



Tokyo Gakugei University Repository

東京学芸大学リポジトリ

<http://ir.u-gakugei.ac.jp/>

Title	教育を目的としたPICアセンブリ言語プログラミングシステムの開発
Author(s)	山田, 朗; 鄒, 松騰
Citation	東京学芸大学紀要. 第6部門, 技術・家政・環境教育, 55: 31-36
Issue Date	2003-11-28
URL	http://hdl.handle.net/2309/2972
Publisher	東京学芸大学紀要出版委員会
Rights	

教育を目的とした PIC アセンブリ言語 プログラミングシステムの開発

山田 朗*・鄒 松騰**

工 学

(2003年7月31日)

YAMADA, A. and SU, S. : Development of PIC Assembly Language Programming System for Educational Purpose.
Bull. Tokyo Gakugei Univ. Sect. 6, 55 : 31 - 36 (2003) ISSN 1341 - 1705

Abstract

In this study, a programming environment for PIC microprocessor assembly language were developed. The system is especially designed for educational purpose, and contains a source editor, assembler, debugger and simulator. With simulator's [advanced mode], user can verify the real time progressing of registers and data memories inside the CPU.

(in Japanese)

Key words : PIC microprocessor, assembly language, programming environment

Department of Engineering, Tokyo Gakugei University, Koganei-shi, Tokyo 184-8501, Japan.

1. はじめに

大学や専門学校でコンピュータによる制御を教授する際には、Microchip Technology 社の PIC マイクロコントローラ¹⁻⁴⁾(以下 PIC と称す)を組み込んだロボットを教材として用いることが多い。PIC は安価な CPU であり、EEPROM メモリと入出力回路が内蔵されているなど構造が単純で扱いやすく、数多くの教材ロボットが開発されている。

これらの教材のほとんどは、専用のアプリケーションを用いて PIC アセンブリプログラムを作成し、パソコンのシリアルまたはパラレルインターフェイスを介してロボットに書き込む形式のものが多く、PIC のアセンブリ言語にまで踏み込んだものはない。これは、アセンブリ言語が、C や BASIC など他の高級言語に比べ初学者になじみにくく、プログラミング言語のひとつとして教授するには不適切であると考えられているためである。確かにアセンブリ言語は C や BASIC などの高級言語に比べ、CPU のメーカーや種類に強く依存し、汎用性に大きく欠ける。また、アセンブリ言語を十分に理解するためには、CPU の内部構造、中央演算装置や入出力ポートの仕組みなど、ハードウェアについての深い知見も必要である。言語の仕様も柔軟性に欠き、例えば16進数を自在に操れるだけのスキルが必要となる。

* 東京学芸大学 (184 8501 小金井市貫井北町4 1 1)

** 東京学芸大学大学院

しかし、アセンブリ言語はCPUに対してもっとも基本的な言語であり、アセンブリ言語が習得できればCPUの内部構造や制御の仕組みをより深く理解することが可能となる。

本研究では、視覚的にわかりやすく、初学者にも簡単に操作できる教材用のPICアセンブリ言語プログラミングシステムを構築した。このシステムは、エディタ、アセンブラ/デバッガ、シミュレータを統合したシステムであり、誰でも簡単にアセンブリ言語が理解できるように配慮されている。特にシミュレーション部では、アセンブリ命令の動作、レジスタ内のデータの変化などをアニメーション表示することによって、CPU内部の処理過程を学ぶことができる。

2. 本システムの概要

PICアセンブリ言語の開発環境としては、Microchip Technology社のフリーウェアMPLAB³がよく知られている。MPLABはアセンブラ・シミュレータを統合したアセンブリ言語開発環境であるが、プログラマを対象に開発されたものであり、教育用途には適していない。本システムでは、初学者でも簡単に扱えるようユーザーインターフェイスに配慮しながら、MPLABの持つ最低限の機能を実装した。主な機能は以下のとおりである。

エディタ機能

アセンブリプログラムを自由入力できる。PICアセンブリ命令の詳細を調べながら、スムーズなプログラム作製が可能である。

アセンブラ/デバッガ機能

アセンブリ言語を任意の時点でアセンブルすることが可能である。文法上のエラーをデバッグすることができる。

シミュレーション機能

処理速度を重視し、プログラム全体の動作を確認できる「通常モード」と、アセンブリ言語・CPUの内部構造の学習のための「拡張モード」がある。「拡張モード」では、アセンブリ命令によりPIC内部の中央演算部がどのように演算を実行するかをアニメーションで表示する。また、プログラムの実行とともにPIC内部の各レジスタやメモリがどのように変化していくかを表示する。

システムの作成にはVisual BASIC 5.0を用いている。対象となるOSはWindows 98以降である。

3. 本システムのユーザーインターフェイス

システムのユーザーインターフェイスを図1に示す。システムは5つのウィンドウから構成されている。「実行状態ウィンドウ」(図2)は利用者が操作をおこなうウィンドウであり、また、PICの入出力ポート、実行中の命令などが表示される。「編集ウィンドウ」(図3)はアセンブリプログラム編集のためのウィンドウ、「プログラムメモリウィンドウ」(図4)はアセンブルされたバイナリ命令が表示される。「データメモリウィンドウ」は(図5)、PICの動作を制御する2つのバンクメモリの内容がリアルタイムに表示される。また、「内部構造ウィンドウ」(図6)は、CPU内部のワーキングレジスタ、演算処理部などの状態をリアルタイムに表示する。このほか、プログラム作成の補助として、現在実行中のアセンブリ命令を表示する「命令解説説明ウィンドウ」(図7)、Microchip Technology社のPIC仕様書を表示する「命令詳細ウィンドウ」(図8)がある。

4. システムの構成

図9に、本システムの流れを示す。

4.1. ソースプログラムのアセンブルとデバッグ

利用者はまず、システムの「編集ウィンドウ」(図3)を用いて、アセンブリ言語のソースを入力する。他のファイルからの読み込み/書き込みも可能である。ソース入力時には、「命令詳細ウィンドウ」(図8)を

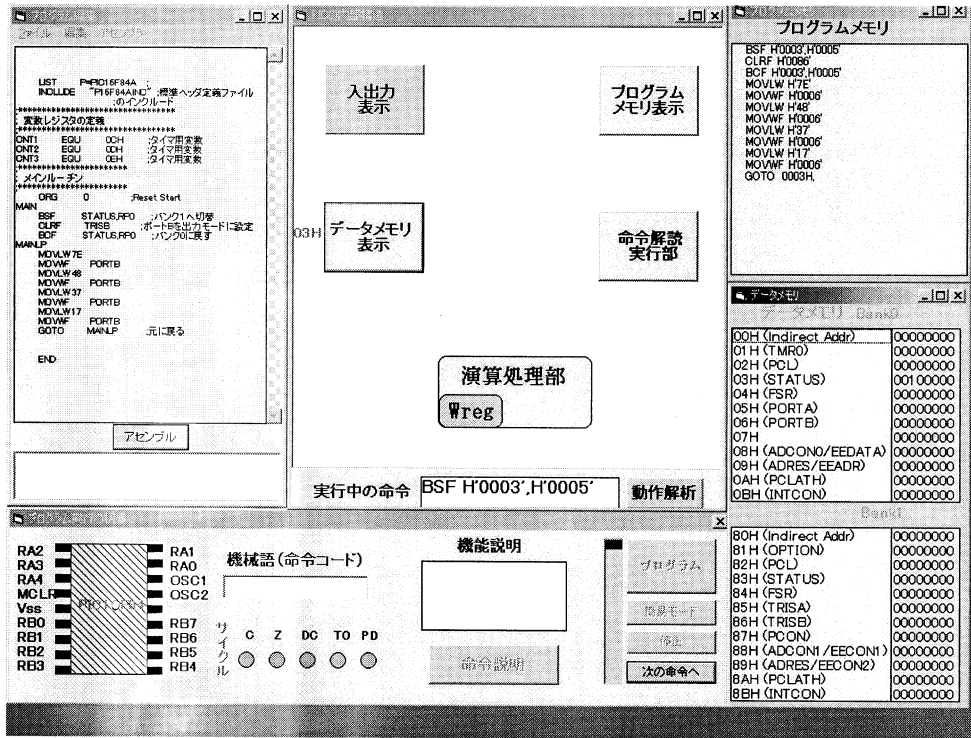


図1 システムのインターフェイス

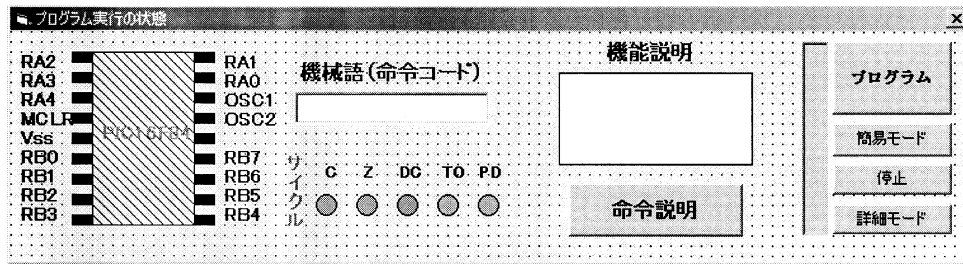


図2 実行状態ウィンドウ

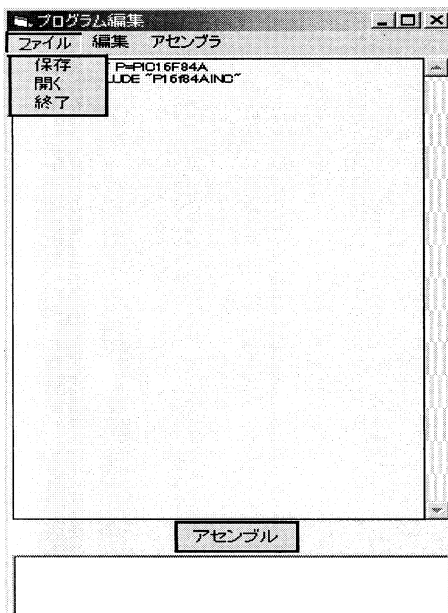


図3 編集ウィンドウ



図4 プログラムメモリウィンドウ

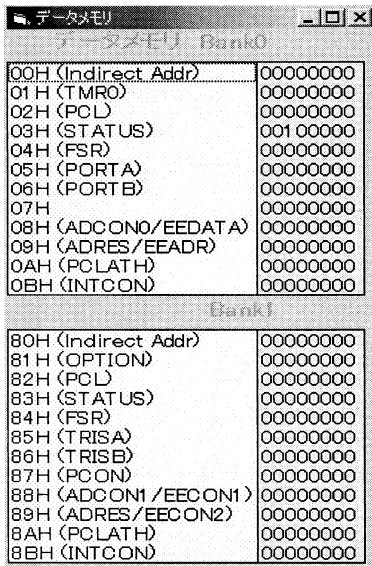


図5 データメモリウィンドウ

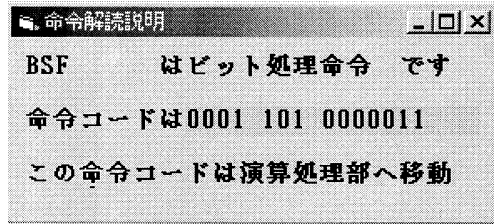


図7 命令解説説明ウィンドウ

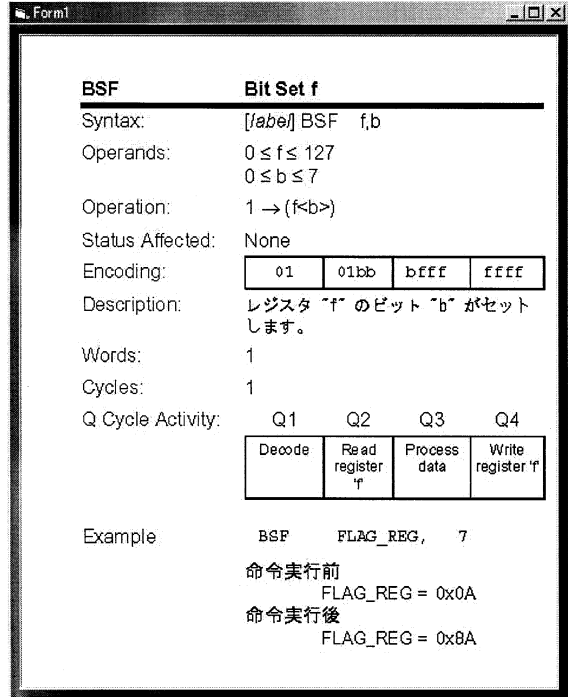


図8 命令詳細ウィンドウ

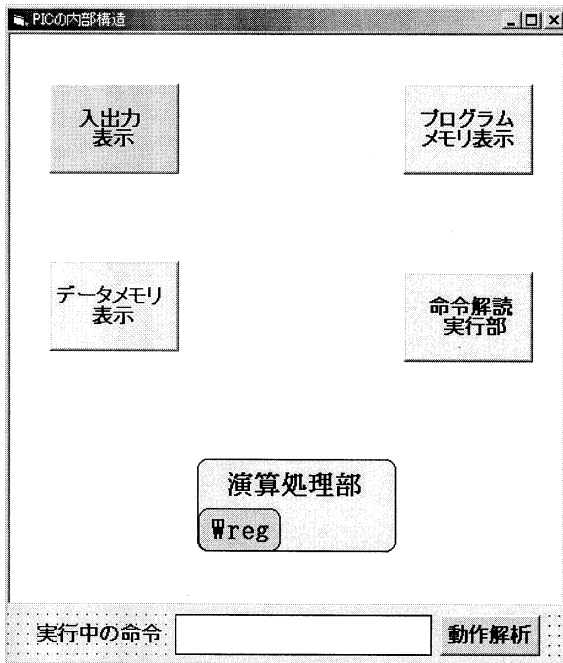


図6 内部構造ウィンドウ

索して、必要な命令の仕様を閲覧することができる。開発したソースプログラムをアセンブルして、バイナリプログラムにする。この過程でソースプログラムの構造がチェックされ、デバッグが可能である。アセンブルエラーがない場合には、バイナリプログラムが「プログラムメモリウィンドウ」(図4)に保存される。「プログラムメモリウィンドウ」

では、表紙形式を16進数形式、アセンブリ命令形式に切り替えることができる。

4.2. シミュレーション

アセンブルされたプログラムをシミュレーションしてプログラムの動作確認をする。シミュレーションには「通常モード」と「拡張モード」があり、通常モードではプログラム全体の動作確認を、拡張モードではアセンブリ命令一つ一つの詳細な動きを確認しながらプログラムを実行することができる。

1) 通常モード

視覚的な解説やアニメーションを省略して、高速で実行する。実行中には、「実行状態ウィンドウ」(図2)の入出力ポートの状態のみを更新する。プログラムの全体的な動作確認のために用いられるモードである。

2) 拡張モード

プログラムの命令一つ一つについて、演算処理の過程をアニメーションで表示しながら実行する。各命令について、以下の処理をおこなう。

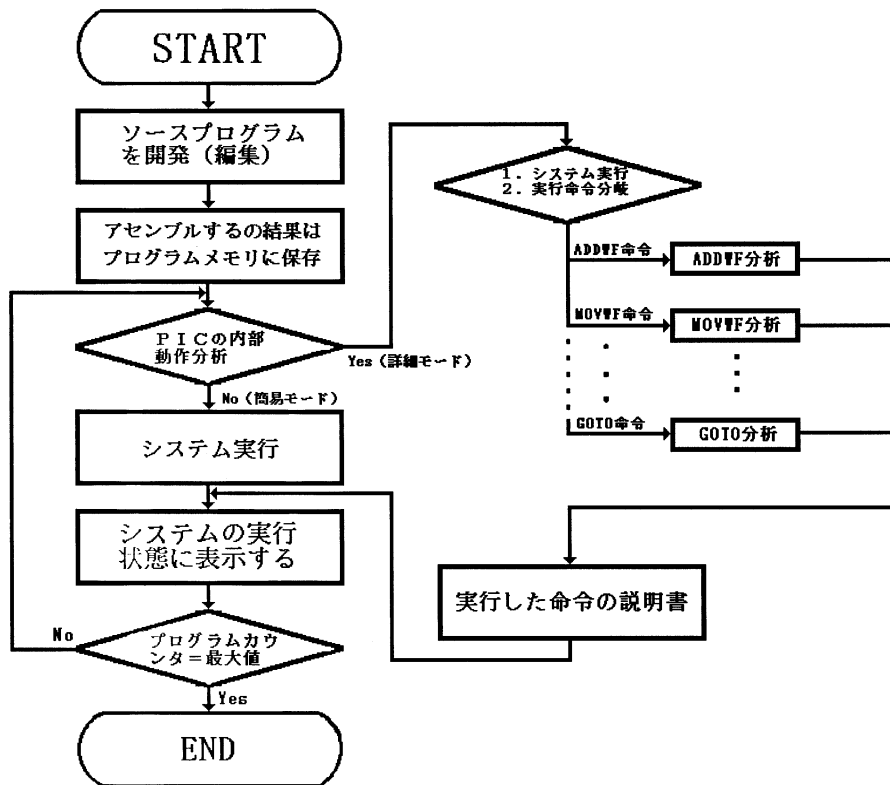


図9 システムの流れ

- (1) 「プログラムメモリウィンドウ」(図4)の実行命令リストから命令を1個命令解読部に移動して、命令の種類を判別する。この時点で、「命令解読説明ウィンドウ」(図7)に実行中の命令の説明が表示される。
 - (2) 処理に必要な数値データをワーキングレジスタまたはデータメモリから取り出し、演算処理部に移動して演算処理をおこなう。必要に応じ、「データメモリウィンドウ」(図5)内のデータメモリの内容が更新される。CPUの各フラグの状態を「実行状態ウィンドウ」(図2)に反映する。
 - (3) 処理の結果を適切にデータメモリまたはワーキングレジスタに移動して格納する。必要に応じ、「データメモリウィンドウ」(図5)内のデータメモリの更新をおこなう。
 - (4) データメモリの状態に応じ「実行状態ウィンドウ」(図2)の入出力ポートの状態を更新する。
- 以上の各過程では、命令やデータの取り出し/移動/格納はすべて「内部構造ウィンドウ」(図6)上でアニメーション表示される。

5. おわりに

本研究では、初学者への教育を目的に、PICアセンブリ言語プログラミングシステムを構築した。このシステムは、エディタ、アセンブラ/デバッガ、シミュレータから構成され、シミュレータの「拡張モード」では、アセンブリ命令の動作、レジスタ内のデータの変化などをリアルタイムに確認することができる。

参考文献および URL

- 1) 後閑哲也：電子工作のための PIC 活用ガイドブック，株式会社技術評論社，(2000)
- 2) PIC16F8X, Microchip Technology Incorporated, Printed in the USA, (2002)
<http://www.microchip.com/download/lit/pline/picmicro/families/16f8x/30430c.pdf>
- 3) MPASM USER'S GUIDE with MPLINK and MPLIB, Microchip Technology Incorporated

<http://www.microchip.com/download/tools/picmicro/code/mpasm/33014g.pdf>

4) インターネット・電子工作の実験室ホームページ「PICの基礎」

<http://www.picfun.com/>