The University of Osaka
Institutional Knowledge Archive

Title	CPU設計導入教育への形式的設計検証手法の適用
Author(s)	北浜, 優子; 北嶋, 曉; 岡野, 浩三 他
Citation	情報処理学会論文誌. 2000, 41(11), p. 3114-3121
Version Type	VoR
URL	https://hdl.handle.net/11094/50250
rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

CPU 設計導入教育への形式的設計検証手法の適用

北浜優子[†] 北嶋 暁[†]
岡野浩三[†] 谷口健一[†]

形式的手法の設計導入教育への適用の有用性を調べるために，大阪大学基礎工学部情報科学科 3 年次学生の CPU 設計実験において，(i) 我々の研究グループで作成した検証システムを用いて設計の正しさを形式的に証明する設計手法（新手法），(ii) 慎重な見直しや波形シミュレーションなどで設計の正しさを確認する設計手法（従来手法），の 2 つのコースを設けて，その 2 つのコース間で作業時間と設計した CPU における誤りの有無について 2 カ年にわたって比較を行った．定められた中間レポート提出期限までに誤りのない CPU を設計した学生は，従来手法コースでは 42 人中 0 人，新手法コースでは 42 人中 41 人であった．従来手法コースでは提出期限後に，教官が誤りを指摘して学生が設計を修正する期間を設け，この期間に 19 人が誤りのない CPU を設計した．中間レポート提出期限までに費やした全作業時間の平均は新手法 43 時間に比べて従来手法のほうが 13 時間ほど短かった．以上より，あらかじめ定められた期限までに正しい CPU を設計するためには，新手法は 30%ほど作業時間がかかるものの効果的であることが確かめられた．

Applying Formal Verification Method to Education in Design of CPU

Yuko Kitahama,[†] Akira Kitajima,[†] Kozo Okano[†]
and Kenichi Taniguchi[†]

In order to compare a formal method with a conventional method for designing CPUs, we have measured several metrics (workload and the number of logical errors of the CPUs) for two years in CPU designing laboratory work for undergraduate students in Computer and Information Science at Osaka University. Here, eighty-four students were divided into two groups; Course V and Course T. Each student in Course V used our verifiers to prove the correctness of his/her design, while each student in Course T carefully observed the design or used a waveform simulator. Nobody (out of forty-two students) correctly designed CPUs in Course T by a deadline, while forty-one (out of forty-two) correctly designed CPUs in Course V by the same deadline. The average working time of Course T students elapsed by the deadline is 30 hours and it is 30% shorter than that of Course V students. From the comparison results, we can conclude that the formal method is useful for designing a CPU correctly within a pre-fixed period.

1. まえがき

形式的手法を用いた設計検証は要求仕様に対する設計の正しさを論理的に保証できる有用性があり，これまでに多くの研究がなされている．たとえば，BDD を用いたモデルチェッキング[1),2)]は要求仕様に対する設計の正しさを自動的に保証でき，精力的に研究されている．一方，設計に対する要求仕様を比較的抽象度の高いレベルで記述する場合には，このような BDD を用いたアプローチは適用が難しく，段階的設計法と形式的手法を組み合わせたアプローチなどがなされる．たとえば，CPU などを対象とした段階的設計法[7),8)]や，形式的証明[3)~6)]などがその方法としてあげられる．また，最近，大学などにおいて講義や設計教育などを通して形式的手法を学ぶコースを設けている例も増えてきている（http://www.cs.indiana.edu/formal-methods-education/Courses/）．このような形式的手法は，とりわけハードウェアの設計において有用であると考えられる．日本国内においても，大学などで教育用 FPGA ボードを用いた設計演習がさかんに行われており[9)~11)]，設計導入教育において形式的手法を併用できる下地ができつつある．要求仕様を正確に満たす設計を行っているという意味での設計の正しさを調べるための，形式的手法による自動証明ツールの利用は以下の理由によりきわめて有効な手段と考えられる．

† 大阪大学大学院基礎工学研究科
Graduate School of Engineering Science, Osaka University

(1) 要求仕様に対する設計の論理的な正しさの保証が可能であること．
(2) ツールが自動であるために，設計の初心者であるユーザにとっても使いやすいと思われること．

実際にどれだけ形式的手法が有効であるかを調べるには多くの設計者に対して実際に設計作業に適用し，従来手法との比較を行い，作業時間，設計に要した期間，設計の完成度，など調べる必要がある．そこで，我々の学科の CPU 設計実験（以下「学生実験」）において，(i) 我々の研究グループで作成した検証システムを用いて設計の正しさを形式的に証明する設計手法（新手法），(ii) 慎重な見直しや波形シミュレーションなどで設計の正しさを確認する設計手法（従来手法），の 2 つのコースに分け，作業時間と設計誤りの有無について比較を行った．

学生実験では，教官が定めた命令セットアーキテクチャの主要部を参考に，学生が最終的な命令セットアーキテクチャを定め，さらにそれを満たす CPU を設計し，FPGA ボード上で動作させる．設計は以下のように行う．要求仕様を完全に決定し，それをもとに 1 つの状態機械で表された制御部，機能部品の動作定義，データパスを設計する．ついでそれらの 1 つの状態機械で表された制御部，機能部品の動作定義，データパスが要求仕様を正しく実現していることを確認する．さらに，1 つの状態機械で表された制御部をもとに，複数の状態機械で表された制御部を設計し，その複数の状態機械で表された制御部が先ほどの 1 つの状態機械で表された制御部を正しく実現していることを確認する．最後に，FPGA ボード上で動作させるための作業を行う．設計の正しさを確認する工程において，新手法では，我々の研究グループで作成した自動検証システムを用いて設計の正しさを形式的に証明する．一方，従来手法では，慎重な見直しや波形シミュレーションを用いて設計の正しさを確認する．平成 9 年度の実験に試験的に新手法を導入し，主に作業時間に関して測定し，従来手法と比較したところ，全体として従来手法の作業時間が少ない[13]ものの，作業の進行などに新手法の効果がいくらか現れていることが確認された．そこで，実験の実施方法を改善して，平成 10 年度の実験に新手法を導入し，作業時間に関してさらに詳しく測定を行うとともに，新たに，設計した CPU の設計誤りの有無に関して調べ，従来手法と比較した．

その結果，学生自身が正しさを確認した設計記述を提出する期限（中間レポート提出期限）までに誤りのない CPU を設計した学生は，従来手法コースでは 42 人中 1 人もいなかったのに対し，新手法コースでは 42 人中 41 人であった．従来手法コースでは中間レポート提出期限後に，教官が誤りを指摘して学生が設計を修正する（この過程をフィードバックと呼ぶことにする）期間を設け，この期間に 19 人が誤りのない CPU を設計した．

誤りのない CPU を設計した学生の中間レポート提出時までの全作業時間（残り数回程度のフィードバックを行えば完成する程度までの設計過程に要した時間）約 30 時間に比べ新手法で完成した 41 人の全作業時間は 10 時間程度多かった．

以上より，あらかじめ定められた期限までに正しい CPU を設計するためには，新手法はやや作業時間がかかるものの効果的であることが確かめられた．

以下，2 章では学生実験の概要について，3 章では検証システムについて述べる．4 章では，平成 9 年度および平成 10 年度の学生実験の実施結果について述べる．5 章で結果について考察する．

2. 学生実験の概要

本章では学生実験の概要，工程，および，新手法と従来手法の概要について述べる．

2.1 学生実験で設計する CPU の概要

学生実験では，教官が定めた命令セットアーキテクチャの主要部を参考に，学生が命令セットアーキテクチャを決定し，さらにそれを満たす CPU を設計し，FPGA ボード上で動作させる．ここで，命令セットアーキテクチャは，(1) 最低限の可視レジスタ（プログラムカウンタ，算用レジスタ，フラグレジスタ）を定め，(2) 各命令の実行を抽象的に一状態遷移としてとらえ，各命令の実行後において各可視レジスタの値が命令実行前の可視レジスタの値を用いてどのように定義されるかを等式で与えた記述である．実現を要求している命令はロード，ストア，算術，分岐など 40 個である．汎用レジスタ数は 2 つ，アドレッシングモードは，ロード，ストアにのみ即値・直接・レジスタ間接を使っている．実装に際して，データパスアーキテクチャ，制御部，命令のフォーマットなどについては，FPGA ボード上で実装可能という制約のもとで，自由度がある．また，命令セットの追加（スタック関係，サブルーチン関係など）を認めた．割込み命令の実現は要求していない．

2.2 工程の概要

学生実験で行う各工程は以下のとおりである．

工程 1 命令セットアーキテクチャの決定
工程 2 RT レベルアーキテクチャの設計
工程 3 1 つの状態機械で表した制御部の設計

工程 4 RT レベルの正しさの確認
工程 5 機能部品の設計
工程 6 複数の状態機械で表した制御部の設計
工程 7 制御部論理設計レベルの正しさの確認
工程 8 各部品の接続情報の CAD システムへの入力
工程 9 FPGA ボード上で実装するための付加情報の入力
工程 10 動作確認用の例プログラムの作成
工程 11 FPGA ボード上での動作確認

工程 1 では，最終的な命令セットアーキテクチャを決定する．次に工程 2 では，要求仕様に基づき，レジスタ，ALU などの機能部品の動作定義とデータパス（制御信号線や各機能部品の接続情報）を決定する．工程 3 では，要求仕様をもとに，1 つの状態機械で表した制御部を設計する．工程 4 では，1 つの状態機械で表した制御部と機能部品の動作定義とデータパスが要求仕様を正しく実現していることを確認する．

工程 5 において，ハードウェア記述言語（本学生実験では CAD システムに合わせて Altera 社の AHDL を用いる）で機能部品を実現し，正しさを確認する．

工程 6 では，1 つの状態機械で表した制御部をもとに，複数の状態機械で表した制御部☆を設計する．工程 7 では，複数の状態機械で表した制御部が 1 つの状態機械で表した制御部を正しく実現していることを確認する．

そして工程 8 では，機能部品，制御部，データパスの接続情報を CAD システムへ入力し，必要なら CAD システム上で合成した後，正しく入力されたか確認を行う．工程 9 では，FPGA 用 CPU データを CAD システムで合成するために，FPGA ボード上の入出力端子，ROM，RAM への接続情報を CAD システムへ入力する．工程 10 では，設計した CPU 上で動作させる動作確認用の例プログラム（電卓プログラム）を作成する．最後に工程 11 では，FPGA ボードに CPU 設計データをダウンロードし，FPGA ボードの ROM に動作確認用の例プログラムを載せ，動作確認を行う．

2.3 新手法と従来手法の違い

図 1 は従来手法と新手法による CPU 設計の概要の違いを表している．従来手法では，工程 4 における設計の正しさの確認は，設計者が慎重に設計を見直すことによって行う．また，工程 7 における設計の正しさの確認は，波形シミュレーションによって行う．一方，形式的手法を用いた新手法では，工程 4，工程 7 において我々が作成した自動検証システム[13]を用いて形式的自動検証を行う．自動検証システムについては次章で述べる．

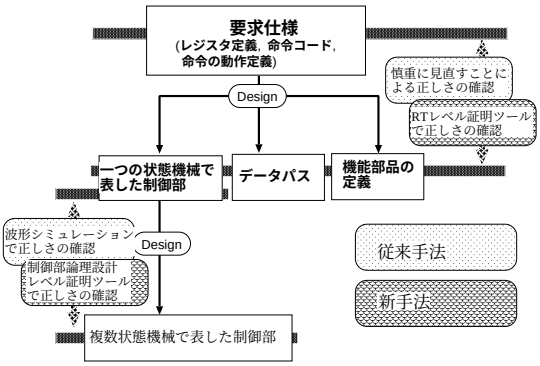

図 1 CPU 設計の概要
Fig. 1 Overview of CPU design.

3. 自動検証システム

自動検証システムは，RT レベル証明ツールと制御部論理設計レベル証明ツールとからなる．各ツールとも，入力となる記述を与えれば自動で設計が正しいかどうかを判定する．各記述に用いる言語は，ツール独自のものではあるが，構文は AHDL に類似したものであり，通常のハードウェア記述言語と本質的な違いはない．以下，2 つのツールについて述べる．

3.1 RT レベル証明ツール

3.1.1 ツールの概要

RT レベル証明ツールでは，1 つの状態機械で表した制御部，機能部品の動作定義，データパスが要求仕様を正しく実現していることの証明を行う．要求仕様の記述では，各命令ごとに，(i) バイナリ表現，(ii) 命令実行前のレジスタ値を用いて表した命令実行後のレジスタ値を表す式，を記述する．データパスの記述では，機能部品の接続関係などを記述する．機能部品の動作定義の記述では，制御信号線の値による機能部品の入出力値の関係を記述する．1 つの状態機械で表した制御部の記述は，各遷移について，実行条件，次の状態，各制御信号線の出力値を記述する．

自動検証のために，証明が可能である CPU のクラスに関して，いくつかの制約を設けているが，本学生実験で設計するような CPU のほとんどはこれらの制約を満たしている．制約は次のとおりである．ハザードはないものと仮定している．また，リセット動作が正しく行われることの証明は対象としていない．また，

☆ 制御部から生成された制御信号にハザードが発生させないための 1 つの方法として，状態遷移においてたかだか 1 つの FF しか値が変化をしない状態機械（本学生実験では，このような状態機械としてジョンソンカウンタを用いる）を複数用いて制御部を実現する．

CPU を停止する命令の正しさの証明も，通常の命令とは異なるため，対象としていない．なお，本ツールはパイプライン方式の命令実行制御は扱えないが，命令の完了前に次の命令コードの読み出し動作のみを行う（命令のプリフェッチ）方式は扱える．

3.1.2 ツールで調べる内容

設計した回路でどの命令をどの順に実行しても，その実行結果が要求仕様のとおりであることを本ツールで調べる．本ツールではその正しさの十分条件を調べており，その十分条件は，以下の各条件が要求仕様で定義された各命令すべてについて成り立つことである．

(1) 制御部の状態遷移図上で，その命令の実行を表す遷移系列（以降パスと呼ぶ）が少なくとも 1 つ存在し，かつ，要求仕様の実行条件のもとでそれらのパスを開始するための実行条件は 1 つだけが真となること．また，そのパスを実行後次の命令が実行可能になること（次の命令をプリフェッチしてもよい）．

(2) その命令を実行するパスすべてについて，そのパスを実行したとき，各レジスタ値が要求どおりに変化すること．

RT レベル証明ツールでは証明失敗時に，次の情報を使用者に提示する．(1) 制御部のある状態から複数の状態へ遷移しうる場合，あるいはある状態から次の遷移が定義されていない場合，その状態までの実行遷移系列と，そのときの命令名対応がとれなかったレジスタ名とそのときの実行命令．(2) 命令の実行完了時に，レジスタ値の変化が命令セットアーキテクチャで定義されいていたものと異なる場合，その命令名とそのレジスタ名，実行遷移系列，および，各状態での当該レジスタの値を表す式．

3.2 制御部論理設計レベル証明ツール

3.2.1 ツールの概要

制御部論理設計レベル証明ツールでは，複数の状態機械で表した制御部が 1 つの状態機械で表した制御部を正しく実現していることの証明を行う．本ツールへの入力としては，1 つの状態機械で表した制御部の記述（RT レベル証明ツールで使用したもの）と複数の状態機械で表した制御部の記述（論理設計レベル）を与える．後者は AHDL のサブセットで記述する．

3.2.2 ツールで調べる内容

本ツールで証明する正しさとは，1 つの状態機械と複数の状態機械とを，初期状態から任意の同じ入力を与えて動作させた場合に，いずれの時点でも同じ出力値であることである．実際に本ツールではその十分条件を調べている．具体的には，1 つの状態機械で表した制御部において，初期状態から実行可能なすべてのパスおよび途中の各状態に対し，複数の状態機械で表した制御部でも対応するパスおよび各状態機械の状態組がそれぞれ 1 つだけ存在し，かつ，対応する遷移それぞれについて，出力値が一致することである．

制御部論理設計レベル証明ツールでは証明失敗時に次の情報を使用者に提示する．(1) 対応がとれなかった実行条件とそれが生じた実行遷移系列．(2) 上位下位の記述で値が一致しなかった信号名とそれが生じた実行遷移系列，および，系列中の各状態での各信号線の出力値．

4. 学生実験の実施

本章では，平成 9 年度と平成 10 年度の学生実験の実施方法，結果などについて述べる．

4.1 平成 9 年度における学生実験の実施

4.1.1 実験の実施方法

平成 9 年度の実施では，形式的手法を用いた新手法を試験的に導入し，作業時間，新手法の導入に関する問題点の有無を調べた．学生全体を新手法コース（14 グループ）と従来手法コース（14 グループ）とに分け，3，4 人のグループで 1 つの CPU を設計した．コースおよびグループの構成は無作為に行った．新手法コースでは，設計の正しさを確認する工程（工程 4 および工程 7）において自動検証システムを使用し，検証に成功するまで原則として次工程に進まないものとした．

4.1.2 測定項目とその測定方法

平成 9 年度の実施においては，学生の作業時間を測定し，新手法コースと従来手法コースとの違いを，主にその作業時間に基づき評価した．データの収集は，学生からの電子メールでの報告の形で行った．

4.1.3 結果とその考察

グループごとの作業項目別の作業時間に関して，コース別に平均をまとめ[13]，次のような結果が得られた．

(i) 設計の正しさを確認する工程 4 において従来手法コースの方が作業時間が短い，また，同じく工程 7 においてはほぼ同じという結果になった．また，(ii) 作業時間の極端に長いグループ，短いグループがあった．(i) の工程 4 に関して新手法コースが長くなった要因として，まず，従来手法コースについては，完全に正しい CPU が作成されている保証がないことがあげられる（4.2.1 項参照）．ほかには，新手法コースは検証システムの使用法や入力の記述およびエラーメッセージの理解に時間がかかったことが考えられる．一方，工程 7 においては，工程 4 と同様に時間がかかってしまう要因はあったが，検証システムの効果が現れ，

作業時間はほぼ同じという結果になっているものと考えられる．また，(ii) の原因としては，作業時間に関しては，設計手法の違いによる差以上に，グループとしての作業時間の認識の違いや報告の不正確さ，各個人の能力の差や各グループの協調性の差が影響していることが考えられる．また，設計とは直接関係のない作業（たとえば，検証システムや波形シミュレーションを含む CAD システムなどの使い方の習得や工程 8 以降の FPGA ボード固有の作業など）に時間をかけていることも原因として考えられる．

4.2 平成 10 年度における学生実験の実施

4.2.1 実験の実施方法

平成 10 年度の実施では，学生全体（84 人）を，新手法コース（42 人）と従来手法コース（42 人）とに分け，各個人で 1 つの CPU を設計することにし，2 つのコースでの作業時間と設計品質の違いを比較した．コース分けは学生の希望を優先して行った☆．

平成 9 年度に形式的手法を用いた新手法を試験的に導入した結果をもとに，以下の (1)〜(3) のように実施方法を改良した．

(1) 設計に直接関係する時間だけを測定する．
(2) 作業時間の報告の方法を改善する．
(3) 設計した CPU に論理的誤りが含まれているか否かについて調べる．

上記 (1) に関しては，設計に直接関係ない前述のような作業を学生が行うことは学生実験という性質上，避けられないが，測定データとしてはこれらの作業項目を分離すべきである．

そこで，工程 7 までの作業時間を 1 つのまとまった課題とし，工程 8 以後の作業を別課題として分離した．また，工程 1 から工程 7 までの作業はすべて個人で行うものとした．

誤りのない複数の状態機械で表した制御部を完成するまでは工程 8 には進まないことになっている．また，機能部品の設計および記述は教官側で行い，学生はその記述ファイルを利用するものとしたため，工程 5 も測定の対象外とした．ただし，証明ツールの習得に必要と思われる時間はあえて作業時間に入れている．

(2) に関しては，平成 9 年度の実施において，グループによっては，各作業項目とそれに要した時間が工程別に区分して報告されていない，あるいは，報告もれがある，などの問題があった．

そこで，学生が平成 9 年度の方法よりも簡単かつ正確に作業時間を報告できるように，WWW ブラウザを利用した作業記録システム（4.2.2 項参照）を作成し，これを用いて報告を行った．

最後に (3) に関しては，平成 9 年度の実施では，設計した CPU が正しく動作することの確認は，動作確認用の例プログラムが正しく動作するどうかにより行った．この方式では，論理的に CPU が正しいことを保証できない．

そこで，動作確認用の例プログラムによる動作試験をせず，従来手法コースの学生が設計した制御部にも教官が検証ツールによる正しさの確認を導入し，設計誤りが含まれているか否かを調べた（4.2.2 項参照）．

4.2.2 測定方法

作業時間については，作業記録システムを用いて報告をさせた．学生が WWW ブラウザから作業記録システムにユーザ名とパスワードを入力すると，学生がその時点で行うべき作業が表示される．作業を開始する場合は「開始」ボタンを押し，中断するときは「中断」ボタンを押す．また，現在行っている作業項目が完了したら「完了」ボタンを押す．これらの操作によって，作業項目別の作業時間が自動的に記録される．

本作業記録システムにおいては，(i) WWW ブラウザを使った作業記録システムの画面に学生がその時点で行うべき作業項目などの情報を表示し，実験作業を支援する，(ii) 学生全員の設計作業進捗状況を WWW ブラウザ上に表示するシステムを作成し，他の学生と比べて自分は進んでいるのか，遅れているのか，ということを学生が把握できるようにする，などの工夫を行った．

従来手法コースの学生が設計した CPU の論理誤りの数については，教官側で検証ツールを適用することによって誤りがないかどうかを調べた．具体的には，平成 10 年度の実験では以下の手順をとった．

「中間レポート提出期限」を設け，両コースとも，学生はその日までに設計記述を教官に提出するものとした．新手法コースについては，制御部論理設計レベル証明ツールで証明が成功するれば提出可能である．従来手法コースについては，波形シミュレーションなどの結果をもとに，設計者が「誤りのない複数の状態機械で表した制御部が完成した」と確信すれば提出可能である．従来手法コースの学生は決められた書式で設計記述を提出し，それを教官側で検証システムの入力記述に変換するツールを用いて検証システムの入力記述に変換し，検証ツールを適用する．

また，従来手法コースにおいて，教官側で検証システムを適用し，証明に失敗した場合は，設計誤りを学

☆ 本実験に最も関連が強いと考えられる「論理設計」の授業の成績を比較して 2 つのコースの成績分布に差違はなかったことより，2 つのコースで学生の能力差はないと考えられる．

生に通知し学生はそれを参考にして設計誤りを修正するものとした．提出日から約 1 カ月間，設計の誤りを修正する期間（フィードバック期間）を設けた．その期間中に，設計者が「誤りのない複数の状態機械で表した制御部が完成した」と判断した時点で，設計記述を再び教官に提出し，教官側で証明に成功するまでこの作業を繰り返す，という形で学生の誤り訂正作業を支援した．証明に成功するまでのフィードバック回数，作業時間，経過した日数なども調べた．

両コースとも，原則としてすべての設計が終了し，定められた実験時間が終了した時点で学生は最終レポートを提出している．

4.2.3 測定結果

実験の実施により以下の測定データを得た．

4.2.3.1 誤りのない制御部を完成させた人数（中間レポート提出日前後別）

新手法コースについては，制御部論理設計レベル証明ツールで証明が成功した人数，従来手法コースについては，教官側で制御部論理設計レベル証明ツールを適用して証明が成功した人数を**表 1** に示す．

4.2.3.2 設計誤りについて

新手法コースでは，提出日までに 42 人中 41 人が設計誤りのない複数の状態機械で表した制御部を設計した（制御部論理設計レベル証明ツールで証明に成功した）．

従来手法において，中間レポート提出日の時点で設計者が「誤りのない複数の状態機械で表した制御部が完成した」と確信していた学生はアンケート調査の結果，21 人いた．しかし，教官側で設計の正しさを調べたところ，この全員について，設計誤りのない 1 つの状態機械で表した制御部，複数の状態機械で表した制御部ともに誤りが発見された．

従来手法コースに関しては，中間レポート提出日の後にフィードバック期間を設けたが，その期間中に誤りのない複数の状態機械で表した制御部を完成させた学生は 19 人，誤りのない 1 つの状態機械で表した制御部までを完成させた学生は 6 人であった☆．

4.2.3.3 作業時間（作業項目別）

作業項目別の作業時間を集計し，コース別に平均を調べた．従来手法コースについては，最終レポート提出時に誤りのない制御部を完成させた学生のデータを**表 2** に示す☆☆．工程 8 以降のデータは省略している．

表 1 誤りのない制御部を完成させた人数（中間レポート提出日前後別）
Table 1 The number of students who completely designed correct CPUs before and after the deadline.

コース（全人数）	提出日前	提出日後	合計
新手法コース（42 人）	41	–	41
従来手法コース（42 人）	0	19	19

表 2 より，最終レポート提出日までに最終的に誤りのない制御部を設計した学生について，中間レポート提出日までの全作業時間を単純には比較できないが，あえて比較すると，新手法コースの作業時間は従来手法コースより 13 時間多い．

5. 考　　察

作業時間と設計誤りに関して考察する．

5.1 作業時間について

表 2 では，各工程別でも提出日までの総作業時間でも，従来手法コースの方が作業に要している時間は短い．正しさの確認に要した時間のみを比較するとその差は 5 時間程度である．新手法では証明ツールの習得の時間も算入されていることも考慮に入れればその差は総作業時間に比べ 17%と小さくなる．

最終的に設計誤りをなくした学生がこれらの過程の中で実際に再設計やバグ取りなどの作業に要した時間の平均は 6.5 時間である（教官が可能な限り誤り内容を指摘しているのでこの期間の作業時間は純粋にテストに波形シミュレーションのみを用いた手法の学生個人の作業時間とは見なせないことに注意）．もし，教官からの情報提供なしに，学生が自力で波形シミュレーションを行って正しさの確認を行うとすると，その作業は 6.5 時間よりも多いと予想される．また，波形シミュレーションの代わりに，FPGA 上でテストを行ったとしてもやはり，多くの時間を要するものと予想される．なお，証明ツールの利用により誤りの箇所を即時に知ることができる．この点で証明ツールはチェック用の波形をそのつど考案しなければならない波形シミュレーションに比べ利用しやすく，結果的に使用時間の増加を招いたとも考えられる．実際，証明ツールの使用時間が多い一方で，同一期間内で従来手法に比

☆ 平年は，基本的にすべての設計が完了した時点でレポートを受け付けて実験の終了と見なしている．平成 10 年度は実験室の計算機入れ替えなどの事情によりフィードバック期間を十分に長く設けることができず，最後まで設計できなかった学生が生じた．教育的配慮から，彼らについては実験室の利用を前提としない追加課題を課すなどして対処した．

☆☆ 最終的に誤りのない制御部を完成させた学生は 19 人であったが，報告が不完全であった学生のデータを除いたため，有効データ数は 18 人とした．最終的に誤りのない制御部を完成させなかった学生のデータについては学生の取り組みによって非常に大きな差があるので集計データから除くことにした．

表 2 作業時間の平均（作業項目別）
Table 2 Averages of working times of tasks.

コース（有効データ数）	作業項目							
	工程 1	工程 2	工程 3	工程 4	工程 5	工程 6	工程 7	合計
新手法コース（38 人）		11:20	5:54	7:01	–	9:49	9:00	43:06
従来手法コース（提出日前）（18 人）		7:07	4:50	2:01	–	7:34	7:43	29:15
従来手法コース（提出日後）（18 人）		0:02		3:14	–		3:17	6:34

– 作業項目 –
工程 1：要求仕様の決定，工程 2：RT レベルアーキテクチャの設計，工程 3：1 つの状態機械で表した制御部の設計，
工程 4：RT レベルの正しさの確認，工程 5：機能部品の設計と正しさの確認，工程 6：複数の状態機械で表した制御部の設計，
工程 7：制御部論理設計レベルの正しさの確認

べ，多くの設計が正しさを保証できていたことはその証左になる．

5.2 設計誤りについて

従来手法コースでフィードバック期間中に，論理的に誤りのない複数の状態機械で表した制御部を完成させた学生の初回提出時における設計の正しさの程度は，平均 4.7 回もフィードバックを行ってすべてなくなった程度であった．平成 9 年度および平成 10 年度の実施においては，テストすべき内容についてはある程度指導しているものの，シミュレーションで用いるテスト波形などは教官側から与えられていない☆．このことや，作業時間，誤りの程度などから考えると，シミュレーションが不十分であったと考えられるし，実際そのような学生が多く見られた．90%の学生は各命令のレジスタ変更の正しさのテストを行っていたが，命令実行後に制御部が次命令のフェッチのための始状態に戻っていることを何らかの手段で確認している学生は全体の 40%程度であった．そのためかなりの数の設計に制御部の実行制御に関する誤りが発見された．また，教官側で行った RT レベル証明ツールでの証明に成功しなかった設計は成功した設計と比較して，従来手法の工程 4 において時間を費やしていないなどということが分かった．**表 3** に提出時における命令グループ別の設計誤り率を示す．表 3 より初回提出時，ロード系，ストア系の命令に比べ，算術系，フラグ系，分岐系などの命令が正しく実現されていないことが多いということも分かった．これらのことは，フラグレジスタのチェックが汎用レジスタのチェックに比べて軽視されがちであることや，算術命令のように複数のモードのチェックを必要とするものでは，代表的なモードのみのチェックしか行われず，結果として，いくつかの論理誤りが見落とされて残ったままであるという傾向を表していると考えられる．

☆ 学生実験ということもあり教育的観点から細かく指導していない．実験結果から，設計初心者はテストでは十分場合を尽すことができないことが分かる．実際の設計では，ウォークスルーを行ったり，一定の指針に従ってテストを行うので従来手法であっても誤り率は改善されることが期待できる．

表 3 提出時における命令グループ別の設計誤り率
Table 3 The ratio of designs with logical errors classified by instruction categories.

ロード系	ストア系	算術系	フラグ系	分岐系
33	34	66	39	50

従来手法コースで最終的に誤りのない制御部を完成させた学生の初回提出時の設計記述における，設計誤りの存在確率.（単位%）

一方，制御部論理設計レベル証明ツールでの証明に最終的に成功しなかった学生は，成功した学生と比較して，テストの実施に関しては大差がなかったことから，制御部論理設計レベルのチェックは RT レベルのチェックに比べて複雑であるため，しっかりチェックしたつもりでも論理的誤りを含む場合があることが分かった．

6. あとがき

学部の学生実験において，形式的手法を用いた新手法を導入し，作業時間，設計誤りに関して，従来手法と比較を行った．その結果，誤りのない制御部を設計するという点で，新手法が有用であることが確かめられた．

検証ツールが提供する設計誤り情報の改善が今後の課題の 1 つである．

謝辞 ご討論いただいた東野輝夫教授，および，データの収集と集計に協力いただいた網浜貴夫君に感謝します．CAD システムは Altera 社の University Program に負う．

参考文献

1) Clarke, E.M. and Kurshan, R.P.: Computer-aided verification, *IEEE Spectrum*, Vol.33, pp.61–67 (1996).
2) Bryant, R.E.: Graph-based algorithms for boolean function manipulation, *IEEE Trans.*

Comput., Vol.C-35, pp.677–691 (1986).
3) Jones, R.B., Dill, D.L. and Burch, J.R.: Efficient Validity Checking for Processor Verification, *Proc. IEEE/ACM International Conference on Computer-Aided Design* (*ICCAD*), pp.2–6 (1995).
4) Burch, J.R.: Techniques for Verifying Superscalar Microprocessors, *Proc. 33rd Design Automation Conference* (*DAC*), pp.552–557 (1996).
5) Park, D.Y.W., Skakkebæk, J.U. and Dill, D.L.: Static Analysis to Identify Invariants in RSML Specifications, *Formal Techniques in Real-Time and Fault Tolerant Systems* (1998).
6) 北嶋　暁，森岡澄夫，島谷　肇，東野輝夫，谷口健一：代数的手法を用いた CPU KUE–CHIP2 の段階的設計の正しさの自動証明，信学論，Vol.J79-D-I, No.12, pp.1017–1029 (1996).
7) 谷口健一，北道淳司：代数的手法による仕様記述と設計及び検証，情報処理，Vol.35, No.8, pp.742–750 (1994).
8) Kitamichi, J., Morioka, S., Higashino, T. and Taniguchi, K.: Automatic Correctness Proof of the Implementation of Synchronous Sequential Circuits Using an Algebraic Approach, *Proc. 2nd Int. Conf. on Theorem Provers in Circuit Design* (*TPCD94*), LNCS, Vol.901, pp.165–184 (1994).
9) 中垣憲一，井上弘士，久我守弘，末吉敏則：上級コース向け教育用マイクロプロセッサ DLX-FPGA の設計と実装，信学技報，CPSY94-57, pp.17–24 (1994).
10) 末吉敏則，井上弘士，奥村　勝，久我守弘：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，*Proc. 1995 Japan FPGA/PLD Conference*, pp.579–588 (1995).
11) Ochi, H., Kamidoi, Y. and Kawabata, H.: ASA ver.1: An FPGA-Based Education Board for Computer, Architecture/System Design, *IEICE Trans. Fundamentals*, Vol.E80-A, No.10, pp.1826–1833 (1997).
12) 木村真也，鹿股昭雄：命令の追加･変更可能な教育用コンピュータシステムの開発，信学論，Vol.J81-D-I, No.12, pp.1241–1248 (1998).
13) 北浜優子，北嶋　暁，岡野浩三，東野輝夫，谷口健一：CPU の高位設計の自動検証システムの作成と学生実験への適用，情報処理学会 DA シンポジウム'98 論文集，pp.101–106 (1998).

(平成 12 年 5 月 12 日受付)
(平成 12 年 9 月 7 日採録)

北濱　優子
平成 9 年大阪大学基礎工学部情報工学科を中退し，同大学院博士前期課程入学．現在，博士後期課程在学中．ハードウェアの形式的検証等に興味を持つ．

北嶋　　暁（正会員）
平成 5 年大阪大学基礎工学部情報工学科中退．平成 10 年同大学院博士後期課程修了．博士（工学）．同年同大学助手．現在に至る．VLSI の上流設計における形式的検証や回路合成等に関する研究に従事．

岡野　浩三（正会員）
平成 2 年大阪大学基礎工学部情報工学科卒業．平成 5 年同大学院博士後期課程中退．同年同大学助手．現在，同大学院基礎工学研究科講師．博士（工学）．ハードウェア・ソフトウェアの形式的仕様記述と検証，分散システム等の研究に従事．

谷口　健一（正会員）
昭和 40 年大阪大学工学部電子工学科卒業．昭和 45 年同大学院基礎工学研究科博士課程修了．工学博士．同年同大学助手，現在，同大学院基礎工学研究科教授．この間，計算理論，ソフトウェアやハードウェアの仕様記述・実現・検証の代数的手法および支援システム，関数型言語の処理系，分散システムや通信プロトコルの設計・検証法等に関する研究に従事．