論文

高度情報化人材育成用 ASIC マイコン教材の開発

森　久直*1)　坂巻佳壽美*1)　佐藤正利*1)　村越英樹*2)　宇賀神孝*3)
村山彰一*3)　波多野八州夫*3)　山本大介*3)

Development of a microcomputer training tool to educate advanced information-technology engineers

Hisanao MORI, Kazumi SAKAMAKI, Masatoshi SATOU, Takashi UGAJIN, Shouichi MURAYAMA,
Yasuo HATANO, Daisuke YAMAMOTO and Hideki MURAKOSHI

Abstract　Embedded systems have come to be applied to a wide range of products. However, a lack of embedded system engineers become a problem at the development stage, and training of embedded system engineers has been speeded up. Therefore, a microcomputer training tool for the training of embedded system engineers was developed. This microcomputer training tool is mainly composed of a remote debugging tool and a MPU(Micro Processing Unit)-COMETⅡ which was implemented on a FPGA(Field Programmable Gate Array).

Keywords　Embedded system, Micro processing unit, COMETⅡ, Remote debugging tool, FPGA¥

1．はじめに

近年，マイコン組み込みシステムを導入した機器が広く普及してきた。しかし，このような機器の需要が高く，今後も高まる傾向があるにもかかわらず，開発現場では組み込みシステム開発技術者が不足している。そのため，電子機器業界では組み込みシステム開発技術者の早急な育成が求められている。しかし，このような技術者育成に適した教材が見当たらないのが現状である。

そこで，組み込みシステム開発技術者の育成に適したマイコン教材を開発した。

2．研究内容

2.1　マイコン教材の構成

開発したマイコン教材は，マイコンボード，ソフトウェア開発ツール，テキストで構成している。

ソフトウェア開発ツールをインストールしたパソコンと,マイコンボードは，図１のようにシリアル通信回線（RS-232C）で接続して使用する。テキストの内容は，MPU の解説，I/O の操作方法，マイコンボードやソフトウェア開発ツールの操作方法，問題と解答となっている。

2.2　マイコンボードの開発

マイコンボードは，MPU ボードと周辺 I/O ボードの２枚のボードから成っている。MPU ボードは「FPGA」,「FPGA 用コンフィグレーション ROM」,「ユーザ用 ROM」,「ユーザ用 RAM」で構成した。また，MPU ボード単体での利用や,他の開発ボードとの接続を可能とするために，「周辺 I/O ボード」とコネクタ接続とした。

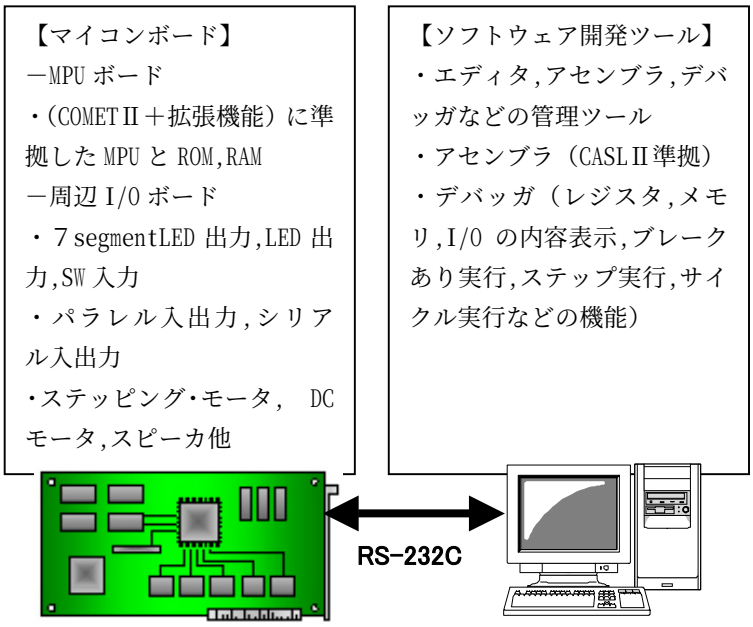


図１　マイコン教材の構成

2.2.1　MPU ボードの設計

教育用という観点から，MPU の仕様は理解しやすいことが望ましい。したがって,①,②,③,④の条件を満たした COMETⅡ準拠の MPU を設計した。COMETⅡは情報処理技術者試験に出てくる仮想コンピュータである。

*1) 情報システム技術グループ　*2) 都立科学技術大学
*3) アンドールシステムサポート（株）

①基本的なハードウェアと命令セットで構成される
②割り込み機能を有する
③MPU 内部の動作を外部から確認できる機構を有する
④教材として既存のメーカー規格にとらわれないこと

【主な仕様】

―16 ビットマイクロプロセッサ―

◇ 1 ワード 16 ビット
◇ データバス，アドレスバス 16 ビット
◇ スーパーバイザ／ユーザモード
◇ マスカブル割り込み 16 本，ノンマスカブル割り込み 1 本（ベクタ割り込みタイプ）

―命令セット―

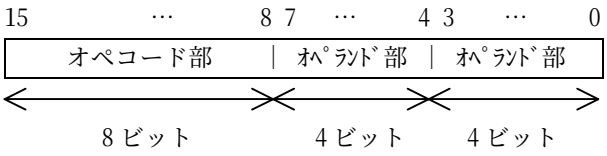
◇ 1 ワードもしくは 2 ワードで, 40 種類

表 1　命令の種類

データ転送命令	LD,ST,LAD
演算命令	ADDA,ADDL,ADDLV,SUBA,SUBL,SUBLV, AND,OR,XOR,CPA,CPL,SLA,SRA,SRA,SRL
分岐命令	JPL,JMI,JNZ,JZE,JNV,JOV,JUMP
スタック操作命令	PUSH,POP
コール・リターン命令	CALL,RET,RETI,SVC,SVR,RST,RETRST
その他の命令	NOP,NOP0,EI,DI,SETV,CLRV

◇ 命令構成

1 ワード目



2 ワード目

―プログラミング資源―

◇ 8 種類のレジスタ

PR(プログラムレジスタ)　:次に実行すべき命令語の先頭アドレスを保持している。
IR(命令レジスタ)　:1ワード目に読み込んだ命令コードを保持している。
TMP1(作業レジスタ1)　:MPUが処理を行う途中で利用するために一時的に値を保持している。
TMP2(作業レジスタ2)　:MPUが処理を行う途中で利用するために一時的に値を保持している。
GR0…GR7(汎用レジスタ)　:値の保持や演算に用いる。GR1…GR7はインデックスレジスタとして用いることができる。
SP(スタックポインタ)　:スタックの最上段のアドレスを保持している。
IntR(割り込みレジスタ)　:割り込み要求を保持する。非同期に、立ち下がりエッジをもって割り込み要求を受け付ける。また、割り込み要求の処理が終了すると、クリアされる。
MR(マスクレジスタ)　:割り込み要求をマスクする。MPUが特権状態のときに、操作可能となる。
FR(フラグレジスタ)　:演算命令や割り込み、モード変更により、次のように値を変化させる。
- SV スーパーバイザモード時に値は '1'、ユーザーモード時に値は '0'
- NF ノンマスカブル割り込み要求を受け付けた時に '1'、そうでない時に '0'
- IF 割り込み許可する時に '1'、割り込み禁止する時に '0'
- OF オーバーフローを起こした時に '1'、そうでない時に '0'
- SF ロード命令や算術演算命令を実行し、値が負の時に '1'、それ以外の時に '0'
- ZF ロード命令や算術演算命令を実行し、値が0の時に '1'、それ以外の時に '0'

◇ 64K ワードのﾒﾓﾘｱﾄﾞﾚｽ空間（ﾒﾓﾘﾏｯﾌﾟﾄﾞ I/O）

―その他―

◇ 8 ビットの MPU ステータス信号出力
◇ /READY 信号入力による MPU ウェイト

前述した主な仕様をもとに，ソフトウェア／ハードウェア・コ・デザインツール「CardTools」を用いて, 設計とシミュレーションを行った。「CardTools」は，ソフトウェア（C 言語）と，ハードウェア（ハードウェアモデリング言語）を同時に設計し，システムの不整合のない仕様を作成するツールである。いくつかの命令を実行したときのシミュレーションでは，個々の資源の動作が正確に行われているか，動作の衝突が起こっていないか，等を確認することができた。

この結果により，MPU の制御回路におけるステートマシンの設計仕様を得ることができた。

次に, VHDL（Very High Speed IC Hardware Description Language）で MPU の機能を仕様記述し，FPGA に書き込んだ [1)]。

本研究で設計するモデル MPU は，教材向けであり，高い処理能力を追求したものではない。したがって, MPU の内部構成にあたっては，命令ごとの動作が分かりやすくなるように設計した [2)]（図 2）。

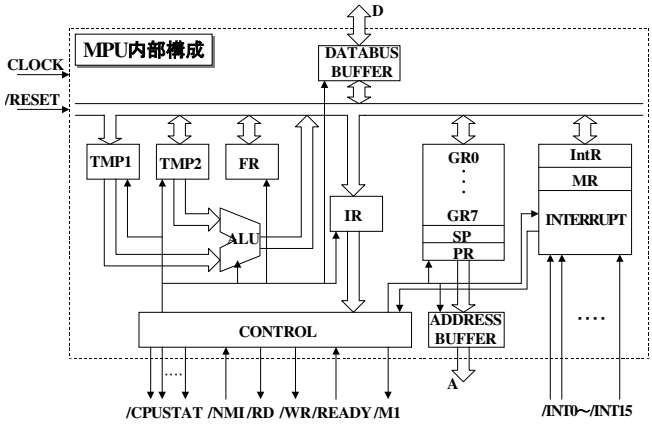


図 2　MPU の内部構成

2.2.2　周辺 I/O ボードの設計

周辺 I/O ボードには，様々なマイコン制御を学習できるように, MPU の周辺 I/O を実装した。入出力には「パラレル」,「シリアル」,「アナログ」の 3 タイプを用意した。その他には,「LED」,「タイマ／カウンタ機能」,「割り込み機能」,「ステッピングモーター」,「直流モーター」を実装した。

周辺 I/O ボードの設計とシミュレーションは,「CardTools」を用いて行った。7segmentLED，スピーカ，ステッピングモータ，直流モータを制御する機能ブロックを個別に設計し，それらの中心となる MPU と各機能ブロックの間に I/O レジスタを配置した。この I/O レジスタを操作することで，7segmentLED 等の機能ブロックを動作させるようにした。

そして，シミュレーションで正確な動作を確認した後,

周辺 I/O ボードを開発した。

2.3　ソフトウェア開発ツールの開発

2.3.1　アセンブラの開発

表 1 に示す命令語を用いたニーモニック表記の CASLⅡソースプログラムファイルを入力して，機械語コードへ変換するアセンブラプログラムを開発した。CASLⅡとは，COMETⅡが処理できるアセンブラ言語である。

アセンブラ変換処理の流れを図 3 に示す。ソースプログラムファイルを 2 回読み込む，いわゆる“2 パス方式”とした。PASS1 では，ラベルテーブルの作成，機械語コード（変換結果）の出力，エラーメッセージ処理などを行い，それぞれをファイルとして出力するようにした。PASS2 では，PASS1 の変換結果を基に未完成部分を補充し，変換結果リストの作成，ROM 化データ（Intel-HEX フォーマットのデータ）の生成，エラー箇所の表示などを行うようにした。

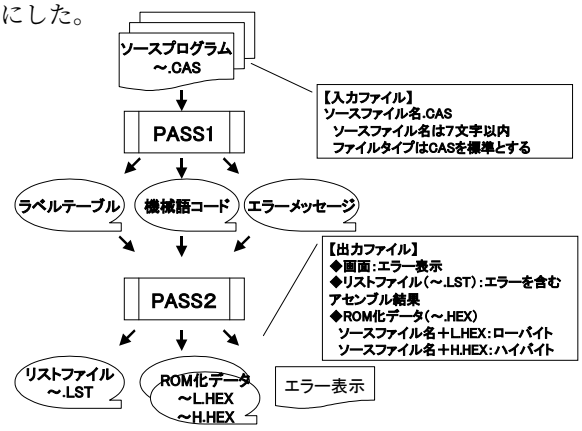


図 3　アセンブラ変換処理手順

特に，エラー出力に関しては，教育用ということを踏まえ,出来るだけ詳細な内容を提供するように心がけた。また，ROM 化への対応は，組み込みシステム開発技術者育成には欠かせない要件であり，今回のアセンブラ開発に於いては，重要な機能となっている。

2.3.2　C コンパイラの開発

現在のシステム設計ではアセンブラ言語とともに C 言語が広く用いられている。そこで，高級言語を用いた組み込みシステム開発の教育を行うことを可能にする独自の COMETⅡ用 C コンパイラを開発した。この C コンパイラにより，C ソースプログラムは CASLⅡソースプログラムに変換・出力される。

開発した C コンパイラは，組み込みシステム開発に必要な演算子，制御文をサポートしている。

図 4 は C ソースプログラムの一例である。この C ソースプログラムをコンパイルした結果を，図 5 に示す。

```
int main( int a, int b ){
   int s, t;
   while(1){
      if(a)
         t=111;
      else if(a + b + s + t)
         t=t + 222;
      else if(b)
         break;
      else
         t=t+999;
    }
}
```

図 4　C ソースプログラム例

```
;;int main( int a, int b ){
        start
;;int s, t;
        push    gr0
        push    gr0
;;while(1){
        ld      gr1, #0000
        svc     gr1
WLL2
        lad     gr0,1
        jze     WLL3
;;if (a)
ld              gr0,3,gr7
        Jze     LL2
;;t = 111;
        lad     gr0,111
        st      gr0,0,gr7
        jump    LL3
LL2
;;else if (a + b + s + t)
        ld      gr0,0,gr7
        push    gr0
        ld      gr0,1,gr7
        push    gr0
        ld      gr0,4,gr7
        push    gr0
        ld      gr0,3,gr7
        pop     gr1
        adda    gr0,gr1
        pop     gr1
        adda    gr0,gr1
        pop     gr1
        adda    gr0,gr1
        jze     LL4
   (右の段へ続く)
```

```
;;t = t + 222;
        lad     gr0,222
        push    gr0
        ld      gr0,0,gr7
        pop     gr1
        adda    gr0,gr1
        st      gr0,0,gr7
        jump    LL5
LL4
else if(b)
        ld      gr0,4,gr7
        jze     LL6
;;break;
        jump    WLL3
        jump    LL7
LL6
;;else
;;t = t + 999;
        lad     gr0,999
        push    gr0
        ld      gr0,0,gr7
        pop     gr1
        adda    gr0,gr1
        st      gr0,0,gr7
LL7
LL5
LL3
;;}}
        jump    WLL2
WLL3
        pop     gr0
        pop     gr0
        ret
        end
```

図 5　コンパイル結果

2.3.3　デバッガの開発

マイコンのプログラムを作成し，動作確認を行う場合に，書き込み装置で ROM に機械語コード（MPU が処理できるデータ）を書き込む方法がある。しかし，プログラムのデバッグ（間違いの原因を探し出し修正する作業）が頻繁に行われることを考慮すると，ROM の書き込みと交換を行う方法は効率的ではない。そこで，プログラムのデバッグを効率的に行うために，パソコンとマイコンボードを通信回線（RS-232C）で接続するリモートモニタ方式を採用した[3]。パソコン上で作成したプログラム，CASLⅡソースプログラムをアセンブラで機械語コードに変換した後，通信回線を通してマイコンボード上の RAM に書き込み，効率的なデバッグが行えるようになる。

開発したデバッガ（MS-DOS 版）は，パソコン側で動作するソフトウェアと，マイコンボード側で動作するソフ

トウェアからなり，協調動作するしくみになっている。マイコンボードのパフォーマンスを考慮し，デバッガプログラムのサイズについてマイコンボード側を小さく，パソコン側を大きくすることにより，デバッガの協調動作を安定させる工夫をした。

プログラムのデバッグに必要となるコマンドは表 2 に示すように 8 種類ある。

表 2　デバッガのコマンド

コマンド	使 い 方	機　　能
R	Rファイル名	デバッグするプログラムをマイコンボードへダウンロードする。
D	D開始アドレス，（終了アドレス）	メモリ内容の表示
F	F開始アドレス，終了アドレス，データ	メモリ内容の初期化・変更
T	T開始アドレス，（ステップ数）	シングルステップ動作
B	BS設定アドレス	ブレークポイント設定・解除
G	G実行開始アドレス	プログラム実行
X	X(レジスタの指定)，（内容）	レジスタ内容の表示・変更
L	L開始アドレス，（ステップ数）	逆アセンブル表示

さらに，本教材では MS-DOS 版のデバッガを，ユーザにとって操作しやすい Windows 版に開発した。

MS-DOS 版のデバッガを活かすため，これをスレッドとし，Windows アプリケーションとの間でコマンドやデータのやり取りをさせる工夫を行った。具体的には図 6 に示すように，MS-DOS 版のデバッガを，Windows アプリケーションからデバッガ用ファンクションとして呼び出しできるよう，MS-DOS 版のデバッガと Windows アプリケーションの間に，インターフェースとなるプログラムを作成した。このインターフェース・プログラムを，ターミナル I/F と称する。

ホスト PC 側 GUI は，マイコンボード上のユーザプログラムをデバッグするために，ターミナル I/F のアクセス関数を呼び出してコマンドを実行する。そして，デバッガスレッドはコマンドを受け取り，マイコンボード上のモニタプログラムの実行制御を行うことができる。

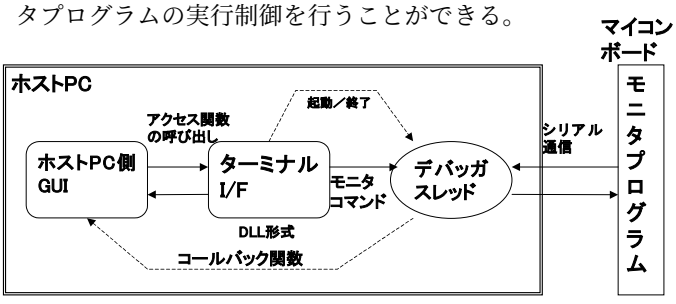


図 6　Windows 版デバッガプログラムの構成

3．結　果

以上の開発内容に基づき，図 7 のようにマイコン教材を製品化した。本教材の中で，ステッピングモータや LED を動作するプログラムをテストしたところ，正確に動作することを確認した。

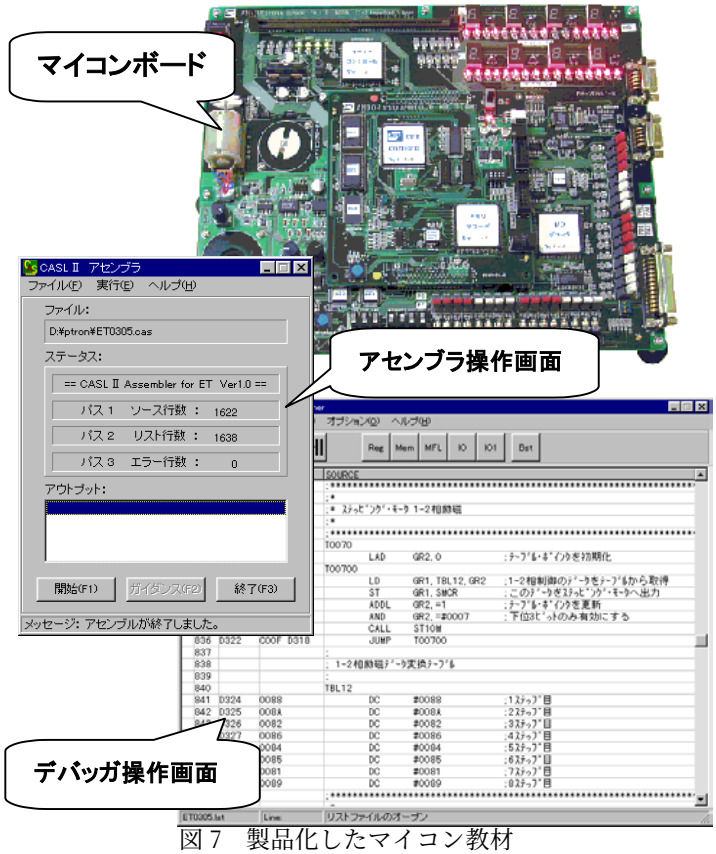


図 7　製品化したマイコン教材

4．まとめ

開発したマイコン教材は,情報処理技術者試験の中で定義されている「COMET Ⅱ」と「CASL Ⅱ」に対応し，既存のメーカ規格にとらわれない学習を可能にした。

本教材の特徴は，周辺 I/O に多くの入出力を実装し，様々なマイコン制御を実習できるようにした点と，デバッガを Windows 版にすることで操作を容易にし，リモートモニタという方式によりプログラム開発を行いやすくした点である。

なお，本教材はアンドールシステムサポート（株），都立科学技術大学，都立産業技術研究所の産学公で共同開発したものであり，現在は都内の教育機関に納品され，講義の中で使用されている。さらに，情報処理技術者試験の導入の動きがある東南アジア等への普及も期待できる。

参考文献

1) 長谷川裕恭:VHDL によるハードウェア設計入門, CQ 出版社(1995).
2) パターソン＆ヘネシー:コンピュータの構成と設計，上巻，下巻，日経 BP 社(1996).
3) 平松隆志,三輪昭生:岡山県工業技術センター報告,高機能ﾏｲｸﾛｺﾝﾋﾟｭｰﾀの ROM 化開発環境の構築，18, 15-23(1992).

（原稿受付　平成 14 年 8 月 1 日）