

教育用仮想コンピュータシステムの考案と授業への実践

青山義弘^{1*}、吉村忠興志²

福井工業高等専門学校 電子情報工学科¹、物質工学科²

(〒916-8507 鯖江市下司町)

*yfa@ei.fukui-nct.ac.jp

Design and Practice of a virtual computer system for education

Yoshihiro AOYAMA^{1*}, Tadayosi YOSHIMURA²

¹ Department of Electronics and Information Engineering, ² Chemistry and Biology Engineering,
Fukui National College of Technology, Geshi-cho, Sabae, Fukui 916-8507, Japan

(Received November 25, 2008; Accepted December 12, 2008)

Abstract

It is important to implement assembler education corresponding to the ubiquitous age of computers. We performed assembler education using a microcomputer chip, which is the virtual computer system of "eAVR" (RISC base, by Atmel Co.). This education was carried out on the subject of "Computer Architecture 2" at Fukui National College of Technology and aims to learn how instructions are processed and performed within the CPU and to employ programming using an assembly language.

Key words: Virtual computer system of "eAVR", Computer architecture, Assembler education

1. はじめに

今日の情報化社会の発展により、ユーザはコンピュータシステムをブラックボックスとして扱う世代になってきている。また「ユビキタス時代」という言葉にも現れているように、多種多様のシステムが身の回りに存在し実働している。

このような世の中では、教育の現場においてもソフトウェアの開発に関してハードウェアをそれほど意識しないC言語をはじめとして様々な高級言語が利用されるようになった。基本情報技術者試験でもCASLIIといったアセンブリ言語が用意されているが、これを選んで受検する学生はほとんどいない状況である。

しかし、従来、教育の現場ではアセンブリ言語のような低級言語によるソフトウェアの開発も盛んに行われ、そのプロ

グラミング教育も行われてきた。そのようなアセンブリ言語を用いたプログラミングをアセンブラ教育と呼ぶ。特に、CASLやCASLIIによるアセンブラ教育がよく行われ、それを通して、ハードウェアの動作に関する講義が行われ、ハードウェア内部の動作、命令の処理方法等を学んでいた。

従来行われていたCASLIIによるアセンブラ教育に代わり、かつ、「ユビキタス時代」に対応できるようなアセンブラ教育が必要である。つまりCASLIIのように仮想コンピュータシステムだけで終わるのではなく、実際のコンピュータシステムに対応できるものが望ましい。

そこで、本研究では1チップマイコンレベルでのアセンブラ教育を行い、将来的に組込み型コンピュータシステムへ発展できるように工夫した教育用システムの開発を行った。1チップマイコンとして現在世の中で使用されているマイコン

から、教育用に特化した形で縮小した仮想 CPU を考案することにした。ただし、プログラミングした結果が直ちに実際の CPU でも動作可能でなければならない。今回の対象として、H8、PIC、AVR 等が考えられるが、命令コード体系やアーキテクチャがシンプルである AVR を対象とすることにした。

そして、この簡略化した仮想 CPU システムを用いて実際にコンピュータのハードウェアの授業でアセンブラ教育を行った。

2. 教育用仮想コンピュータシステム eAVR(教育用 AVR-CPU)の仕様

今回考案した教育用仮想コンピュータシステムは AVR[1] を教育用に特化し、1 チップマイコンとして考案したものである。これを eAVR と呼ぶ。

2.1 従来の仮想システムの問題点

まず、アセンブラ教育を行う目的を整理した[2]。まず、第一に考えられるのは

- コンピュータシステムの仕様の理解。
対象とするシステムの動作、命令処理の様子を学ぶ。
ということである。そして、
- 高級言語（コンパイラ言語やインタプリタ言語）とアセンブリ言語の相違点の理解。
例:通常書いている C 言語のプログラムは CPU が動作するレベルまでどの程度分解されるのかを学ぶ。
などを学ぶ。それ以外にも言語教育という観点では
- アセンブリ言語プログラミングの作成方法を修得し、適用する。
- 演算処理、制御処理を行うプログラムの作成方法を修得し、適用する。
- 表を使った処理、入出力処理を行うプログラムの作成方法を修得し、適用する。
- スタック、及びスタックを用いたサブルーチンコールの仕組みと用法を修得し、適用する。

以上の目的を果たすために、従来は CASL II によるアセンブ

ラ教育が行われてきたが、

- CASL は命令数が多い
- ハードウェア的には可変 bit シフトを行うパレルシフタを用意し、大規模回路となる
- CASL II においてロードストア型アーキテクチャに対応するが、まだ不十分
- 現在主流のパイプライン処理には不向きな可変長命令の採用

など、ソフトウェア、ハードウェア両方からの問題点が考えられた。そこで、簡易 8bit CPU を提案しシミュレータ作成演習を行った[3]。このシステムを用いて 6 年間講義を行ってきた。しかし、以下のような問題点が残った。

- (1) あくまでも仮想計算機上のアセンブラ教育である。実際に動作させたという感動が無く、学生も机上のみで実体験できないことに不満を持った。
- (2) VHDL で設計し FPGA ボードで実機設計を行ったが、そのボードで講義を行うには台数が少なすぎる (FPGA ボードが高価なため)。

そして、単に仮想コンピュータシステムとして、机上チェックのみで終了してしまうものではなく、将来的に実機を用いて動作可能なシステムの導入が必要になってきた。

2.2 eAVR のハードウェア構成

このように将来的に実機を用いることを念頭に置き、かつ初心者理解しやすくハードウェアの動作理解にも役立つシステムとして 1 チップマイコンを導入することが考えられた。その代表として H8、PIC、AVR などが考えられたが、表 1 の比較表などを参考に AVR を選択した。

- アセンブラ教育が初心者にわかりやすい (レジスタ直交性が高くニモニックの書き換えて動作理解しやすい)
- ハードウェア教育を行うことで CISC ではなく RISC を選びたい、また将来的に実機チェックを行いたいので開発環境があること、C 言語の利用ができるものが望ましい。また、ハーバード・アーキテクチャの参考例、メモリマップ・ド・I/O の説明もしやすい。

ただし、通常の AVR でも規模が大きく、そのサブセットを考える必要がある。

図1に今回提案する仮想コンピュータシステム、1チップマイコン eAVR のブロック構成図を示す。基本的には市販されている TinyAVR2313 のサブセットとして考案した。

表1 1チップマイコンの比較

マイコン名	H8	PIC	AVR
アーキテクチャ	CISC	RISC	
		ハーバード・アーキテクチャ	ハーバード・アーキテクチャ
レジスタ数(個数)	少(8)	多(64) 汎用レジスタは少	多(32) レジスタ直交性高
命令数	約60	約30	約100
開発環境	有償	有償	無償 C言語も可能

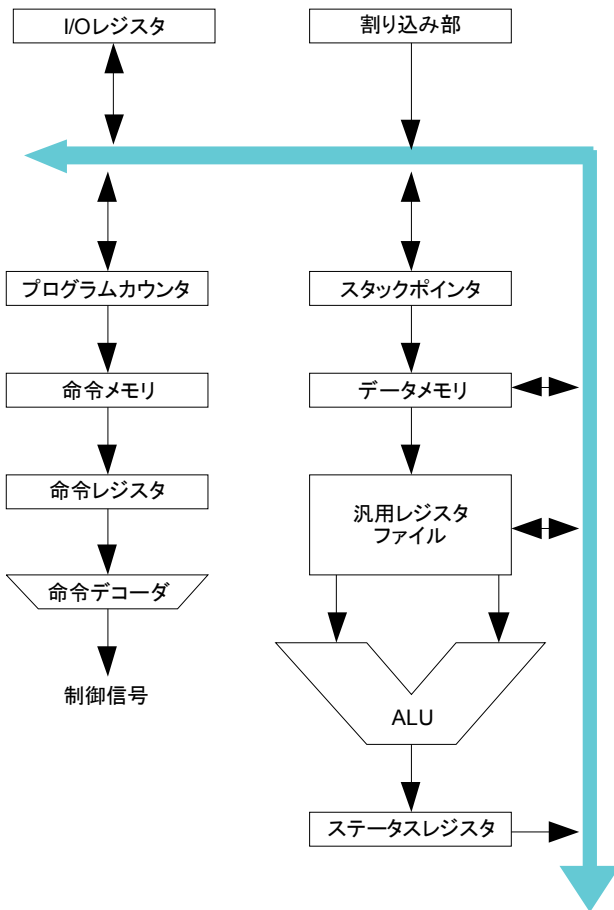


図1 eAVR ブロック図

実際に市販されている AVR の基本となるマイコンの TinyAVR2313 から複雑な I/O 類 (ウォッチドッグタイマ、通常のタイマ、UART など) を省き、I/O として1つの出入

力レジスタをサポートすることにした。実際のアプリケーションを作成する目的であれば、それらを省いた複雑な I/O 類も必要であるが、今回はハードウェア教育の一環として、コンピュータが命令処理を行う動作理解が第一目標であり、組み込み型コンピュータのプログラミングを学ぶことは2次のな目標である。また、この CPU は命令メモリ (プログラム用フラッシュメモリ) とデータメモリが分離しているハーバード・アーキテクチャを採用している。

2.3 基本命令セット

実際の AVR が用意している命令は 100 種類以上あり、初心者にとってすべてを使いこなすことは困難である。そこで、eAVR では命令数を大幅に削減した。命令は大きく分けると3通りに分類することができる。

- データ転送命令
- 演算命令
- 制御命令 (分岐命令など)

表2に今回用いた命令のニモニック表を示す。CISC の H8 に比べれば少ないが、実際の AVR は PIC に比べ命令数も多く100個以上存在する。そこで、CASL、CASLIIやPICの命令数に近い約30個に命令数を減らすことにした。

また、命令は固定長命令とし、メモリアドレスの指定は相対アドレス指定、またはレジスタ間接アドレス指定を行うようにした。授業では、直接指定、間接指定 (レジスタ、メモリ)、相対指定、ベース指定、修飾などのアドレッシングを学び、直接指定が初心者には動作の理解がしやすいが、オブジェクトに多くのビットが必要となるため固定長命令では直接指定ができない。そして、例外として認めることとし、なるべく使わないように机上でのプログラミング演習では指導した。

しかし、これもフリーのアセンブラである AVR studio を用いてラベルによる指定を行うことにより問題は生じない。むしろ、ラベルによる指定でわかりやすいプログラムを作成するための練習となる。

3. 授業での実践

今回、対象とした授業は本校電子情報工学科4年「計算機構成論Ⅱ」である。この科目は、

- (1) 計算機の歴史
- (2) ノイマン型コンピュータ
- (3) 高速化、高性能化技術

などを学ぶ講義である。この中で、ノイマン型コンピュータの動作理解をするための教材としてアセンブラ教育を取り入れている。昨年度まではCASLⅡあるいは8bit仮想コンピュータを用いて、アセンブラ教育を行っていた。

表2 eAVR ニモニク表

命令の種類	ニモニク	オペランド1	オペランド2	flag	
データ転送	mov	Rd	Rr	-----	$Rd \leftarrow Rr$
	movw	Rd	Rr	-----	$Rd+1:Rd \leftarrow Rr+1:Rr$
	ldi	Rd	K	-----	$Rd \leftarrow K$
	ld	Rd	Ri	-----	$Rd \leftarrow M[Ri]$
	lds	Rd	k	-----	$Rd \leftarrow M[k]$
	st	Ri	Rr	-----	$M[Ri] \leftarrow Rr$
	sts	K	Rr	-----	$M[k] \leftarrow Rr$
	in	Rd	P	-----	$Rd \leftarrow P$
	out	P	Rr	-----	$P \leftarrow Rr$
	push	Rr		-----	$STACK \leftarrow Rr$
	pop	Rd		-----	$Rd \leftarrow STACK$
演算命令	add	Rd	Rr	SVN ZC	$Rd \leftarrow Rd + Rr$
	adc	Rd	Rr	SVN ZC	$Rd \leftarrow Rd + Rr + C$
	sub	Rd	Rr	SVN ZC	$Rd \leftarrow Rd - Rr$
	sbc	Rd	Rr	SVN ZC	$Rd \leftarrow Rd - Rr - C$
	and	Rd	Rr	S0N ZC	$Rd \leftarrow Rd \text{ and } Rr$
	or	Rd	Rr	S0N ZC	$Rd \leftarrow Rd \text{ or } Rr$
	eor	Rd	Rr	S0N ZC	$Rd \leftarrow Rd \text{ eor } Rr$
	com	Rd		S0N Z1	$Rd \leftarrow \text{not } Rd$ ($Rd \leftarrow \$FF - Rd$)
	inc	Rd		SVN Z-	$Rd \leftarrow Rd + 1$
	dec	Rd		SVN Z-	$Rd \leftarrow Rd - 1$
	cp	Rd	Rr	SVN ZC	$Rd - Rr$
	lsl	Rd		SVN ZC	$Rd \leftarrow Rd \ll 1$
	lsr	Rd		SVN ZC	$Rd \leftarrow Rd \gg 1$ $Rd(7) \leftarrow 0$
	asr	Rd		SVN ZC	$Rd \leftarrow Rd \gg 1$ $Rd(7) \leftarrow Rd(7)$
	rol	Rd		SVN ZC	$Rd \leftarrow Rd \ll 1$ $Rd(0) \leftarrow C$ $C \leftarrow Rd(7)$
rор	Rd		SVN ZC	$Rd \leftarrow Rd \gg 1$ $Rd(7) \leftarrow C$ $C \leftarrow Rd(0)$	
制御命令	rjmp	K		-----	$PC \leftarrow PC + 1 + k$
	rcall	K		-----	$STACK \leftarrow PC, PC \leftarrow PC + k + 1$
	ret			-----	$PC \leftarrow STACK$
	brbs	S	k	-----	If $SREG(s) = 1$ then $PC \leftarrow k$ else $PC \leftarrow PC + 1$
	brbc	S	k	-----	If $SREG(s) = 0$ then $PC \leftarrow k$ else $PC \leftarrow PC + 1$
擬似命令	nop			-----	No operation

レジスタ類 Rd: ディスティネーションレジスタ Rr: ソースレジスタ
 他オペランド K: 8bit 即値 P: I/O ポート番号 s: ステータスレジスタビット番号
 フラグ S: サイン V: オーバフロー N: 負 Z: ゼロ C: キャリー

```

reset:  ldi    r16, 1
        ldi    r17, 1
        ldi    r19, 10
        ldi    r31, 0
        ldi    r30, $60
        st     z, r16
        inc   r30
        dec   r19
        st     z, r16
        inc   r30
        dec   r19
        mov   r18, r16
loop:   add   r18, r16
        mov   r16, r17
        mov   r17, r18
        st     z, r17
        inc   r30
        dec   r19
        brbc  1, loop
fin:    rjmp  fin

```

図2 eAVR のプログラム例 (フィボナッチ数列)

CASL II にはすでにシミュレーション可能な環境が世の中にあり、後者の 8bit 仮想コンピュータにはマイクロソフト社の EXCEL マクロでアセンブラ、シミュレータ環境を構築していた。また、このシステムはその発展として、実験で VHDL というハードウェア記述言語を用いてその CPU を構築することを行っていた。

今回はそれらの授業に代わり、eAVR を用いて以下の手順でアセンブラ教育を行った。

(1) システム構成の理解

それまで学んだ計算機構成に照らし合わせ、図1のeAVR ブロック図を元にシステム構成を理解する。

(2) ニモニックと処理手順の理解

ニモニックに対応する機械語をどのようにフェッチし、どのようにデコードし、どのように実行するか、その処理手順の理解を RTL で記述し、各構成部の役割を学ぶ。

(3) プログラミング演習

図2に示されたようなプログラムを作成し動作確認を行うなどのプログラミング演習を行った。

また、これとは別件として以下の開発を行っている。

(4) ハードウェアによる実現 (開発中)

卒業研究において、単に仮想コンピュータシステムとしてではなく、VHDL (VHSIC Hardware Description Language) を用いて開発し、実際に FPGA (Field Programmable Gate Array) に実装する研究を行っている。

今後の予定としては、高価な FPGA による実機チェックではなく、市販の AVR を搭載した実験ボードを作成して、USB 接続でパソコンにつなぎ、前述の開発ツール AVR studio を用いて、C 言語またはアセンブリ言語でのプログラム開発を行い、AVR に実装して動作チェックを行うといった組み込み型コンピュータの開発を学ぶといった授業に発展させ、世の中のニーズに対応できる学生の育成を検討している。

4. おわりに

今年度このシステムをはじめて授業内で使用しアセンブラ教育を行った。その結果、昨年度と今年度で学生が異なるため単純比較はできないが、このシステムが少しは役割を果たしていると考える。しかし、このシステムは今後、実機でのチェックができるようになり、各自が自由に演習できる環境を整えることができるように検討する予定である。

5. 参考文献

- [1] Atmel Co. 「8-bit AVR Instruction Set」 (2008)
- [2] 独立行政法人 情報処理推進機構 IT 人材育成本部 情報処理技術者試験センター、「基本情報技術者試験 (レベル 2) シラバス Ver.1.0」 p.12 (2008)
- [3] 青山義弘他、「CPU アーキテクチャ教育へのコンピュータの利用」、平成 14 年度情報処理教育研究集会講演論文集 (2002)