設計自動化 85－15
(1997. 10. 29)

完全なインターロックを行なうパイプライン CISC/RISC の設計教育

～マイクロコンピュータ設計教育環境 City-1 の 2 年目～

高橋 隆一，吉田 典可

広島市立大学 情報科学部 情報工学科

〒 731-31 広島市安佐南区大塚東 3-4-1

ryuichi@ce.hiroshima-cu.ac.jp, yoshida@ce.hiroshima-cu.ac.jp

あらまし　City-1 ではクイックソートなどの適当なアプリケーションを指定して，「これが実行可能であること」以上の仕様を課していない．学生一人一人が，命令セットアーキテクチャはもちろん，ハードウエア構成やバス構成を自由に定めて課題に取り組む．制御信号発生のための符号化に際しては，結線論理制御方式とマイクロプログラム制御方式を同一視させている．平成 9 年度は，半クロックで命令をフェッチし，次の半クロックでデコードを行ない，次の 1 クロックで実行を行なう，2 相のクロック信号を用いた，3 段パイプライン CISC の不完全な動作記述を与え，「完全なインターロックを行なう」というパイプライン設計の基本を指導した．

キーワード　**マイクロプログラム制御方式, 結線論理制御方式, パイプライン, インターロック,CAD, 設計教育**

An Educational Course on Fully Interlocked Pipeline CISC/RISC Design

～ 2nd report of Microcomputer Design Educational Environment City-1 ～

Ryuichi TAKAHASHI, Noriyoshi YOSHIDA

Faculty of Information Sciences
Hiroshima City Univ.
3-4-1 Ozukahigashi, Asaminami-ku
Hiroshima City, 731-31 Japan

Abstract　In City-1, the specification is given only by showing some application programs that should run on their machines. Every junior student designs not only his/her own instruction set but also his/her own hardware organization and his/her own bus architecture. Hardwired-controls and microprogrammed-controls are treated equally in a sense that the same code can be used for both of these controls. An incompletely described three stages pipelined CISC specification was given, as an example, with three-steps-operation using two-phase-clock-signals; a half clock cycle for the fetch, another half clock cycle for the decode and one clock cycle for the execution, emphasizing that the fully-interlocked-pipeline-design is the basics.

key words　**microprogrammed-control, hardwired-control, pipeline interlock, computer aided design, design education**

1 はじめに

マイクロコンピュータ設計教育環境 City-1[13] は広島市立大学情報科学部情報工学科 3 年前期の学生実験, 情報工学実験 III における, FPGA コンピュータの自由な設計製作教育環境である. クイックソートなどの適当なアプリケーションを指定して, 「これが実行可能であること」以上の仕様は課していない. 学生はひとり 1 台, 自分なりの命令セットアーキテクチャ, ハードウエア構成やバス構成を自由に定め, さらに 2 人 1 組になって, これら 2 人のコンピュータのどちらをも実現できるスーパーセットのボードを製作する. CPU は FPGA で実現される. 各自, それぞれの設計データを FPGA に送り込み, 指定した課題のプログラムが動くことを確認して, いわば商品企画から出荷検査にいたるまでの全工程を体験する. ほとんどのレポートが, 「デバッグ時は, なぜ動かないのかという思いで悔しかったが, それだけに, 動いたときは, たとえようのない喜びを味わった」に類する感想で結ばれる.

制御信号発生のための符号化に際しては, 結線論理制御方式とマイクロプログラム制御方式を同一視させている. 初年度も 2 年目の今年も, CISC と RISC のいずれをも結線論理制御方式で実現させてきた. パイプライン化していない場合の CISC の記述例として学生に提供する記述は, 結線論理制御方式を, あたかもマイクロプログラム制御方式であるかのように表現する手法とみることができる [13].

初年度の平成 8 年度の場合は, 履修登録した 54 人のうちの 50 人が, 割算ができるという意味で, 完全なコンピュータを完成させた. 必要最小限の I/O に関する事項を指導することで製作の手間を軽減した結果, 約半数の学生がクイックソートを走らせるまでの達成度を示し, 3 人は 2 ～ 4 段パイプラインの RISC プロセッサを稼働させた [13].

2 年目の平成 9 年度は, 電源を既配線とする汎用のボードを用意し, 課題をクイックソートからユークリッドの互除法に代えて負担を軽減する一方で, パイプライン設計の基本を徹底的に学習するよう配慮した. 情報工学実験 III は学生ひとりひとりが独自のコンピュータを設計し, 製作することを通して, コンピュータの仕組みと設計方法の基本を学ぶことが目的である. 近年, ときにコンパイラとハードウエアの協調設計という観点からも, 「遅延分岐」や「遅延ロード」など, ハードウエアが本来備えるべき機能をコンパイラに頼ることで削減するという設計が見受けられるようになった. しかし, このような設計を行なう以前に, あるいはこのような設計が行なわれるならなおさら, 完全なインターロックを徹底的に理解することが基本であり不可欠であると筆者らは考えた. 学生に示す設計の具体例としては, 2 相のクロック信号を用いる 3 段パイプライン CISC の不完全な動作記述を工夫した.

このような指導の結果, 履修登録した 54 人のうちの 47 人が, 課題として指定したユークリッドの互除法を走らせるまでの達成度を示した. 47 人のうち 28 人は 3 ～ 4 段パイプライン RISC を完成させ, 19 人が 3 ～ 4 段パイプライン CISC を完成させた. 履修登録した 54 人の全員が, 完全なインターロック機構を搭載した 3 段以上のパイプラインコンピュータの設計と製作を行なった.

本稿では, RISC を, (1) 固定フォーマットの命令セットをもち, (2) 演算のオペランドとしてメモリ上のデータを指定することがなく, (3) 1 クロックあたり 1 命令のスループットを有するコンピュータの意味に用いる.

2 設計教育環境 City-1 の概要

マイクロコンピュータ設計教育環境 City-1 は広島市立大学情報科学部情報工学科 3 年前期の情報工学実験 III における, FPGA コンピュータの自由な設計製作教育環境である [13].

2.1 履修の背景

1 年後期の「論理数学」,2 年前期の「論理回路」によって, 論理回路設計のための基本的な知識はすでに有していることを前提にできる [6],[7], [8]. 2 年前期の「情報工学実験 I」では, 既成のマイコンによってアセンブラを学び, カウンタ程度の簡単な回路の設計と製作を経験済みである. 2 年後期の「情報工学実験 II」では, スタックマシンを前提とする, yacc

とlexを用いたコンパイラの製作を履修済みである.

3年前期に並行して履修する「論理シミュレーション」では, City-1 (情報工学実験III) で使用する超高速論理シミュレータやロジックシンセサイザを具体例として, CADを用いたディジタルシステムの構築技術を指導して情報工学実験IIIを支えている.

2.2 履修の目的

1人1台, 思い思いのコンピュータを設計し, 製作し, 課題として与えられたプログラムを, 自らの機械語で書き下し, 自らのコンピュータ上で走らせることによって, 「コンピュータの仕組みと設計方法の基本を学ぶ」ことが目的である.

結果的に, HDLによる動作記述を出発点とするトップダウン設計に習熟するが, そのこと自体は目的でない.

2.3 指導の方針

City-1 (情報工学実験III) は次のような点が特徴的である:

(i) クイックソートなどの適当なアプリケーションを指定して, 「これが実行可能であること」以上の仕様を課していないこと.

(ii) レジスタトランスファレベルでの動作記述を出発点として, ロジックシンセサイザ, 自動配置配線を用いるトップダウン設計であること.

(iii) マイクロアーキテクチャの設計/動作記述に際しては, モジュラリティ (モジュール分割の大切さ) を明確に意識させていること.

(iv) 制御信号発生のための符号化に際しては, 結線論理制御方式とマイクロプログラム制御方式を同一視させていること.

(v) 必要最小限の周辺回路設計を指導するなどして学生の負担を減らし, 自ら製作までを行なわせることで「作る喜び」を実感させていること.

周辺回路にはトグルスイッチとLEDを用いた. 自立して動作できる, 必要最小限である. 2年目の平成9年度は, さらに, 電源を既配線とする汎用のボードを用意して本質的でない負担を軽減した.

図1に, City-1 (情報工学実験III) で製作されたボードを例示する.

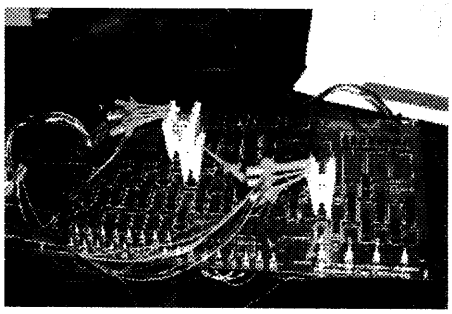

図1:製作されたボードの例

自由な設計が許されるが, 設計があまり大規模にならないよう, また, 学生に過度の負担を強いないよう, 「主要な演算は8ビットで行なうものとする」という制約は課している. しかしこのことはバスの幅が8ビットであることを意味するものではない. 実際, 平成9年度は, 主要な演算は8ビットだが, アドレスバスの幅はインストラクションとデータで独立した15ビットずつの2系統, 計30ビットの幅をもち, インストラクションの幅は24ビット, データの幅は16ビットというコンピュータも設計され, 製作された. ちなみにこのマシンは3段パイプラインのCISCであったが, 無事にユークリッドの互除法の計算をこなし, 設計者自身は「3バイト命令」を自賛し, 感想は「おなかいっぱい」であった.

2.4 指導の内容

2.4.1 指導の進め方

City-1 (情報工学実験III) のシラバスはトップダウンの設計/製造過程を忠実に反映したものになっている [13]. 学生は商品企画から出荷検査までのすべての工程を体験する.

学生の能力には差がある. 著しく進まない学生をつくらないよう, 提出物の期限は, 順を追って, 小刻みに指定するよう配慮している [13].

2.4.2 HDLの選択

HDLとしては下記のような理由からVerilog-HDLを選んで用いている:

[**実績**] IEEE1364 として標準に指定されているだけでなく, SPARC や PowerPC などの豊富な実績があって, 事実上の業界標準として世界的に広く普及している.
[**単純さ**] 基本的には, 信号線 (wire) と記憶素子 (reg) という 2 種類の変数しかなく, ポートは入力 (input), 出力 (output) 双方向 (in-out) だけという必要十分な型のみを用いており, わかりやすい.
[**型宣言**] 信号の値が 6 論理値 9 ストレングスに予め決められているなど, 繁雑な型宣言の必要がなく, 型宣言の不一致 (設計ごとに型宣言が無意味に異なること) を招く危険もない.
[**素子の扱い**] クロック信号を特別扱いし, 最小 (min), 最大 (max), 典型 (typ) の遅延時間が指定でき, 必要ならスイッチレベルの記述も可能であるなど, 半導体ベンダーが責任をもって製造できる厳密な記述が可能.

Verilog-HDL に対しては, 数多くの処理系が供給されているが, この実験では論理シミュレータとして Verilog-XL, ロジックシンセサイザとしては Synergy を選び, HDL desktop というランチャーから起動して用いている.

Synergy が自動生成する回路図情報 (ネットリスト) は Composer (dfII) というツールを介して XACT という自動配置配線プログラムに送っている. ターゲットは Xilinx 社の XC 4000 シリーズとしている.

2.4.3 計算機環境

情報科学部の教育用機器として, City-1 が利用可能な WS が学生用に 60 台, 教官用に 1 台 (SS20) 導入されている. 学生用 WS の OS は Sun OS 4.1.3_U1 である. ホームディレクトリは FDDI 上の 2 台のファイルサーバーに置かれている.

別に, 情報工学科共用の機器として Sun WS を学生用に 30 台, 教官用に 1 台導入している. 学生用 WS の OS は Solaris2.5.1, 教官用 WS の OS は Sun OS 4.1.3_U1 である. 情報工学科では, 学部の 2 台のファイルサーバーとは別に, 独自のファイルサーバーとして Sun の Enterprise3000 を導入している. OS は Solaris 2.5.1 である [13]. すべて X-Window 環境で使用している.

図 2 に学部環境での CAD ツール利用風景を示す.

図 2 : CAD ツールの利用風景

3 制御信号の符号化

ここでは結線論理制御方式とマイクロプログラム制御方式を同一視について詳述する. 組合せ論理回路や順序回路の合成方法の指導については文献 [13] を参照されたい.

3.1 同一視の方法

データパスに送る制御信号の符号化に際して, 結線論理制御方式とマイクロプログラム制御方式の間に本質的な差はなく, これらは一括して扱うことができる [9],[10].

図 3 にこの考え方を示す.

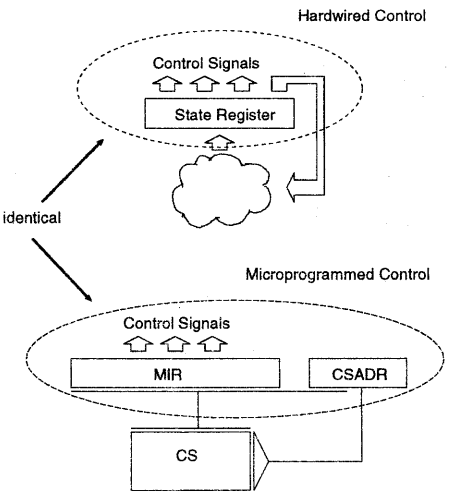

図 3 : 制御信号符号化における同一視

マイクロプログラム制御方式において, 制御メモリから制御信号を読み出すにあたり, 制

御メモリアドレスレジスタ（CSADR）にアドレスを置き，呼び出した信号をマイクロ命令レジスタ（MIR）に置く場合は，これら2つのレジスタをまとめて単一のレジスタとみれば，これら2つのレジスタがちょうど結線論理制御方式における状態レジスタに対応するというのが，その基本的な考え方である．

このような同一視は，従来マイクロプログラム制御方式にしかできないと考えられることもあった，複雑で長く，組織的な開発の行なわれる制御を結線論理制御方式に可能ならしめる．結線論理制御にしかできない制御というものも存在しない．

情報工学実験IIIでは，結線論理制御方式の制御部の設計に，従来，水平型のマイクロプログラム制御方式に用いられてきた符号化をまず指導している．以下に記述例を示す：

```
reg [39:0] STATE;
// STATES
'define I_FETCH_1   40'b0100  ...   1000010010
'define I_FETCH_2   40'b0100  ...   1000010001
'define LOADREG0    40'b0100  ...   0100000000
                    :
assign {FUNC_CTL_A,IR_LEFT,IR_RIGHT}=STATE[2:0]
                    :
       {MEM_WRITE,MEM_READ}=STATE[39:38];
always @ (posedge CLK1)
    begin : state_machine
        case (STATE)
            'I_FETCH_1:  STATE <= 'I_FETCH_2;
                begin
                    case (DATA_BUS)
                        :
            'I_FETCH_2:  :
```

3.2 Mealy型とMoore型

制御部は，結線論理制御方式の場合，制御信号発生のための順序回路そのものである．上述した同一視は，「メモリは組合せ論理回路だとみなすことができる」という認識に基づいている．この意味で，従来マイクロプログラム制御方式で用いられてきた，「特定のビットを立てることで，別ビットの意味を変える」という間接符号化[2]も，全く等価な形で，結線論理制御方式に用い得る[10]．制御回路が順序回路（状態機械）であると認識すれば，Mealy型とMoore型の得失にも注目することになる．

図4に示すようなシフトレジスタを用いた制御信号の生成は，シフトレジスタの組を状態レジスタと見れば，いわゆる，One hot encodingを用いた状態割り当てとみることができる[10]．

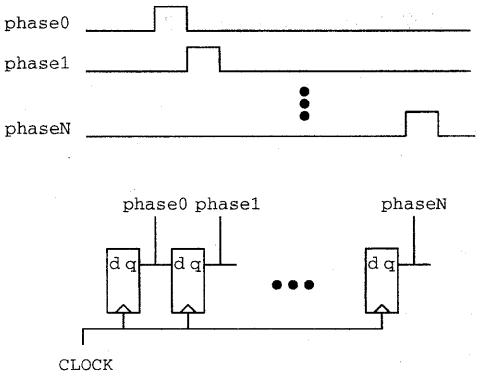

図4： One hot encoding

各信号線がどの状態にあるかを示すことなる．このとき，どの信号線が1であるかによって示される状態は，「フェーズ」と呼ばれることがある．この「フェーズ」という概念は後述する「マシンサイクル」という概念が前提だと考えられる．この制御方式の場合，各状態で行なわれる処理内容が排反でないなら，制御点（制御信号が供給される点）に最終的に供給する信号を作るための付加回路が必要になる．この意味で，通常，Mealy型とすることを強いられる．Mealy型はデータパス系の負担を増加させる．1制御点1本の符号化は，データパス系の負担を増加させない．Mealy型とMoore型のどちらも，実際の設計に用いられているが，Moore型の方が状態数は多くなる傾向がある[11]．

City-1が，水平型のマイクロプログラム制御方式に用いられてきた，1制御点1本という符号化をあえて指導しているのは，(i) 制御回路は順序回路（状態機械）であり，符号化には多くの自由度があること，(ii) HDLを用いたトップダウン設計によって，今やその任意の符号化を自由に使い分けることが可能になっていることを徹底的に理解させるためである．水平型のマイクロプログラム制御方式に用いられてきた符号化を結線論理制御方式の制御部の設計の一例として示すことは，Mealy型とMoore型の差に注目させることにも有効である[13]．

Mealy型とMoore型の得失を理解し，動作可能クロック周波数を高めようとする学生も多数見受けられた．

4 完全なインターロック

インターロックは，ハザードを検出し，回避する機構である [1]．計算の本質はデータ依存関係であって，3 とおりのデータ依存関係が 3 とおりのハザードに対応している [11]．City-1 においては，完全なインターロックを徹底的に理解することが基本であり不可欠であると筆者らは考えた．

4.1 予約表

近年，横軸は時間だが，縦軸に命令を列挙し，それぞれがどのステージの処理を受けているかでパイプラインを表現する場合 [1] が見受けられる．しかし，City-1 では，横軸は時間で，縦軸にはパイプラインステージ（ユニット）を並べ，各ステージ（ユニット）がどの状態にあるかを示すことでパイプラインを設計させている．いわゆる予約表（Reservation Table）である．この表現はタイミングチャートの自然な拡張であって，パイプライン設計の基本は決して高度でもなく複雑でもないことを理解させることに役立っている．タイミングチャートでは信号の値の時間的な変化が示されるのに対し，予約表では各ステージの状態がどの命令を処理しているかで示される．イニシエーションの間隔をレイテンシーと呼ぶことは，いずれの表現方法においても同じである．

完全なインターロックを行なうということは，機械語がどのような順序で与えられても，注目するコンピュータが期待どおり動作することを意味する．学生には，このことを繰り返し説明すると同時に，「どうあっても NOP に頼ってはならない」という指示を与えた．

4.2 学生に示す具体例

学生に示す記述は，もちろん完成されたものであってはならないし，巧みな設計であってもならない．技術的に優れていることを誇るべきは学生の側であって，指導する側ではない．

学生に示す設計の具体例としては，2 相のクロック信号を用いる 3 段パイプライン CISC の不完全な動作記述を工夫した．図 5 に予約表を示す．

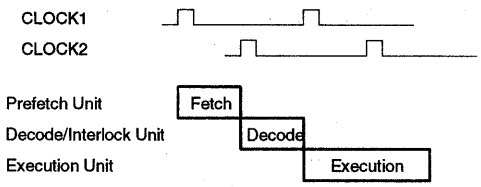

図 5：具体例として示した CISC の予約表

ここでは「相」（フェーズ）という言葉を，注目するコンピュータの主要な動作サイクルを細分化する位相という意味に用いた．図 5 の予約表をもつコンピュータにおいてレイテンシーは CLOCK1 の 1 クロックである．この 1 クロックが，コンピュータの主要な動作サイクルという意味で，1 マシンサイクルである．これはスループットだともいえる．CLOCK2 はこれを細分化しており，1 マシンサイクルを 2 分している．

このコンピュータは CLOCK1 の半クロックで命令をフェッチし，次の半クロックでデコードを行ない，次の 1 クロックで実行を行なっている．実行のための時間を 2 倍確保しているのは，CISC の場合，メモリからのデータをフェッチし，さらにこれをオペランドとする演算までを行なわなくてはならないからである．学生に渡すときは，まずこの点を強調した．

これに手を加えて RISC を実現しようとする学生は，RISC の場合，オペランドをメモリからフェッチして演算までを行なうことがないため，2 倍のクロック周波数を用いた単相の 3 段パイプラインを設計することになる．他方，CISC でさらにパイプライン化を進めようとする場合は，オペランドフェッチを独立した別のステージとする 4 段のパイプラインを設計することになる（図 6）．

CISC のままパイプライン化を進めようとする学生には，「4 段の CISC とした場合，最終ステージがメモリへの書き込みを行なうと，オペランドフェッチを行なう 3 段目のステージとの間で RAW ハザードが生じる．CISC であっても，メモリアクセスを 3 番目のステージだけに集中できればレジスタまわりの RAW ハザードだけを考えれば済むようになる」という助言を与えた．

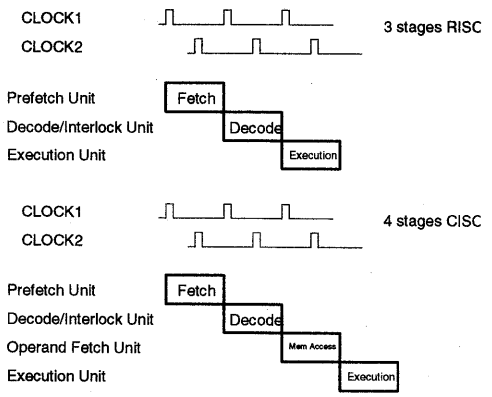

図 6：模範解答の予約表

ステージ間のデータの受渡しには，深さ 1 だという意味で通常のバッファを置き，併せて次のステージにレディー信号を送るという単純な手段を指導した．以下に，デコード / インターロックユニットの記述例を示す：

```
reg [13:0] DECODED_INSTRUCTION_BUFFER;
reg DECODED_INSTRUCTION_READY;

assign STALL=FRAG_BUSY && (INSTRUCTION==...

always @ (posedge CLK1)
   begin
      DECODED_INSTRUCTION_BUFFER
        =decoder(INSTRUCTION_BUFFER[15:8]);
      DECODED_INSTRUCTION_READY
        =IF_INSTRUCTION_READY;
      MEM_OPERAND_ADDRESS
        =INSTRUCTION_BUFFER[7:0];
                  ⋮
```

プリフェッチユニットは，IDLE, FETCH, WAIT の 3 状態をもち，上に示したデコード / インターロックユニットから STALL の信号を受けとると WAIT の状態に入る．

フォワーディングについては，別途，タイミングチャートと記述例を提供した．

5 履修の状況

履修登録した 54 人のうちの 47 人が，課題として指定したユークリッドの互除法を各自の機械語で書き下し，各自のコンピュータ上で走らせた．47 人のうち 28 人は 3 ～ 4 段パイプライン RISC を，19 人が 3 ～ 4 段パイプライン CISC を完成させた．履修登録した 54 人の全員が，完全なインターロック機構を搭載した 3 段以上のパイプラインコンピュータの設計と製作を行なった．

図 7 にボードの製作風景を示す．

図 7：2 人ひと組での製作風景

命令数は 23 個～ 85 個と様々であった．オペランド数は 2 ～ 3 アドレス方式であった．ちなみに，3 アドレス方式を採用していた学生の方が，同じ内容のプログラムを処理するためのクロック数が少ないという意味において，高速なコンピュータを実現していたようである．レジスタ数は 4 ～ 16 個と様々であった．

ほとんどの学生にとって，スタックポインタを用いたサブルーチンコール / リターンは，やはり，最初に解くべき課題であった．

簡単な指導だけで，アセンブラを開発した学生が数名いたというのは 2 年目も同じであった．yacc,lex が使用された．

ほとんど全員が，インストラクションの幅が 16 ビット，データの幅が 8 ビットの，2 系統のメモリをもつという意味の，ハーバードアーキテクチャを採用していた．スループットは CISC も RISC も，ほぼ，1 命令 1 クロックであった．

4 段パイプラインの RISC は，書き込みを最終ステージとしていた．ハザードの対策にはフォワーディングが多用されていた．

4 段パイプラインの CISC には，前述した模範解答の他に，ユニークな設計が見受けられた．図 8 にその予約表を示す．

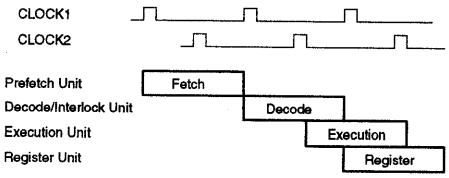

図 8：4 段パイプラインの CISC の別解

初年度の経験があったため，授業日数の点でも，学生の負担は軽減できた．

1日4.5時間, 週1日の他, 同じ4.5時間の予備日が1日の1週計2日, 全体で15週が正規の授業日程である. はじめの3カ月, 予備日を使うことはなかった. デバッグの過程では, 課外利用申請も行なわれたが, 夏休み中の課外利用（補講）は1週だけで済んだ. 夏休みに入る前の試験期間中にデバッグをする必要は皆無だった.

City-1は自由な設計製作環境であることに力点を置いている. 設計の現場で他社が出荷する市販品と競うことを思うと前途遼遠である. この実験では,「いたずらに高度な技術が要求されている」という印象を与えることも,「コンピュータを作ることは難しい」という印象を与えることも極力避けるよう配慮している.

レポートでは, 約15人が, 極めて前向きな感想を述べていた.「一般情報処理」という1学年前期の, いわゆる「コンピュータリテラシイ」の授業[14]では, 単に使い方を指導するだけで,「電子メール」に対し, 予想をはるかに越える関心が示される. 特別の準備も工夫も必要ない.「通信」, あるいは「コミュニケーション」というものに対する, 根源的な関心だと思われる. 広島市立大学では, 入学と同時に, 全学生にメールアドレスが割り当てられる. City-1（情報工学実験III）のレポートでは,「とにかく, 作っているときが楽しかった」という率直な感想が示される.「将来は, 人間と会話のできるコンピュータを作ってみたい」などの夢も語られる.

6 まとめ

本稿では, 広島市立大学情報科学部情報工学科3年前期の設計教育環境City-1（情報工学実験III）の2年目を示して,CAD利用技術, FPGA利用技術を用いた, 大学での教育研究の現状を紹介した.

技術的にはCity-1が有するバス構成の自由度をより一層引き出させること, 日程という面では, 負担を平準化して, 終盤の負担を軽減することが今後の課題である.

本研究で用いているCADシステムの, 超高速論理シミュレータ, ロジックシンセサイザのライセンスはCADENCE社のアカデミック・プログラム, 自動配置配線プログラムのライセンスはXilinx社のユニバーシティー・プログラムによるものである.

参考文献

[1] Hennessy J. L. and Patterson D. A.: Computer Architecture A Quantitative Approach 2nd edition, Morgan Kaufmann Publishers, Inc.（1996）

[2] 萩原 宏：マイクロプログラミング, 産業図書（1977）

[3] 相磯秀夫, 飯塚 肇, 元岡 達, 田中英彦：計算機アーキテクチャ, 岩波書店（1982）

[4] 村岡洋一：コンピュータアーキテクチャ, 近代科学社（1981）

[5] Hwang, K.: Computer Arithmetic, John Wiley & Sons（1979）

[6] 吉田典可：論理数学II- 組合せ論理回路 - 共立出版（1978）

[7] 吉田典可：論理数学III- 順序論理回路 - 共立出版（1978）

[8] 当麻喜弘：スイッチング回路理論, コロナ社（1986）

[9] Takahashi R., Yoshimura T. and Goto S.: A VLSI Architecture Evaluation System, Proc. ICCD'86 pp.60-63（1986）

[10] 高橋隆一, 吉村 猛：ハイレベルシンセシスの動向, 信学論A, Vol.J74-A, No.2, pp143-151（1991）

[11] 高橋隆一：システム開発と設計, サイエンス社（1996）

[12] 稲吉秀夫, 富田真治, 日比野 靖, 平山正治, 山本昌弘, 飯塚肇: パネル討論会「RISCはCISCに勝るか」, 情報処理 vol.30, No.11, pp.1376-1394（1989）

[13] 高橋隆一, 児島 彰, 上土井陽子, 吉田典可：マイクロコンピュータ設計教育環境City-1, 情処研報 Vol.97, No.17, pp.41-48（1997）

[14] 高橋隆一, 内田智之：コンピュータ入門としての一般情報処理教育, 情報処理教育研究集会講演論文集 pp.47-50（1994）