情報処理の基礎・第 13 回

2013 年 12 月 18 日

今回の内容

13.1 機械語命令と機械語プログラム

CPU はメモリに格納されている機械語命令を (アドレス順に)1 つ 1 つ実行していきます。1 つの機械語命令は、数 bit から数十 bit の大きさのビット列で表現されており、どのようなビット列であれば、どのような作業を行うかについての約束事が CPU 毎にあらかじめ決められています。CPU が実行できる機械語命令として、最低でも次のような機能を持つ命令群が用意されているのが普通です。

- ロード命令 — メモリ中の特定のアドレスのデータを (キャッシュメモリを介して) CPU の特定のレジスタにコピーします。また、特定のアドレスのデータではなく、(その機械語命令の一部に埋め込まれている) 定数 (特定の 0/1 のパターン) をレジスタに格納する命令もあります。
- ストア命令 — CPU の特定のレジスタに記憶されているデータを (キャッシュメモリを介して) メモリ中の特定のアドレスにコピーします。
- 演算命令 — レジスタに記憶されているデータの間で四則演算などの計算を (ALU を使って) 行い、その計算結果を特定のレジスタに格納します。
- 分岐命令 — 今実行した機械語命令の次のアドレスに格納されている命令に進むのではなく、別の離れたアドレスから機械語命令の実行を続けます。無条件に分岐を行う命令以外にも、先に実行された演算命令の結果に基づいて、特定の条件 (計算結果が 0 であった、負であったなど) が成り立ったときだけ分岐する命令も用意されています。

C プログラムと機械語プログラム

C 言語も、機械語と同じようにプログラミング言語のひとつではありますが、そのままでは、CPU が内容を理解して、その指示を実行することはできません。C 言語で書かれたプログラムは、コンパイラ[1]によって機械語プログラムに変換されることで、初めて CPU が理解可能な (実行できる) ものになります。

たとえば、右の C プログラム (の断片) は、次ページのような機械語プログラムに変換できれば、CPU が相応の仕事を行うことができるようになります。本来、機械語は CPU によって異りますが、ここでは仮想的な 32 bit CPU の機械語で書いたプログラムを紹介しています。この 32 bit CPU は、(少なくとも) `A` と `B` という 2 つの 32 bit の汎用レジス

```
 ⋮
int i, sum;

sum = 0;
for (i = 1; i <= 10; i++)
    sum = sum + i;
 ⋮
```

[1]情報処理実習室の Linux 環境での cc コマンドがこれにあたります。

タを持っており、メモリのアドレスや定数、レジスタを指定しての定数ロード命令(LDI)、ロード命令(LD)、ストア命令(ST)、ALU によるレジスタ間の加減乗除命令(ADD や SUB)が実行できるようになっています。その他、分岐先番地を指定して無条件の分岐を行う命令(JAL)や、先に実行した引き算の結果が負のときだけ分岐を行う命令(JLT)などの分岐命令[2]も備えています。

本来の機械語命令は 1 〜数 byte のビット列として表現されるわけですが、ここでは、命令の種類や、レジスタの指定、メモリアドレスの指定が何であるかを人間が理解しやすいような記法で書いてあります。この機械語プログラムは、C プログラムの変数 i と sum の値を、それぞれ、メモリの 1200 番地と 1204 番地に、32 bit の符号付き整数として記憶していることに注意してください。

機械語命令のアドレス	機械語命令	機械語命令の意味
⋮	⋮	⋮
100	LDI A, 0	レジスタ A に定数 0 を格納(定数ロード命令)
104	ST A, 1204	レジスタ A の内容をメモリの 1204 番地(変数 sum)にコピー(ストア命令)
108	LDI B, 1	レジスタ B に定数 1 を格納(定数ロード命令)
112	ST B, 1200	レジスタ B の内容をメモリの 1200 番地(変数 i)にコピー(ストア命令)
116	LDI A, 10	レジスタ A に定数 10 を格納(定数ロード命令)
120	SUB A, B	レジスタ A からレジスタ B の値を引く(演算命令)
124	JLT 152	引いた結果が負ならば、152 番地にジャンプ(分岐命令)
128	LD A, 1204	メモリの 1204 番地の内容(変数 sum)をレジスタ A にコピー(ロード命令)
132	ADD A, B	レジスタ A にレジスタ B の値を足し込む(演算命令)
136	ST A, 1204	レジスタ A の内容をメモリの 1204 番地(変数 sum)にコピー(ストア命令)
140	LDI A, 1	レジスタ A に定数 1 を格納(定数ロード命令)
144	ADD B, A	レジスタ B にレジスタ A の値を足し込む(演算命令)
148	JAL 112	常に 112 番地にジャンプ(分岐命令)
152		
⋮	⋮	⋮

13.2 演習問題

1. 上の機械語プログラムの例を参考にして、次の C プログラム(の断片)に対応する(同じ CPU の)機械語プログラムを書きなさい。ただし、変数 x、y、z の値は、それぞれ 1000 番地、1020 番地、1040 番地に、32 bit の符号付き整数として記憶しているものとする。

```
z = x + y;
y = y + 1;
```

2. 同様に、次の C プログラム(の断片)に対応する(同じ CPU の)機械語プログラムを書きなさい。機械語命令 SUB A, B は、2 つのレジスタ A、B に対して、「A = A - B;」に相当する処理を行うことに注意すること。

```
if (x < y)
    z = y - x;
else
    z = x - y;
```

情報処理の基礎・第 13 回・終り

[2]他に、引き算の結果が正のときだけ分岐を行う JGT、0 のときだけ分岐を行う JE、0 でないときだけ分岐を行う JNE、0 以上のときだけ分岐を行う JGE、0 以下のときだけ分岐を行う JLE などもあるはずです。