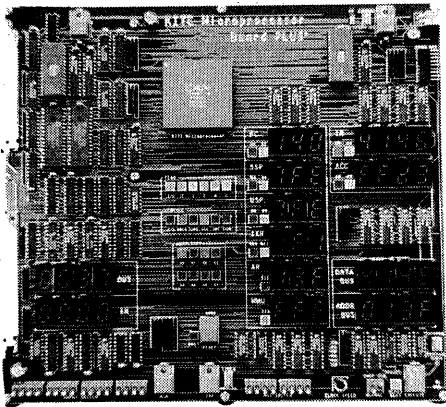


図 3: KITE マイクロプロセッサ PLUS+

3.2 クロスソフトウェア環境

クロスソフトウェアとしては、Cコンパイラ、アセンブラ、逆アセンブラ、シミュレータを用意しており、ホスト計算機上で利用できる。

シミュレータは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、ホスト IF ボードを介してオブジェクトコードを KITE マイクロプロセッサボードの RAM 領域にも転送できるので、設計した KITE マイクロプロセッサの詳細な動作確認に利用される。

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開発実験でのデバッグに役立つだけでなく、アセンブル言語や C 言語によるプログラミング演習を行ったり、さらには自らが設計した KITE マイクロプロセッサのためのオペレーティングシステムや言語処理系の作成演習にも利用できる。

3.3 デバッグフレームワーク

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、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（例えば、ALU）ごとに実装し動作確認を行えることである。そこで、構成要素ごとに必要となる動作確認ならびにデバッグを効率良く行うため、汎用性のある各種デバッグ用回路（デバッグフレームワークと呼ぶ）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を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。

4 設計教育事例

ここでは、本学情報工学科知能情報工学科において実施した、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。

4.1 回路図による KITE-1 の設計事例

従来の論理回路実験の代替テーマとして 1993 年度前学期から暫定的に始めた学生実験であり、回路図による KITE-1 マイクロプロセッサの開発実験を実施している。実験内容の詳細については既に報告しているので、ここではこの 2 年間における学生実験の集計結果を紹介する。

(1) 回路図による実験の概要

学部 3 年生を対象とした学生実験であり、基本的には 2 人一組でマイクロプロセッサの開発実験を遂行する。ただし、一人でマイクロプロセッサすべてを設計したい旨の本人の申し出は、CAD 設備が許す限り要望を受け入れている。

実験スケジュールを図 4 に示す。実験日は週あたり 2 回（午後半日）であり、3 年生を 4 つのグループ（A, B, C, D）に分けて 1 グループ（20~24 人）毎に 3 週間ずつ実施する。つまり、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。

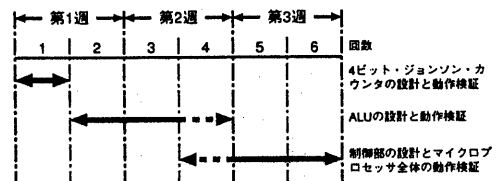


図 4: 実験スケジュール

(2) 回路図による実験の結果

1993 年度および 1994 年度の実施結果を図 5 に示す。別の学生実験の代替テーマとしてのため実験期間が短いにも拘らず、最終的には 1993 年度が 43 組中 32 組（75%）で、1994 年度は 50 組中 45 組（90%）に達している。なお、一人で設計したい旨の要望は、1993 年に 2 名あり共に完成させたが、1994 年は 7 名に増加したが完成は 3 名にとどまった。

また、1993 年度および 1994 年度の実験で完成に至った KITE-1 マイクロプロセッサすべて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 6 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、4,000 ゲート以上の数%を除けば設計サンプルと比べて同等か多少大きい程度のもの

が多く、サンプルよりもむしろ小規模にできたものも少なくないことが実験結果から確認できた。

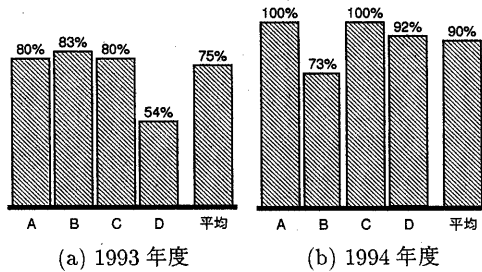


図 5: 1993, 1994 年度の実施結果 (完成率)

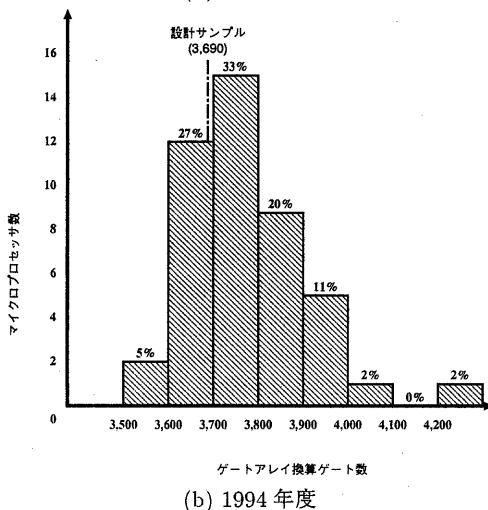
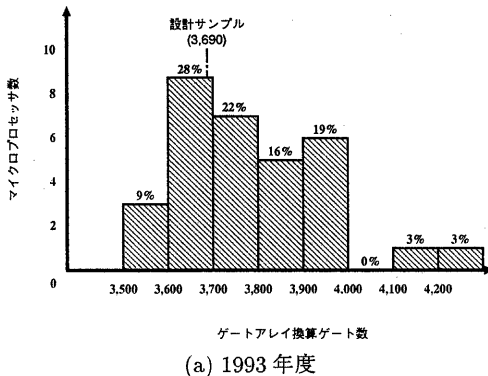


図 6: 1993, 1994 年度の回路規模結果

4.2 HDL による KITE-2 の設計事例

(1) HDL による実験の概要

計算機アーキテクチャのように論理回路設計よりもシステム設計・機能設計に重点をおく設計教育では、今後の LSI 設計において主流になりつつある HDL を用いた RT (Register Transfer) レベルでの設計が効果的であり、開発期間の短縮も期待できる。来年度から実施を予定している HDL によるマイクロプロセッサ開発の予備調査として、KITE-1 マイクロプロセッサに OS に必要な機能を拡張した KITE-2 マイクロプロセッサの VHDL による開発実験を行った。

この実験では、回路図入力による KITE-1 マイクロプロセッサの部分設計を行った経験はあるが、HDL の使用経験のない学生を対象とした。学生には KITE-2 マイクロプロセッサの仕様書のみを与えて、VHDL の学習、CAD ツール操作法の学習の後、KITE-2 マイクロプロセッサの設計・実装を行ってもらった。マイクロプロセッサの構成については特に制限を設けず自由に設計してもらい、回路規模、実装時間、開発期間等を調査した。

(2) HDL による実験の結果

VHDL による KITE-2 マイクロプロセッサの実装結果を表 3 に示す。8 名の学生のうち、3 名が KITE-2 の拡張機能である割込み機能、デュアルモード機能、メモリ管理機能すべてを実装完了したが、メモリ管理機能である MMU (Memory Management Unit) については 5 名が未実装となった。この 5 名はいずれも MMU の設計を完了していたが、実装時の FPGA 配線資源の不足のために MMU を省いたものである。また、ハードウェア量 (ゲート数) にかなり個人差が生じた結果となったが、これはマイクロプロセッサの構成方式の違いよりも VHDL 記述の違いによって、論理合成された回路の規模に差が生じたためである。

HDL による設計の場合、現段階では記述可能な HDL のセマンティクスと論理合成可能なセマンティクスにギャップがあるため、設計者は論理合成を意識した記述を行う必要がある。今回の実験では、VHDL の記述方法について特にガイドラインを設けず行ったため、学生は試行錯誤的に記述を修正しながら設計を行い、記述方法の選択に多くの時間を費やす結果となった。このことから、HDL による開発実験を円滑に進めるためには、記述に関して論理合成・最適化ツールを考慮した適切なガイドラインを設ける必要がある。

表 3: VHDL による KITE-2 の実装結果

| Design Entry | VHDL | | | | | | | |
|---------------------------------------|--------|---------|--------|--------|---------|---------|---------|---------|
| | A | B | C | D | E | F | G | H |
| Bus Organization | 1バス | | 2バス | | 3バス | | | |
| Architecture | KITE-2 | KITE-2' | KITE-2 | KITE-2 | KITE-2' | KITE-2' | KITE-2' | KITE-2' |
| Input XNF Design Statistics | | | | | | | | |
| Number of Logic Symbols | 1707 | 1524 | 1721 | 1670 | 1953 | 1562 | 1563 | 1497 |
| Number of Flip Flops | 209 | 138 | 207 | 220 | 189 | 164 | 124 | 127 |
| Number of 3-State Buffers | 176 | 144 | 176 | 176 | 128 | 176 | 128 | 160 |
| Equivalent Gate Array Gates | 6722 | 5868 | 6584 | 6393 | 6814 | 6276 | 5565 | 5360 |
| Partitioned Design Utilization | | | | | | | | |
| Occupied CLBs [Utility (%)] | 93 | 92 | 97 | 94 | 93 | 85 | 89 | 85 |
| Packed CLBs [Utility (%)] | 80 | 76 | 77 | 75 | 83 | 67 | 74 | 68 |
| Flip Flops [Utility (%)] | 18 | 12 | 18 | 19 | 16 | 14 | 11 | 11 |
| 3-State Buffers [Utility (%)] | 18 | 15 | 18 | 18 | 13 | 18 | 13 | 16 |
| Routing Summary | | | | | | | | |
| Number of Total lines used | 6517 | 6840 | 6768 | 6951 | 6354 | 6430 | 5893 | 6215 |
| Cpu Times on SS10/M41 | | | | | | | | |
| Total (min:sec) | 12:01 | 15:00 | 10:49 | 13:17 | 10:35 | 10:06 | 09:29 | 09:26 |

* without MMIO

4.3 他プロセッサの試作実験

KITE マイクロプロセッサボード PLUS+は、実装デバイスとして FPGA を使用しているため、教育目的に応じて KITE マイクロプロセッサとは異なるアーキテクチャを持つマイクロプロセッサを実装することもできる。そこで今回、そのようなマイクロプロセッサの実装可能性を調査するため、COMET と M32 という 2 種類のマイクロプロセッサの実装を試みた。

(1) COMET の試作実験

COMET とは、情報処理技術者試験に出題される 16 ビット仮想計算機である。COMET マイクロプロセッサの設計仕様は、基本的に情報処理技術者試験受験案内書で定義されている仕様に準拠した。情報処理技術者試験受験案内書で定義されていない制御信号やメモリアクセスタイミングなどは、実装環境を考慮して KITE マイクロプロセッサの仕様に合わせている。また、この COMET には FPGA 実装後に内部状態を外部から容易に把握できるように、レジスタの値や制御信号を観測するための観測用端子を用意している。

今回試作した COMET では、設計手法として回路図入力による方法 (1 バス) と VHDL による方法 (1 バス, 3 バス) を用いた。表 4 に示す実装結果から分かるように、設計手法の違いが回路規模に大きな影響を及ぼしているが、いずれの場合も余裕をもって実装可能であることが確認できた。

(2) M32 の試作実験

M32 とは、岩波講座ソフトウェア科学「計算機システム入門」で紹介されている 32 ビット仮想計算機である [4]。M32 仮想計算機における命令実行時の動作や内部構成は、この文献 [4] で詳細に記述されている。今回試作した M32 マイクロプロセッサの設計仕様については、図 7 に示す内部構成を含め

表 4: COMET の実装結果

| Design Entry | 回路図 | | VHDL |
|---------------------------------------|-------|-------|-------|
| | 1バス | 3バス | |
| Bus Organization | | | |
| Input XNF Design Statistics | | | |
| Number of Logic Symbols | 925 | 1820 | 2104 |
| Number of Flip Flops | 189 | 214 | 183 |
| Number of 3-State Buffers | 144 | 144 | 192 |
| Equivalent Gate Array Gates | 4774 | 7130 | 7778 |
| Partitioned Design Utilization | | | |
| Occupied CLBs [Utility (%)] | 69 | 92 | 98 |
| Packed CLBs [Utility (%)] | 50 | 78 | 77 |
| Flip Flops [Utility (%)] | 16 | 19 | 20 |
| 3-State Buffers [Utility (%)] | 15 | 15 | 16 |
| Routing Summary | | | |
| Number of Total lines used | 4162 | 6398 | 7714 |
| Cpu Times on SS10/M41 | | | |
| Total (min:sec) | 06:59 | 18:31 | 24:33 |

て文献 [4] で定義されている仕様に準拠しているが、以下の点については変更を加えた。つまり、M32 のシフト命令は、最大 32 ビットのシフトが可能であるため、バレルシフタが必要となる。しかし、バレルシフタの実装は回路規模の増大が予想されるので、ここではシフト命令は 1 ビット単位のシフトに限定した。また、M32 では 7 レベルの外部割り込みをサポートしているが、KITE マイクロプロセッサボードの仕様により外部割り込みを 1 レベルとした。

M32 の設計手法としては、回路規模の増大が見込まれたため、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を行った。実装結果を表 5 に示す。試作した M32 の NAND 換算ゲート数は 7,411 ゲートであるが、このうち RAM マクロが 1,600 ゲートを占めている。M32 のようにビット幅が大きいレジスタを多数実装する場合には、RAM マクロによる実現が効果的である。

また、M32 を KITE マイクロプロセッサボード PLUS+ に実装する際の問題は、メモリバス幅とメモリアクセスのための制御信号が異なることである。M32 は 32 ビットマイクロプロセッサである

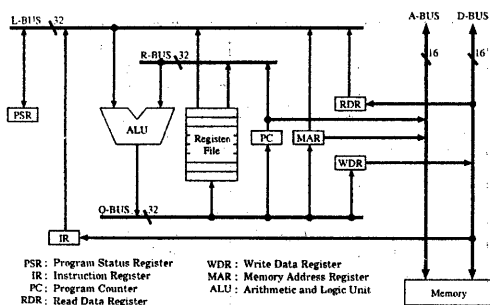


図 7: M32 の内部構成

表 5: M32 の実装結果

| Design Entry | 回路図 |
|--------------------------------|-------|
| Architecture | M32 |
| Input XNF Design Statistics | |
| Number of Logic Symbols | 1323 |
| Number of Flip Flops | 221 |
| Number of 3-State Buffers | 171 |
| Equivalent Gate Array Gates | 7411 |
| Partitioned Design Utilization | |
| Occupied CLBs [Utility (%)] | 96 |
| Packed CLBs [Utility (%)] | 76 |
| Flip Flops [Utility (%)] | 19 |
| 3-State Buffers [Utility (%)] | 17 |
| Routing Summary | |
| Number of Total lines used | 7398 |
| Cpu Times on SS10/M41 | |
| Total (min:sec) | 30:54 |

のに対し、KITE マイクロプロセッサボードのメモリバス幅は 16 ビットであるため、M32 では 2 回のメモリアクセスが必要となる。この問題は M32 マイクロプロセッサ本体以外に、簡単なメモリインタフェース部を FPGA 内に実現することにより解決した。つまり、メモリインタフェース部では、M32 からのメモリアクセス要求を受け付けると 2 回の 16 ビットメモリアクセスを実行してから M32 へ完了信号を返し、32 ビットメモリアクセスを実現する。このようにメモリインタフェース部を提供することで、KITE マイクロプロセッサボードを利用して 32 ビットマイクロプロセッサを無理なく実装できることを確認した。

5 情報工学向けカリキュラム

情報系学科において、KITE マイクロプロセッサを利用して実施できる教育内容としては、

- 計算機入門教育
- 論理回路設計教育
- 計算機アーキテクチャ教育
- システムソフトウェア教育

などがある。一例として、九州工業大学情報工学部知能情報工学科のカリキュラムを紹介する。知能情報工学科では 1993 年度からスタートした新カリキュラムで、入門教育から専門教育まで KITE マイクロプロセッサを積極的に利用し、図 8 に示す計算機工学教育を実施している。

5.1 計算機入門教育

入門教育の実験演習では、最近の複雑な計算機ではなく、高速化技法等の枝葉を切り捨てた、裸の簡素な計算機システムをできることが望ましい。そこで、KITE-1 マイクロプロセッサを利用して、下記の実験演習を行っている。

(1) 動作原理の理解

簡単なプログラムについて、通常動作、命令単位、クロック単位の順序で実行させ、計算機の動作原理を段々と詳細に理解する。

(2) 内部バス構成の違いによる効果

検証済みの構成データ (1 バス構成、3 バス構成) を ROM に実装しており、内部構成が異なるマイクロプロセッサの動作の違いを観る。

(3) アセンブリ言語演習

ホスト計算機と接続し、クロスソフトウェア環境を利用したアセンブリ言語演習を行う。

5.2 論理回路設計教育

回路図入力による KITE マイクロプロセッサの開発実験を行う。実装デバイスとして、書換え可能な FPGA を用いており、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いので、階層設計に従って計算機構成要素毎に実装し動作確認を行う。例えば、週 1 回の実験毎に計算機構成要素である ALU、デコーダ、シーケンサ等を順次作り上げて、最終的にそれらを組み合わせて KITE マイクロプロセッサを完成させる。

1993 年度学生実験では学部生 84 人のうち最終的に約 75% が完成させており、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な効果が認められた。

5.3 計算機アーキテクチャ教育

システム設計・機能設計という計算機の構成方式を重視する教育では、RT レベルによる設計が可能なハードウェア記述言語による開発演習の方が効果的である。そこで、回路図入力による KITE-1 マイクロプロセッサの開発実験後に、ハードウェア記述言語による KITE-2 マイクロプロセッサの開発実験を実施している。KITE-2 は一度設計した KITE

| 学年 | 1 年 | | 2 年 | | 3 年 | |
|-----------------|---|--|---|---|---|-----------------------------|
| KITE 実験演習の内容 | | | 回路図入力による KITEマイクロプロセッサの設計 | | オペレーティング システムの作成 | コンパイラの作成 |
| | 入門教育(動作原理, アセンブリ言語) | | | | | ハードウェア記述 言語による設計 |
| 講義・ 実験演習 | 計算機システムI (講義:1コマ) 論理回路・同演習 (講義+演習:2コマ) | 情報工学 基礎実験I (1回/週) 計算機システムII (講義:1コマ) | 情報工学 基礎実験II (1回/週) デジタルシステム 設計・同演習 (講義+演習:2コマ) | 知能情報工学 実験演習I (1回/週) 計算機 アーキテクチャ (講義:1コマ) | 知能情報工学 実験演習II (1回/週) プログラミング 言語処理系 (講義:1コマ) オペレーティング システム・同演習 (講義+演習:2コマ) | 知能情報工学 実験演習III (2回/週) |

図 8: KITE マイクロプロセッサを利用したカリキュラム例 (関連部分のみ)

マイクロプロセッサの機能拡張版であるため、RT レベルによる機能設計やハードウェア記述言語の理解がいっそう深まる。

なお、1993 年度に試行的に行った RT レベルによる KITE-1 マイクロプロセッサの設計教育事例では、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。情報系学科の学生のようにプログラミングや OS の操作法についての要素があれば、ハードウェア記述言語の使用経験の無い初心者でも比較的短期間に完成できており、高い学習効果を期待できる。

5.4 システムソフトウェア教育

計算機入門教育で使用し、かつ専門教育における学生実験で自分が開発した KITE-1 や KITE-2 マイクロプロセッサを、システムソフトウェア演習のターゲット計算機として利用する。

- (1) 言語処理系 (コンパイラ) 作成演習
クロスソフトウェアを利用して、C コンパイラの作成演習を実施する。既に複数の学生がコンパイラを完成させ、適当な課題である。
- (2) 内部バス構成の違いによる効果
KITE-2 マイクロプロセッサを用いて、マルチプロセスをサポートし、セグメント方式に基づくメモリ管理を行う OS を実装する。

6 おわりに

本稿では、FPGA を利用した教育用マイクロプロセッサ KITE の現況を報告すると共に各種設計教育事例を紹介し、教育事例に基づいて情報工学向

け教育カリキュラムを提示した。現在はさらに、パイプラインを採用した RISC 型の KITE-RISC マイクロプロセッサも開発中である。これら KITE マイクロプロセッサの設計データやソフトウェア等は全て公開する予定であり、本稿で紹介したような ASIC 時代にふさわしい実践的なハードウェア教育を実施される際の参考になれば幸いである。

謝 辞

学生実験の指導にご協力頂いた本学マイクロ化総合技術センターの柴村英智助手、ならびに TA を務めて頂いた院生諸君に感謝します。また、短い実験期間にも拘らず、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。なお、本研究の一部は九州計測器 (株) との共同研究による。

参 考 文 献

- [1] 末吉敏則: 教育への FPGA 応用例, 情報処理, Vol. 35, No.6, pp. 519-529, 1994.
- [2] 末吉, 田中, 久我: 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, 電子情報処理学会技術研究報告 (VLD93-85), 情報処理学会研究報告 (93-ARC-103-12, 93-DA-69-12), 1993.
- [3] 末吉, 小羽田, 野崎, 田中, 久我: FPGA を利用した教育用マイクロプロセッサ KITE-2 システムソフトウェア教育への対応, 情報処理学会研究報告 94-ARC-106-4, 1994.
- [4] 所真理雄: 計算システム入門, 岩波書店, 1991.