

シフト命令

山本昌志*

2004年10月22日

1 前回の復習と本日の学習

1.1 復習

前回は、比較命令とジャンプ命令を学習した。これらを組み合わせて、FORTRAN でいうところの、IF と GO TO に相当する構文ができることを理解したはずである。

1.2 本日の学習内容

本日は、シフト命令を学習する。これは、乗算や除算を行うときに使う。CASL II のシフトめいれいは、次の通りである。

- レジスタの内容をシフトさせることにより、 2^n 倍したり、 2^{-n} 倍する。レジスタの内容は、符号付き整数として取り扱われる

算術左シフト	SLA	レジスタの内容を符号ビットを除き左にシフト。空きには0が入る。
算術右シフト	SRA	レジスタの内容を符号ビットを除き左にシフト。空きには符号ビットが入る

- レジスタの内容をシフトさせることにより、 2^n 倍したり、 2^{-n} 倍する。レジスタの内容は、符号なし整数として取り扱われる

論理左シフト	SLL	1 語全てを、左にシフト。空きには0が入る
論理右シフト	SRL	1 語全てを、右にシフト。空きには0が入る

*国立秋田工業高等専門学校 電気工学科

2 ビットシフトの意味

2.1 基数 N の場合

10進数を10倍、100倍、1000倍、…するのは簡単である。 10^n 倍するためには、左にゼロを n 個付けば良い。これは、左シフトである。同様に $1/10$ 倍、 $1/100$ 倍、 $1/1000$ 倍、…するのは簡単である。 10^{-n} 倍するためには、小数点の位置を n 個左に寄せれば良い。これは右シフトである。

16進数の場合も同じである。たとえば、 $(EF35)$ を $(16)_{10} = (10)_{16}$ 倍や $(16)_{10}^2 = (10)_{16}^2$ 倍、 $(16)_{10}^{-1} = (10)_{16}^{-1}$ 倍や $(16)_{10}^{-2} = (10)_{16}^{-2}$ 倍すると

$$\begin{aligned}(EF35)_{16} \times (16^2)_{10} &= (EF35)_{16} \times (100)_{16} \\ &= (EF3500)_{16} \\ (EF35)_{16} \times (16)_{10} &= (EF35)_{16} \times (10)_{16} \\ &= (EF350)_{16} \\ (EF35)_{16} \times (16^{-1})_{10} &= (EF35)_{16} \times (0.1)_{16} \\ &= (EF3.5)_{16} \\ (EF35)_{16} \times (16^{-2})_{10} &= (EF35)_{16} \times (0.01)_{16} \\ &= (EF.35)_{16}\end{aligned}\tag{1}$$

となる。やはり、右や左にシフトさせれば良い。

2進数の場合も全く同じである。この場合、 2^n の計算が簡単である。 n が正の整数の場合、左に n ビットシフトさせる。一方、 n が負の整数の場合、 n の絶対値分、右にシフトさせる。

$$\begin{aligned}(110011)_2 \times (2^2)_{10} &= (110011)_2 \times (100)_2 \\ &= (11001100)_2 \\ (110011)_2 \times (2)_{10} &= (110011)_2 \times (10)_2 \\ &= (1100110)_2 \\ (110011)_2 \times (2^{-1})_{10} &= (110011)_2 \times (0.1)_2 \\ &= (11001.1)_2 \\ (110011)_2 \times (2^{-2})_{10} &= (110011)_2 \times (0.01)_2 \\ &= (1100.11)_2\end{aligned}\tag{2}$$

2.2 CASL II の場合

2.2.1 ビットシフト

CASL II で取り扱う 16 ビットの整数を 2^n 倍する事を考える。もし、その 16 ビットが正で有れば、それは簡単である。先に示したように、 n ビット右や左にシフトさせれば良い。

問題は、符号付き整数で、第 15 ビットが 1 の負の場合である。これは、実例を示した方が分かりやすい。たとえば、 $(-12)_{10}$ を 2 倍と 4 倍する事を考える。2 倍すると $(-24)_{10}$ で、4 倍すると $(-48)_{10}$ である。そ

れぞれを、2の補数で取り扱おうと、となる。

$$(-12)_{10} \rightarrow (1111\ 1111\ 1111\ 0100)$$

$$(-24)_{10} \rightarrow (1111\ 1111\ 1110\ 1000)$$

$$(-48)_{10} \rightarrow (1111\ 1111\ 1101\ 0000)$$

従って、CASL IIの符号付き16ビット整数の場合、 2^n する場合は、左に n ビットシフトさせて、空いたビットに0を入れれば良い。

次に、 $1/2$ 倍と $1/4$ 倍する事を考える。すると

$$(-12)_{10} \rightarrow (1111\ 1111\ 1111\ 0100)$$

$$(-6)_{10} \rightarrow (1111\ 1111\ 1111\ 1010)$$

$$(-3)_{10} \rightarrow (1111\ 1111\ 1111\ 1101)$$

となる。この場合も右に n ビットシフトさせれば良いのであるが、空いたビットには1を入れなくてはならない。

ポイント

- 符号無し整数の場合
 - 2^n 倍する場合、左に n ビットシフトさせて、空いたビットに0を入れればよい。
 - 2^{-n} 倍する場合、右に n ビットシフトさせて、空いたビットに0を入れればよい。
- 符号付き整数の場合
 - 2^n 倍する場合、左に n ビットシフトさせて、空いたビットに0を入れればよい。
 - 2^{-n} 倍する場合、右に n ビットシフトさせて、空いたビットには符号ビットを入れればよい。

2.2.2 端数の処理

CASL IIの整数をビットシフトを用いて2で割ったりすると、端数(小数部)が生じる。この端数は、16ビットを越えるので、無視される。ここで、商が切り上げなのか切り下げなのか、疑問が発生する。これについても、実際の整数で考える。

(5)₁₀ と (-5)₁₀ を 1 ビット右にシフトさせて、1/2 倍してみる。

$$\begin{aligned}(5)_{10} &\rightarrow (0000\ 0000\ 0000\ 0101) \\ &1\text{ ビット右} \rightarrow (0000\ 0000\ 0000\ 0010) \rightarrow (2)_{10} \\ (-5)_{10} &\rightarrow (1111\ 1111\ 1111\ 1011) \\ &1\text{ ビット右} \rightarrow (1111\ 1111\ 1111\ 1101) \rightarrow (-3)_{10}\end{aligned}$$

この結果から、以下のようにまとめることができる。

ポイント

- 正の整数の場合、端数は切り下げとなる。
- 負の整数の場合、端数は切り上げとなる。

3 算術シフト 命令

算術シフト命令は、左シフト (SLA) と右シフト (SRA) の 2 つがある。いずれも、符号付き整数を取り扱い、前者は 2^n 倍、後者は 2^{-n} 倍する。算術シフト命令は、符号も考慮していることが重要である。

3.1 算術左シフト (SLA)

3.1.1 内容

命令語	SLA
語源	Shift Left Arithmetic (shift:移す left:左 arithmetic:算術)
役割	レジスタの内容を n ビット左に移動させる。空いたビットには、0 が入る。これは、符号付き整数を 2^n 倍しているのと同じ。
書式	教科書 (p.59) の通り。第一オペランドは汎用レジスタ。第二オペランドはアドレス。
機能	教科書 (p.59) の通り
フラグレジスタ	教科書 (p.59) の通り。

この命令は、符号付き整数を 2^n 倍する。従って、シフトにより空いたビットには、0 が入る。

3.1.2 使用例

```
SLA  GR0,2      ;GR0 の内容を 2 ビット左へシフト
SLA  GR0,0,GR1  ;GR0 の内容を GR1 の値、左へシフト
```

3.2 算術右シフト (SRA)

3.2.1 内容

命令語	SRA
語源	Shift Right Arithmetic (shift:移す right:右 arithmetic:算術)
役割	レジスタの内容を n ビット右に移動させる。空いたビットには、符号ビットが入る。これは、符号付き整数を 2^{-n} 倍しているのと同じ。
書式	教科書 (p.62) の通り。第一オペランドは汎用レジスター。第二オペランドはアドレス。
機能	教科書 (p.62) の通り
フラグレジスタ	教科書 (p.62) の通り。

この命令は、符号付き整数を 2^{-n} 倍する。従って、シフトにより空いたビットには、符号ビットが入る。

3.2.2 使用例

```
SRA   GR0,2           ;GR0の内容を2ビット右へシフト
SRA   GR0,0,GR1       ;GR0の内容をGR1の値、右へシフト
```

4 論理シフト命令

論理シフト命令は、左シフト (SLL) と右シフト (SRL) の2つがある。いずれも、符号無し整数を取り扱い、前者は 2^n 倍、後者は 2^{-n} 倍する。論理シフト命令は、符号は考慮していない。

4.1 論理左シフト (SLL)

4.1.1 内容

命令語	SLL
語源	Shift Left Logical (shift:移す left:左 logical:論理上の)
役割	レジスタの内容を n ビット左に移動させる。空いたビットには、0が入る。これは、符号無し整数を 2^n 倍しているのと同じ。
書式	教科書 (p.65) の通り。第一オペランドは汎用レジスター。第二オペランドはアドレス。
機能	教科書 (p.65) の通り
フラグレジスタ	教科書 (p.65) の通り。

この命令は、符号無し整数を 2^n 倍する。従って、シフトにより空いたビットには、0が入る。

4.1.2 使用例

```
SLL  GR0,2          ;GR0の内容を2ビット左へシフト
SLL  GR0,0,GR1      ;GR0の内容をGR1の値、左へシフト
```

4.2 論理右シフト (SRL)

4.2.1 内容

命令語	SRL
語源	Shift Right Logical (shift:移す right:右 logical:論理上の)
役割	レジスタの内容を n ビット右に移動させる。空いたビットには、0が入る。これは、符号付き整数を 2^{-n} 倍しているのと同じ。
書式	教科書 (p.67) の通り。第一オペランドは汎用レジスタ。第二オペランドはアドレス。
機能	教科書 (p.67) の通り
フラグレジスタ	教科書 (p.67-68) の通り。

この命令は、符号無し整数を 2^{-n} 倍する。従って、シフトにより空いたビットには、符号ビットが入る。

4.2.2 使用例

```
SRL  GR0,2          ;GR0の内容を2ビット右へシフト
SRL  GR0,0,GR1      ;GR0の内容をGR1の値、右へシフト
```

5 課題

課題を課すので、レポートとして提出すること。課題内容は、以下の通り。

- CASL II の整数演算で、 $643/32+5$ を計算するプログラムを作成せよ。
- 計算結果である整数の値は、いくらか？

期限 11月12日(金)PM5:00まで

用紙 A4

提出場所 山本研究室の入口のポスト

表紙 表紙を1枚つけて、以下の項目を分かりやすく記述すること。

授業科目名「電子計算機」

課題名「シフト命令」

3E 学籍番号 氏名

提出日