

# ソフト・ハードウェア連携を主眼としたライトレーザ開発カリキュラム

静岡県立科学技術高等学校 安田倫己

## 1 はじめに

ソフトウェア，ハードウェアともにその技術的發展は目覚しく，工業高校における情報技術教育の内容も高度化しつつある。コンピュータが誕生したときは，もちろんハードウェアとソフトウェアの区別などなかったであろうし，ハードに強いソフトウェア技術者やソフトに強いハードウェア技術者の育成が言われた時代もあった。また，近年，ハードウェアをソフト的に仮想化する技術や組み込みマイコン等の技術が発展し，ややもするとソフトウェアに偏重した情報技術教育に陥りやすい。しかし，工業高校生のような初心者に対する情報技術教育として，バランスの取れた学習が効果的と考える。基本概念をハード・ソフトウェアの区別を意識することなく学習するために，簡単なライトレーザを題材にソフト・ハードウェアの連携に留意した実践を行い，生徒へのアンケート調査を実施し，良好な結果を得たので報告する。

## 2 本研究でのカリキュラム

研究に先立ち，コンピュータの動作の仕組み学習することを目的に，1年生前半，アセンブラ（CASL II）の学習を取り入れた。これを受け，1年生後半から2年生にかけて，C言語の学習をおこなった。はじめにアセンブラ学習を行うことで，以後の言語学習が容易かつ多様に展開できたようだ。

これらの生徒を対象として，3年生においてライトレーザを設計・製作する実習を行った。実習は，次のような順序で実践した。まず，UMLモデリングツール「JUDE/Community」(\*)により，思考，設計の経験と効率化を計りながら，ライトレーザの設計を体験した。次に，比較的シンプルなZ-80IPCore(\*\*)のFPGAへの組み込みとその操作を行うインターフェイスを製作し，LEDの点滅，モーターの駆動や8セグメントLEDによる表記などを体験した。そして，これらの設計計画と制御機構を用いて，ライトレーザの製作を行う。ライトレーザなどのものづくりにおける機械加工精度の重要性はいうまでもないが，残念ながら，情報技術科においては，その方面の十分な教育を行うことができない。そこで，今回は，LegoMindstormによる実装を行った。RCXタイプのLegoMindstormをカスタマイズする予定である。

## 3 実践内容

### (1) JUDE/Communityによる設計

思考支援ソフトウェア「JUDE/Community」によるUML設計を行った。ユースケース図，状態チャート図，クラス図，シーケンス図（図1）などの作成を実習した。このようなソフトウェアの存在を早くから知りたかったとの生徒の感想が印象的だった。

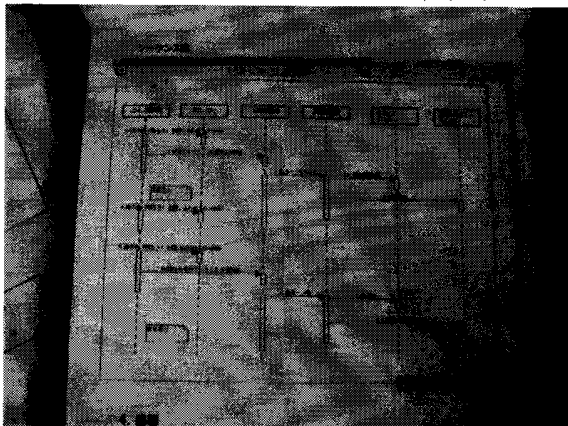


図1 シーケンス図

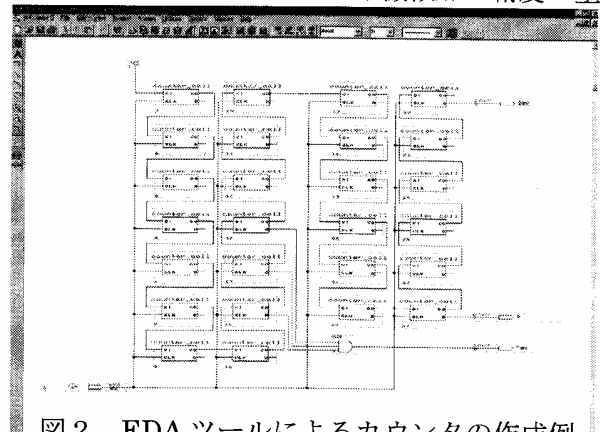


図2 EDAツールによるカウンタの作成例

### ソースファイル

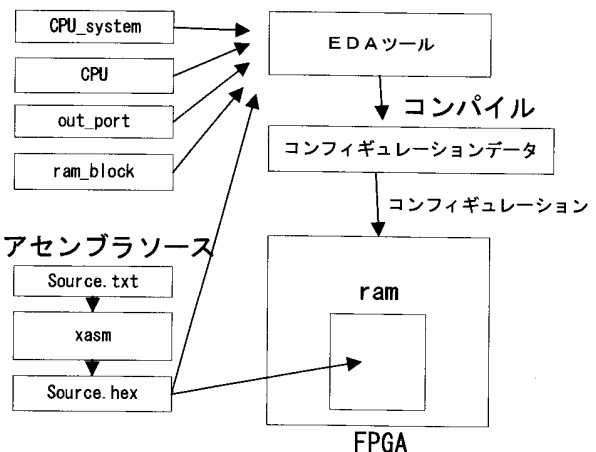


図3 CPU設計のモデル

## (2) FPGAによるZ-80の設計と製作

VerlogHDLによる回路設計方法を学び、EDA ツール(図2)を用いて、コンピュータを構成する基本的な論理回路の設計を行った。CPU設計のモデルとして図3のような仕組みに、コードを記述した。アセンブラ命令と機械語コードとして、Z80ライクなコードを採用した。ファイルは、CPUのシステムの部分、CPUコードのコアの部分、入出力の部分、メモリの部分といったそれぞれの役割を持つ4つのファイルで構成した。

アセンブラプログラムの処理は、テキストファイルを「xasm」というCPU用クロスアセンブラにかけ、hexファイルに変換しFPGAの内蔵メモリに格納した。アップダウンカウンタやスイッチの入出力用のアセンブラプログラムを作成し、自作プロボード上で動作確認をした。

プロトボードは、スイッチやリレー回路、8セグメントLED、FPGAの書き込み用の端子をのせたFPGAを中心とした入出力装置として作製した。FPGAの端子とプロトボード上の入出力デバイスとの関連付けをピン配置情報としてFPGAに書き込んだ。その一部を、図4に示した。高校生にも十分に対応可能なテーマであることが確認できた。

```
NET "GCLK<0>" LOC = "P38";
NET "LED<1>" LOC = "P0";
NET "GCLK<1>" LOC = "P39";
NET "LED<2>" LOC = "P4";
NET "GCLK<2>" LOC = "P36";
NET "LED<3>" LOC = "P8";
NET "GCLK<3>" LOC = "P37";
NET "LED<4>" LOC = "P11";
```

図4 ピン配置情報

## (3) LegoMindstormをライントレーサの機構とした活用

自立型ライントレーサの評価の要因の一つに、ギヤやシャーシの加工技術の良し悪しがある。情報技術を主に学習する生徒は、加工技術の本格的な習得が難しい。この部分を補うものとして、LegoMindstormを採用してみた。

実際にRCX(H8マイコン)にて、生徒に自由にライントレーサを製作させたところ、SygwinとBrickOSを組み合わせたC言語、JAVA、アセンブラライクNQCによる制御など積極的な参加が見られた。

今後、RCXの制御部分をFPGA/IPCoreに変更して、汎用のZ80用のCコンパイラによる機構制御を行っていく予定である。

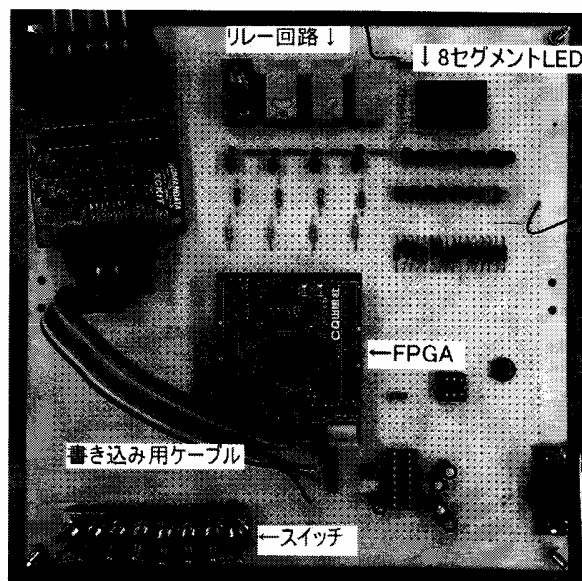


図5 実験用のプロトボード

## 4 おわりに

情報技術科のカリキュラムは、ややもするとソフトウェアに傾く傾向がある。しかし、1・2年生において、十分にハードウェアを意識した指導を行い、情報技術を学習する生徒の弱点をカバーする題材を選ぶことで、情報技術科のものづくりにあった一つのカリキュラムを構築できるのではないかと。今回の実践では、FPGAなどのやや高度なデバイスも使用しているが、既存のツールを活用することで、高校生でも対応可能なテーマとなった。まだ、いくつか開発すべき部分が残されているが、今後一連のカリキュラムの完成とその実践を目指したい。

最後に、技術的協力をいただいたアルファシステムズ(株)吉田氏と実践に協力してくれた生徒諸君とに感謝します。なお、本研究は、平成19年度科学研究費補助金(奨励研究)の助成を受けて行ったものです。

(\*)株式会社チェンジビジョン

(\*\*)アークピット社: xasm <http://www.arcpit.co.jp/index.htm>

```
#include <dsensor.h>
#include <dmotor.h>
#include <unistd.h>
#include <conio.h>
Const unsigned int THRESHOLD = 40;
int main (int argc, char *argv[ ]) {
    unsigned char s_speed =100; /* 速い*/
    unsigned char t_speed =100; /* 遅い*/
    cputs("YUPA"); /* 液晶「YUPA」*/
    ds_active( &SENSOR_2 );
    motor_a_speed( s_speed );
    motor_c_speed( t_speed );
    sleep( 1 );
    motor_a_dir( fwd );
    motor_c_dir( fwd );
    while( !( shutdown_requested() ) ) {
        sleep( 0.2 );
        lcd_int( LIGHT_2 );
        if ( LIGHT_2 < THRESHOLD ) {
            motor_a_dir( rev );
        }
    }
}
```

図6 生徒のソースコード例(一部)