

コンピュータ工学 講義プリント(10月23日)

教科書で扱っている PIC16F84A というマイコンは、搭載メモリの少なさなどの性能上の理由から、C 言語などの高級言語でプログラムを開発する事は少なく、アセンブリ言語でプログラム開発することが多い。今回は、PIC マイコンのアセンブリ言語の概要について説明する。

・機械語とアセンブリ言語 (教科書 P.60 参照)

9月4日の講義で説明したとおり、CPUはC言語などの高級言語のプログラムを理解することはできない。CPUが直接理解できる言語を機械語あるいはマシン語という。

PIC16F84Aの場合、機械語は14ビットの2進数からなる命令を集めたものである。すなわち、機械語は、非常に多くの桁の2進数の数列という事になる。

人間が2進数の命令を記憶するのは非常に困難なので、通常は、動作を表わす数文字のアルファベットの単語で命令を表わして、機械語のプログラムを作成する。この数文字のアルファベットの事をニーモニックと呼ぶ。例えば、教科書 P.70 には MOVF という 1 バイトの転送命令の説明が載っているが、この MOVF がニーモニックである。

直接2進数で機械語の命令を記述するのではなく、ニーモニックを使って、人間の理解がしやすい方法で機械語の命令を記述するタイプの言語をアセンブリ言語と呼ぶ。(教科書ではアセンブリ言語の事をアセンブラ言語と呼んでいるが、一般的ではないので注意)

アセンブリ言語で記述されたプログラムを、2進数の機械語に変換する作業の事をアセンブルと呼ぶ。また、アセンブリ言語を自動的に機械語に変換するソフトウェアの事をアセンブラとよぶ。

・命令と擬似命令 (教科書 P.60 参照)

アセンブリ言語では、機械語の命令に対応したニーモニック以外にも、アセンブラに対して色々な指示をするための擬似命令を使う。PIC16F84Aの全命令のニーモニックの一覧表は教科書 P.61 の表 4.1 に、また MPASM というアセンブラの擬似命令の一覧表は教科書の表 4.2 に載っている。

・アセンブリ言語の書式 (教科書 P.61 参照)

アセンブリ言語の記述の仕方は、アセンブラによって多少異なるが、ここでは MPASM というアセンブラに限定して説明する。

アセンブリ言語は、多数の行で記述される。1行に記述できる命令(擬似命令を含む)は、最大1命令である。

1行は、(1)ラベル欄、(2)オペコード欄、(3)オペランド欄、(4)コメント欄の順に、いくつかの欄が並んでいる。欄の区切りには、1つ以上のスペースやタブを用いる。(1)~(4)の欄の内いくつか(あるいは全部)が省略される事がある。ラベル欄は、行最初から始まる。ラベル欄を省略してオペランド欄からはじめる場合は、行の最初に一つ以上のスペースまたはタブを挿入する。

次に、(1)~(4)の欄について、説明する。

(1) ラベル欄

ラベルは、メモリのアドレスまたは数値定数につける名称である。ラベルを使えば、ジャンプ先の命令のアドレスや、データ格納先の RAM のアドレスなどを自分で数えて数値で記述する必要はなくなり、具

体的な数値の管理はアセンブラに任せられるようになる。

ラベルは、英数字、アンダーバー()、クエスチョンマークからなる 32 文字以内の名称である。ただし、ラベルの一文字目はアルファベットかアンダーバーでなければならない。大文字と小文字は区別される。

(2) オペコード欄

PIC の命令のニーモニックや、アセンブラの擬似命令を記述する。

(3) オペランド欄

操作対象とするレジスタやビット、数値定数などを記述する。命令によってはオペランド欄を必要としないものもある。

(4) コメント欄

プログラムを読む人が理解しやすいように記述する注釈を書く欄である。行内にセミコロン(;)を書くと、それ以降はコメント欄として扱われる。

アセンブリ言語で記述されたプログラム内に、例えば次のような行があったとしよう。

```
LOOP1   BSF     STATUS, RPO ;バンク 1 を選択
```

この場合、”LOOP1”がラベル欄に、”BSF”がオペコード欄に、”STATUS, RPO”がオペランド欄に、”;バンク 1 を選択”がコメント欄になる。

教科書 P.64 のリスト 4.1 には LED 点灯プログラムをアセンブリ言語で記述した例が載っている。このリストの最初の行は、

```
; *****
```

となっているが、行がセミコロンで始まっているので、行全体がコメント欄である。(つまりラベル欄、オペコード欄、オペランド欄がない)

リスト 4.1 の 6 行目(”初期設定”の次の行)は、空白行になっているが、ラベル欄、オペコード欄、オペランド欄、コメント欄の全てがない行である。

リスト 4.1 の 12 行目は

```
ORG     0           ; プログラムを格納する先頭アドレスの指定
```

となっているが、”ORG”がオペコード欄、”0”がオペランド欄、”; プログラムを格納する先頭アドレスの指定”がコメント欄となっている。(ラベル欄がない)

リスト 4.1 の 23 行目は

```
END     ; プログラムの終わり
```

となっているが、”END”がオペコード欄、”; プログラムの終わり”がコメント欄となっている。(ラベル欄とオペランド欄がない)

オペコード欄の命令または擬似命令の種類により、オペランド欄が必要かどうかが決まっている。ORG という擬似命令は、オペランド欄が必要であるが、END という擬似命令はオペランド欄が不要である。

・プログラムの基本スタイル (教科書 P.64 参照)

アセンブリ言語でプログラムを開発する場合、最初に使用する PIC マイコンの種類を指定したり、使用する設定ファイルを指定したり、プログラムの先頭アドレスを指定したりと、決まったスタイルでプログラムを書き始める。詳しくは、教科書 P.64~65 を参照する事。

・数値定数の記述方法

MPASM で数値定数を記述する場合、10 進数、16 進数、8 進数、2 進数、ASCII コードの各形式で記述する事ができる。それぞれの記述法で数値定数を記述する具体的な方法を、表 1 に示す。

表 1、数値定数の記述方法

記述形式	記述方法	例	(参考)C 言語の場合
10 進数(Decimal)	D’<10 進定数>’	D’123’	123
16 進数(Hexadecimal)	H’<16 進定数>’ 0<16 進定数>H	H’A3’ 0A3H	0xa3
8 進数(Octal)	O’<8 進定数>’	O’123’	0123
2 進数(Binary)	B’<2 進定数>’	B’01101110’	
ASCII コード	’<文字>’ A’<文字>’	’C’ A’C’	’C’