

比較命令とジャンプ命令

山本昌志*

2004年10月15日

1 前回の復習と本日の学習

1.1 復習

1.1.1 これまでの学習内容

前回の授業では、COMET IIのシミュレーター WCASL IIを実際に触れて、その使い方を学習した。今後の授業に生かして欲しい。

今まで、学習したCOMET IIの命令は、次の通りである。これを思い出して、本日の学習内容と絡めて、理解を深める必要がある。

- アセンブラ命令

開始	START	プログラムの先頭を示し、入口名と実行開始番地を定義
終了	END	プログラムの終わりを明示
領域確保	DS	プログラムで使うメインメモリーを予約
定義	DC	メインメモリーの初期値設定

- 機械語命令

- データ転送命令

データ転送	LD	メインメモリーやレジスターの内容を汎用レジスタに転送
データ転送	ST	汎用レジスタのデータをメインメモリーへ転送
アドレス転送	LAD	実効アドレスを汎用レジスターへ転送

- 算術、論理演算

*国立秋田工業高等専門学校 電気工学科

算術加算	ADDA	1 語のデータを符号付き整数と見なし、加算を行う。
論理加算	ADDL	1 語のデータを符号無し整数と見なし、加算を行う。
算術減算	SUBA	1 語のデータを符号付き整数と見なし、減算を行う。
算術減算	SUBL	1 語のデータを符号無し整数と見なし、減算を行う。
論理積	AND	1 語のデータのビット毎の論理積の演算を行う。
論理和	OR	1 語のデータのビット毎の論理和の演算を行う。
排他的論理和	XOR	1 語のデータのビット毎の排他的論理和の演算を行う。

1.1.2 フラグレジスタ

本日の学習では、フラグレジスタが重要な役割を果たす。それは、次のようなものであったことを思い出して欲しい。

- レジスタとは CPU にある記憶領域である。
- フラグレジスタとは、データや演算結果の状態を示すレジスタである。
- 演算結果により、0 か 1 が設定される。フラグ (flag) とは、旗のことで、サッカーの試合で、反則があると旗を上げるのと同じである。
- COMET II には、3 つのフラグレジスタが用意されている。それぞれは、1 ビットで 0 か 1 の値である。

サインフラグ	(Sign Flag)	SF	第 15 ビットが 1 の時、SF=1 となる。
ゼロフラグ	(Zero Flag)	ZF	全てのビットが 0 の時、ZF=1 となる。
オーバーフロー	(Overflow Flag)	OF	演算結果などが 16 ビットを越えた場合、OF=1 となる。
ローフラグ			

1.1.3 符号付き整数と符号無し整数

CASL II で使われる整数は、符号付きと符号無しがある。符号付きと符号無し整数の違いは、次の通りである。なぜこのようになるか、忘れた者は以前のノートを見よ。

表 1: 符号付きと負号無し整数

ビットパターン	符号無し整数	符号有整数
000000001010011	$(83)_{10}$	$(83)_{10}$
100000001010011	$(32851)_{10}$	$(-32685)_{10}$

1.2 本日の学習内容

本日は、比較とジャンプ命令を学習する。これは、セットで使われることが多く、数の比較を行い、その結果を受けて、処理の実行を変える。FORTRAN や C 言語では、

```
IF(A.GT.B)GO TO 200          if(a>b)goto next_step;
```

と書かれる構文とにている。A.GT.B が比較命令で、IF と GO TO がジャンプ命令に相当する。CASL II の比較とジャンプの命令は、次の通りである。

- 整数の大小の比較命令を学習する。比較の結果は、フラグレジスタ (FR) を設定することで示される。

算術比較	CPA	データは符号付き整数と見なし、比較を行う。
論理比較	CPL	データは符号無し整数と見なし、比較を行う。

- フラグレジスタ (FR) の値に基づいて、処理を分岐させる。

正分岐	JPL	SF と ZF がともに 0 の時、指定の実効アドレスに分岐
負分岐	JMI	SF が 1 の時、指定の実効アドレスに分岐
非零分岐	JNZ	ZF が 0 の時、指定の実効アドレスに分岐
零分岐	JZE	ZF が 1 の時、指定の実効アドレスに分岐
オーバーフロー分岐	JOV	OF が 1 の時、指定の実効アドレスに分岐
無条件分岐	JUMP	無条件に、指定の実効アドレスに分岐

2 比較命令

2.1 算術比較 (CPA)

2.1.1 内容

命令語	CPA
語源	ComPare Arithmetic (compare:比較する arithmetic:算術)
役割	符号付き整数の比較を行う命令
書式	教科書 (p.56) の通り
機能	教科書 (p.56) の通り
フラグレジスタ	教科書 (p.56) の通り。

この命令は、符号付き整数の差の演算結果の状態がフラグレジスターに設定すると考えれば良い。たとえば、CPA GR1,GR2 の場合、GR1-GR2 の演算結果の正負、あるいはゼロか否か、オーバーフローが有るか無いかフラグレジスタに設定される。ただし、オペランドである GR1 や GR2 の値は変わらない。

2.1.2 使用例

```
CPA   GRO,GR1      ;GRO-GRO の状態をフラグレジスターに設定
```

CPA GR0,A ;GR0-(アドレス A の内容) の状態をフラグレジスターに設定
 CPA GR0,A,GR1 ;GR0-(アドレス [A+GR1] の内容) の状態をフラグレジスターに設定
 CPA GR0,=5 ;GR0-5 の状態をフラグレジスターに設定

教科書の例題を実行したときのメモリーとレジスターの内容を表 2 示す。1, 5, 6, 7 行はアセンブラ命令なので実行されない。そのため、メモリーやレジスタの値は空白としている。

表 2: 教科書 List4-11(p.56) の実行例。

行	プログラム			GR1	OF	SF	ZF	AA	BB
1	PGM	START							
2		LD	GR1,AA	2	0	0	0	2	-1
3		CPA	GR1,BB	2	0	0	0	2	-1
4		RET		2	0	0	0	2	-1
5	AA	DC	2						
6	BB	DC	-1						
7		END							

教科書の List4-11 はつまらない例題で、CPA コマンドも動作も分からないし、説明もおかしい。次のようなサンプルの方が良い。

表 3: CPA の実行例。

行	プログラム			GR1	OF	SF	ZF	AA	BB
1	PGM	START							
2		LD	GR1,AA	2	0	1	0	-2	-3
3		CPA	GR1,BB	2	0	0	0	-2	-3
4		RET		2	0	0	0	-2	-3
5	AA	DC	-2						
6	BB	DC	-3						
7		END							

2.2 論理比較 (CPL)

2.2.1 内容

命令語 CPL
 語源 ComPare Logical (compare:比較する logical:論理上の)
 役割 符号無し整数の比較を行う命令
 書式 教科書 (p.57) の通り
 機能 教科書 (p.57) の通り
 フラグレジスタ 教科書 (p.57) の通り。

この命令は、符号無し整数の差の演算結果の状態がフラグレジスターに設定すると考えれば良い。たとえば、CPA GR1,GR2 の場合、GR1-GR2 の演算結果の正負、あるいはゼロか否か、オーバーフローが有るか無

いかがフラグレジスタに設定される。ただし、オペランドである GR1 や GR2 の値は変わらない。

2.2.2 使用例

```
CPL  GRO,GR1      ;GRO-GRO の状態をフラグレジスタに設定
CPL  GRO,A        ;GRO-(アドレス A の内容) の状態をフラグレジスタに設定
CPL  GRO,A,GR1    ;GRO-(アドレス [A+GR1] の内容) の状態をフラグレジスタに設定
CPL  GRO,=5       ;GRO-5 の状態をフラグレジスタに設定
```

教科書の例題を実行したときのメモリーとレジスタの内容を表 4 示す。1, 5, 6, 7 行はアセンブラ命令なので実行されない。そのため、メモリーやレジスタの値は空白としている。

このサンプルプログラムで注意することは、符号付き整数でアドレス AA の値を決めているが、比較を行うときは符号無し整数としている。それらは、

アドレス	ビットパターン	符号無整数	符号有整数
AA	0000000000000010	(2) ₁₀	(2) ₁₀
BB	1111111111111111	(65535) ₁₀	(-1) ₁₀

となっている。

表 4: 教科書 List4-12(p.58) の実行例。

行	プログラム			GR1	OF	SF	ZF	AA	BB
1	PGM	START							
2		LD	GR1,AA	2	0	0	0	2	-1
3		CPL	GR1,BB	2	0	1	0	2	-1
4		RET		2	0	1	0	2	-1
5	AA	DC	2						
6	BB	DC	-1						
7		END							

3 ジャンプ命令

3.1 正分岐 (JPL)

3.1.1 内容

命令語	JPL
語源	Jump PL us (jump:ジャンプ plus:プラス (正))
役割	フラグレジスタ (FR) の SF(Sign Flag) が 0、かつ、ZF(Zero Flag) が 0 のとき、指定の実効アドレスに制御が移る。
書式	教科書 (p.71) の通り
機能	教科書 (p.71) の通り
フラグレジスタ	変化せず。

この命令は、これ以前の演算の結果の状態が、0 を含まない正の場合、指定の実効アドレスに分岐させる。通常は、比較命令とペアで使われることが多いが、そうでない場合もある。この命令を使う場合は、フラグレジスタの状態をよく考える必要がある。

3.1.2 使用例

```
JPL A ;SF=0 かつ ZF=0 の場合、アドレス a にジャンプ
JPL A,GR1 ;SF=0 かつ ZF=0 の場合、アドレス [A+GR1] にジャンプ
```

教科書の例題 (p.71 List4-22) のプログラムを図 1 に示す。

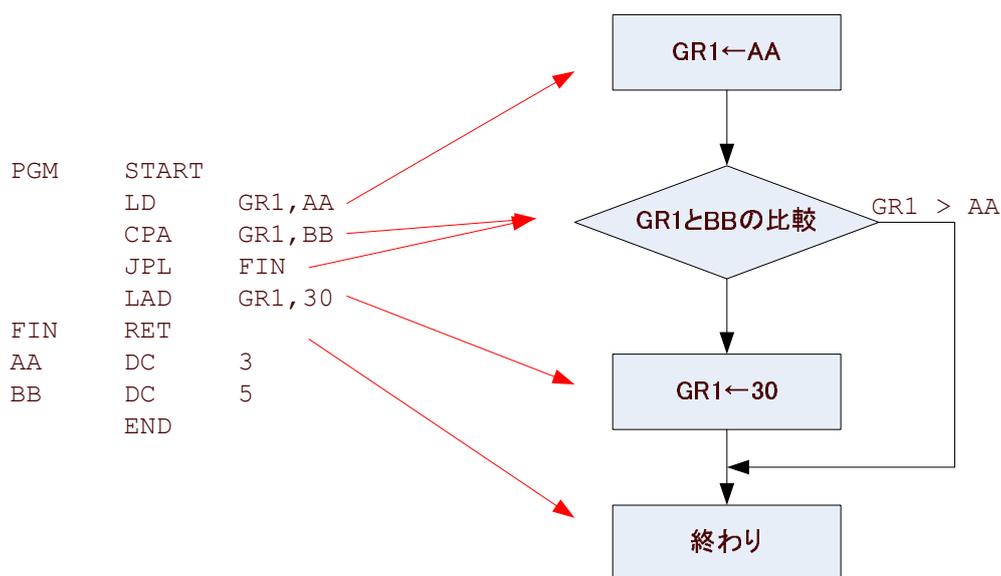


図 1: List4-22 のプログラムとフローチャート

3.2 負分岐 (JMI)

命令語	JMI
語源	JumpMInus (jump:ジャンプ minus:マイナス(負))
役割	フラグレジスタ (FR) の SF(Sign Flag) が 1 のとき、指定の実効アドレスに制御が移る。
書式	教科書 (p.72) の通り
機能	教科書 (p.72) の通り
フラグレジスタ	変化せず。

3.3 非零分岐 (JNZ)

命令語	JNZ
語源	Jump on Non Zero (jump:ジャンプ non:非 zero:ゼロ)
役割	フラグレジスタ (FR) の ZF(Zero Flag) が 0 のとき、指定の実効アドレスに制御が移る。
書式	教科書 (p.73) の通り
機能	教科書 (p.73) の通り
フラグレジスタ	変化せず。

3.4 零分岐 (JZE)

命令語	JZE
語源	Jump on ZERo (jump:ジャンプ zero:ゼロ)
役割	フラグレジスタ (FR) の ZF(Zero Flag) が 1 のとき、指定の実効アドレスに制御が移る。
書式	教科書 (p.73) の通り
機能	教科書 (p.74) の通り
フラグレジスタ	変化せず。

3.5 オーバーフロー分岐 (JOV)

命令語	JOV
語源	Jump on OVerflow (jump:ジャンプ overflow:あふれる)
役割	フラグレジスタ (FR) の OF(Overflow Flag) が 1 のとき、指定の実効アドレスに制御が移る。
書式	教科書 (p.74) の通り
機能	教科書 (p.74) の通り
フラグレジスタ	変化せず。

3.6 無条件分岐 (JUMP)

命令語	JUMP
語源	unconditional Jump (jump:ジャンプ)
役割	フラグレジスタ (FR) に関係なく、強制的に指定の実効アドレスに制御を移す。
書式	教科書 (p.75) の通り
機能	教科書 (p.75) の通り
フラグレジスタ	変化せず。

4 課題

課題を課すので、レポートとして提出すること。課題は、教科書 p.75 の List4-23 のプログラムについてである。

- フローチャートを作成せよ。そして、ソースプログラムの各行とフローチャートの対応を示せ。
- 使っている汎用レジスタの値を各行が実行された後、どのようになるか示せ。
- 各行の実行後のフラグレジスタの値を示せ。
- 各行の実行後のアドレス AA と BB の値を示せ。

提出方法は、次の通りとする。

期限 10月22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「比較とジャンプ命令」

3E 学籍番号 氏名

提出日