

# プロセッサと機械語

## 2年次前期 (第7回)

中島 克人

情報メディア学科

nakajima@im.dendai.ac.jp

1

# 本日の内容

- 条件分岐
  - データにより, 処理を切り替える
  - ループ
    - 同じコードを何度も書かずに済ませる
    - = プログラムがコンパクトになり, 読みやすくなる

2

# プロセッサと機械語

## 条件分岐命令

にも頻繁に使用される

3

# COMET II マシンの機械命令

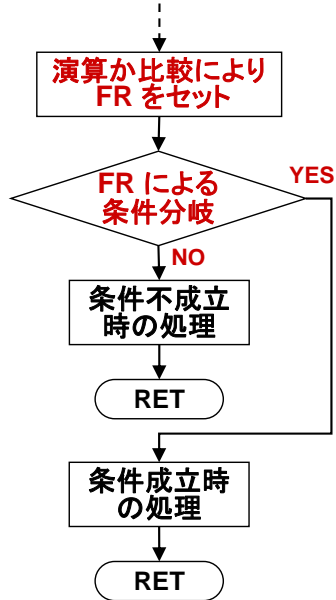
(教科書 p.25)

- 無条件分岐 (jump) 命令
  - [書き方] JP adr[,xr]
  - [機能] e → PR
- 正分岐 (jump on plus) 命令
  - [書き方] JPL adr[,xr]
  - [機能] if SF=0 and ZF=0 then e → PR
- 負分岐 (jump on minus) 命令
  - [書き方] JMI adr[,xr]
  - [機能] if SF=1 then e → PR
- 零分岐 (jump on zero) 命令
  - [書き方] JZE adr[,xr]
  - [機能] if ZF=1 then e → PR
- 非零分岐 (jump on non zero) 命令
  - [書き方] JNZ adr[,xr]
  - [機能] if ZF=0 then e → PR
- オーバフロー分岐 (jump on overflow) 命令
  - [書き方] JOV adr[,xr]
  - [機能] if OF=1 then e → PR

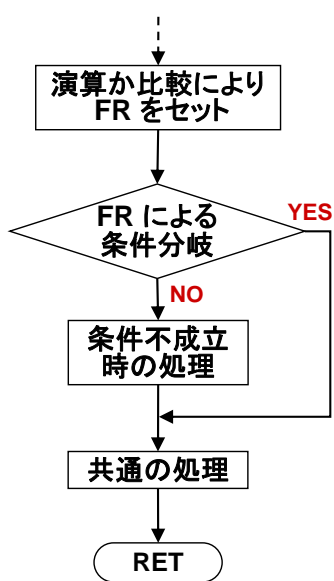
4

# COMET II マシンの機械命令

- 条件分岐命令の使い方



- 条件分岐命令の上手な使い方



# アセンブリ言語CASL II

(教科書 p.34)

- 演習BIGR ... CPL と JPL を使って, 大きい方を求めるプログラムを作成し, WCASL II で実行せよ
- なお, ラベル DAT1 には自分の学籍番号下3桁を, DAT2 には隣席学生の学籍番号の下3桁を DC せよ

```

BIGR  START
      LD  GR0,DAT1
      CPL [ ]; <GR0> と <DAT2> の大小比較
      JPL [ ]; 結果が+(プラス)ならジャンプ
L1    LD  GR0,DAT2
      ST  GR0,RSLT
      RET
DAT1  DC [ ]; 自分の学籍番号の下3桁
DAT2  DC [ ]; 隣席学生の学籍番号の下3桁
RSLT  DS 1
      END
    
```

# アセンブリ言語CASL II

- 演習BIGR ... 素直だが上手とは言えないプログラム例

```

BIGR  START
      LD  GR0,DAT1
      [ ]; <GR0> と <GR1> の大小比較
      [ ]; 結果が+(プラス)ならジャンプ
L1    ST  GR0,RSLT
      RET
DAT1  DC 001
DAT2  DC 002
RSLT  DS 1
      END
    
```

==<DAT1>    ==<DAT2>

# アセンブリ言語CASL II

- 演習 JZE ... 下記のプログラムの DAT に自分の学籍番号を入れて完成させ, RSLT に格納される結果を確かめよ

```

JZE  START
      LD  GR0,DAT ; 学籍番号のロード
      AND GR0,MSK1 ; 最下位ビットの取り出し
      JZE EVEN
      LAD GR0,#FFFF ; #FFFF → GR0
EVEN ST  GR0,RSLT ; <GR0> → RSLT
      RET
DAT  DC [ ]; 自分の学籍番号を10進数で入力
MSK1 DC 1
RSLT DS 1
      END
    
```

- このプログラム JZE の機能(何をしているのか)を言葉で説明せよ

# アセンブリ言語CASL II

- 演習 LDST3 ... DAT~DAT+2番地の内容をADR~ADR+2番地に格納するプログラム作成し, WCASL IIで実行せよ


```

LDST3 START
; DAT 番地の内容を GR0 に
; <GR0> → ADR
; 1 → GR1
; <e>(=<DAT+1>) → GR0
; <GR0> → ADR+1
; 2 → GR1
; <e>(=<DAT+2>) → GR0
; <GR0> → ADR+2
RET
DAT DC #0001,#0002,#0003 ; 定数#0001~#0003を連続配置
ADR DS 3 ; 3語分の領域をここに確保
END
    
```

問題: DAT~DAT+9番地の内容をADR~ADR+9番地に格納するには?


# アセンブリ言語CASL II

- DAT~DAT+9番地の内容をADR~ADR+9番地に格納する

<pre> LDST10 START &lt;DAT&gt; → GR0 &lt;GR0&gt; → ADR 1 → GR1 &lt;DAT+1&gt; → GR0 &lt;GR0&gt; → ADR+1 2 → GR1 &lt;DAT+2&gt; → GR0 &lt;GR0&gt; → ADR+2 ... 9 → GR1 &lt;DAT+9&gt; → GR0 &lt;GR0&gt; → ADR+9 RET                 </pre>		<pre> LDST10 START 0 → GR1 1 → GR2 &lt;DAT+&lt;GR1&gt;&gt; → GR0 &lt;GR0&gt; → ADR + &lt;GR1&gt; &lt;GR1&gt; + &lt;GR2&gt; → GR1 &lt;DAT+&lt;GR1&gt;&gt; → GR0 &lt;GR0&gt; → ADR + &lt;GR1&gt; &lt;GR1&gt; + &lt;GR2&gt; → GR1 ... &lt;DAT+&lt;GR1&gt;&gt; → GR0 &lt;GR0&gt; → ADR + &lt;GR1&gt; (&lt;GR1&gt; + &lt;GR2&gt; → GR1) RET                 </pre>
---	---	---

# アセンブリ言語CASL II

- DAT~DAT+9番地の内容をADR~ADR+9番地に格納する

<pre> LDST10 START 0 → GR1 1 → GR2 &lt;DAT+&lt;GR1&gt;&gt; → GR0 &lt;GR0&gt; → ADR + &lt;GR1&gt; &lt;GR1&gt; + &lt;GR2&gt; → GR1 &lt;DAT+&lt;GR1&gt;&gt; → GR0 &lt;GR0&gt; → ADR + &lt;GR1&gt; &lt;GR1&gt; + &lt;GR2&gt; → GR1 ... &lt;DAT+&lt;GR1&gt;&gt; → GR0 &lt;GR0&gt; → ADR + &lt;GR1&gt; (&lt;GR1&gt; + &lt;GR2&gt; → GR1) RET                 </pre>		<pre> LDST10 START 0 → GR1 1 → GR2 LP(ラベル) &lt;DAT+&lt;GR1&gt;&gt; → GR0 &lt;GR0&gt; → ADR + &lt;GR1&gt; &lt;GR1&gt; + &lt;GR2&gt; → GR1 if &lt;GR1&gt; &lt;input type="checkbox" value="10"/&gt; 10 then ラベルLPまでジャンプ RET                 </pre>
---	---	--

問題: 上記の  に入る大小記号はどれ? (a) > (b) ≥ (c) < (d) ≤

# アセンブリ言語CASL II

- DAT~DAT+9番地の内容をADR~ADR+9番地に格納する  
実際のプログラム

<pre> LDST10 START 0 → GR1 1 → GR2 LP &lt;DAT+&lt;GR1&gt;&gt; → GR0 &lt;GR0&gt; → ADR + &lt;GR1&gt; &lt;GR1&gt; + &lt;GR2&gt; → GR1 if &lt;GR1&gt; &lt;input type="checkbox" value="10"/&gt; 10 then ラベルLPまでジャンプ RET                 </pre>	<pre> LDST10 START LAD GR1,0 LAD GR2,1 LP LD GR0,DAT,GR1 ST GR0,ADR,GR1 ADDA GR1,GR2 CPA GR1,TEN JMI LP RET TEN DC 10 DAT DC X,X,X, ..., X ; 定数を10連続配置 ADR DS 10 ; 10語分の領域確保 END                 </pre>
---	---

# アセンブリ言語CASL II

- 演習 JUF ... 下記のプログラムのDATに自分の学籍番号下3桁を入れて完成させ、RSLTに格納される結果を確かめよ

```
JUF  START
      LD  GR0,DAT      ; 学籍番号のロード
      CPL GR0,CMP1     ; 50 との大小比較
      JMI UDR50
      SUBA GR0,CMP1    ; <GR0>-50 → GR0
      CPL GR0,CMP1     ; 50 との大小再比較
      JMI UDR50
      SUBA GR0,CMP1    ; <GR0>-50 → GR0
UDR50 ST  GR0,RSLT     ; <GR0> → RSLT
      RET
DAT   DC  [ ]         ; 自分の学籍番号下3桁を10進数で入力
CMP1  DC  50
RSLT  DS  1
      END
```

13

## 今週は宿題はありません。

14