

プロセッサと機械語

2年次前期（第4回）

中島 克人

情報メディア学科

nakajima@im.dendai.ac.jp

1

メモリ参照と演算

2

アセンブリ言語CASL II

● プログラム例 (SMPL)

； コメント行は「;」よりも左から行末まで

SMPL **START** ; ラベルは左端カラムから, **START**は必須
; 命令(ラベル以外)は 2カラム目以降に

ラベルは左端から書く

LD GR0,DAT ; DAT 番地の内容を GR0 に
; 即ち, <DAT> → GR0

LAD GR1,1 ; 1 → GR1

プログラムの最初と最後を示す

ADDA GR0,GR1 ; <GR0> + <GR1> → GR0

ST GR0,ADR1 ; <GR0> → ADR1

RET ; このプログラムを起動したOSに制御を戻す(必須)

DAT DC #0002 ; 定数#0002 をここに配置

ADR1 DS 1 ; 1語分の領域 をここに確保

END ; ENDは必須

3

COMET II マシンの機械命令

● ロード(load) 命令

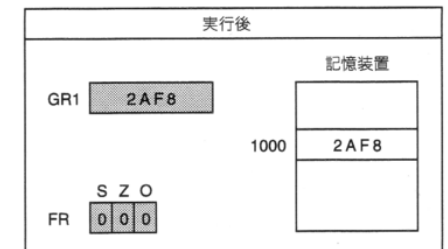
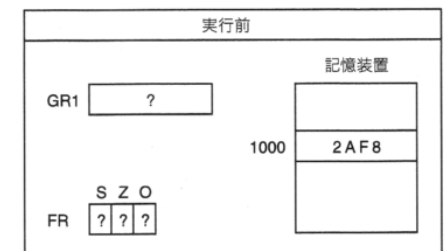
[書き方] LD gr,adr[,xr]

[機能] <e> → GRn; 0→OF;

SZ設定

if <GRn>=0 then 1→ZF
else 0→ZF
<GRn[15]>→SF;

[実行例] LD GR1,1000



gr : GR0~GR7
xr : GR1~GR7
[]: 省略可
e : 実効番地 (EA)
<α>: α の内容
→: 値の代入 (コピー)

4

COMET II マシンの機械命令

● ロード(load) 命令

[書き方] LD gr1,gr2

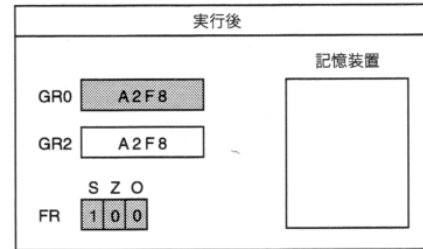
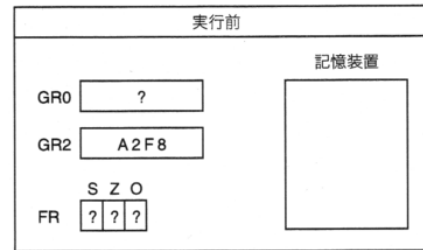
[機能] <GRn2> → GRn1;

0→OF;

SZ設定

{ if <GRn>=0 then 1→ZF
else 0→ZF
<GRn[15]>→SF;

[実行例] LD GR0,GR2



5

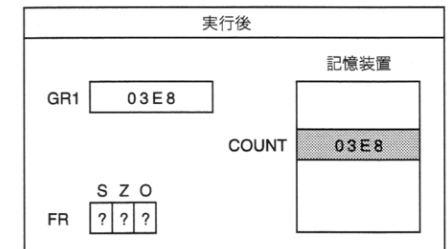
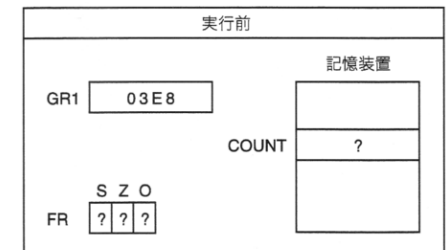
COMET II マシンの機械命令

● ストア(store) 命令

[書き方] ST gr,adr[,xr]

[機能] <GRn>→e

[実行例] ST GR1,COUNT



6

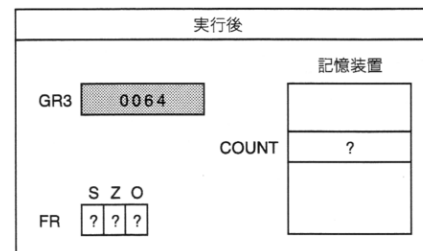
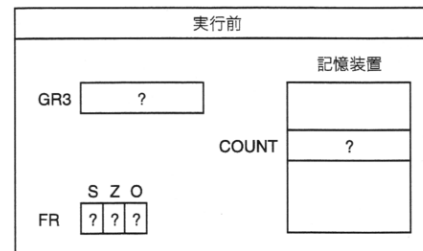
COMET II マシンの機械命令

● ロードアドレス(load address) 命令

[書き方] LAD gr,adr[,xr]

[機能] e → GRn

[実行例] LAD GR3,COUNT



7

WCASL II で予想を確かめて見よう

● 右図のような時、以下の命令を実行後の GR0 の内容は？

問題1: LAD GR0,#0025

予想解答: 解答:

問題2: LAD GR0,DAT

予想解答: 解答:

問題3: LAD GR0,DAT,GR1

予想解答: 解答:

問題4: LD GR0,DAT

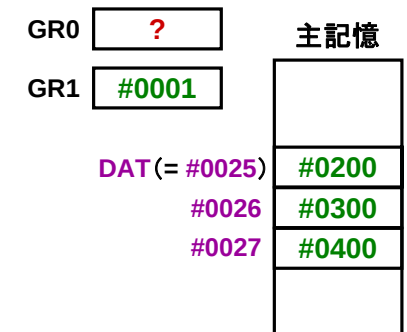
予想解答: 解答:

問題5: LD GR0,DAT,GR1

予想解答: 解答:

問題6: LD GR0,#0024,GR1

予想解答: 解答:



8

アセンブリ言語CASL II

• テスト用プログラム例 (TEST)

```
TEST  START      ; STARTは必須
      ;
      LAD GR1,1    ; 1 → GR1
      ;
      LD  GR0,#0025 ; #0025番地の内容を GR0 に
      ; 即ち, <#0025> → GR0
;;      ST  GR0,RSLT
      RET      ; このプログラムを起動したOSに制御を戻す(必須)
DAT    DC  #0200,#0300,#0400
      ; 定数#0002,#0300,#0400 をDAT番地から3語分に配置
;; RSLT DS  1      ; 結果格納用に1語確保
      END      ; ENDは必須
```

9

アセンブリ言語CASL II

• テスト用プログラム例 (TEST)

```
TEST  START      ; STARTは必須
      ;
      LAD GR1,1    ; 1 → GR1
      ;
      LD  GR0,#0025 ; テスト対象の命令
      ;
      ST  GR0,RSLT
      RET      ; このプログラムを起動したOSに制御を戻す(必須)
DAT    DC  #0200,#0300,#0400
      ; 定数#0002,#0300,#0400 をDAT番地から3語分に配置
RSLT  DS  1      ; 結果格納用に1語確保
      END      ; ENDは必須
```

10

プロセッサと機械語

算術演算と オーバーフロー

11

COMET II マシンの機械命令

(教科書 p.17)

• 算術加算 (add arithmetic) 命令

[書き方] ADDA gr,adr[,xr]

[機能] <GRn> + <e> → T;

<T> → GRn;

if (<T> > 32767 = 2¹⁵ - 1
or <T> < -32768)

then 1 → OF else 0 → OF;

SZ設定

if <GRn> = 0 then 1 → ZF

else 0 → ZF

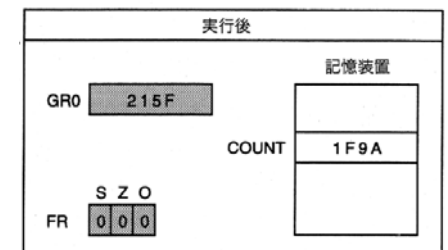
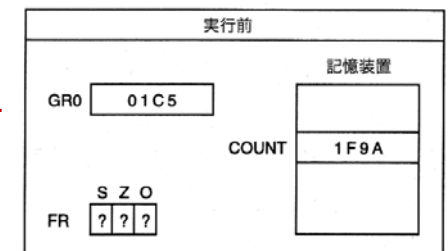
<GRn[15]> → SF;

(1語命令)

[書き方] ADDA gr1,gr2

[機能] 上記と同様

[実行例] ADDA GR0,COUNT

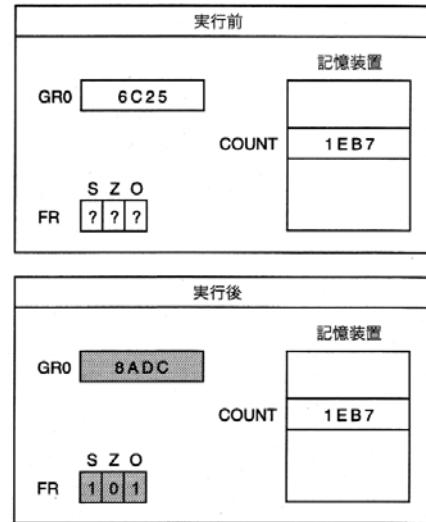


12

COMET II マシンの機械命令

- 算術加算 (add arithmetic) 命令 (教科書 p.18)

[実行例] ADDA GR0,COUNT



13

COMET II マシンの機械命令

- 論理加算 (add logical) 命令 (教科書 p.18)

[書き方] ADDL gr,adr[,xr]

[機能] $\langle \text{GRn} \rangle + \langle e \rangle \rightarrow \text{T};$

$\langle \text{T} \rangle \rightarrow \text{GRn};$

if $\langle \text{T} \rangle > 65535 = 2^{16} - 1$

then $1 \rightarrow \text{OF}$ else $0 \rightarrow \text{OF};$

SZ設定

if $\langle \text{GRn} \rangle = 0$ then $1 \rightarrow \text{ZF}$

else $0 \rightarrow \text{ZF}$

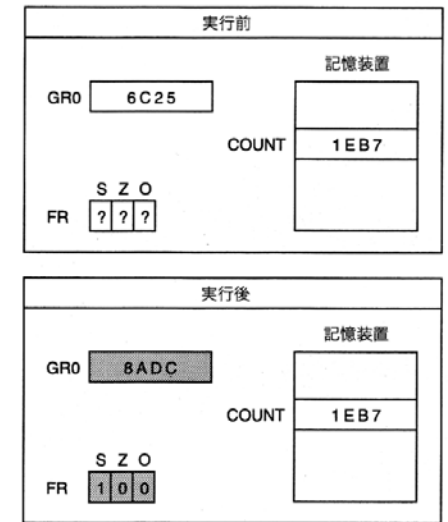
$\langle \text{GRn}[15] \rangle \rightarrow \text{SF};$

(1語命令)

[書き方] ADDL gr1,gr2

[機能] 上記と同様

[実行例] ADDL GR0,COUNT



14

アセンブリ言語CASL II

- プログラム例 (SMPL2)

```
SMPL2 START
LAD GR0,#0001      ; #0001 を GR0 に
LAD GR1,1          ; 1 → GR1
LD GR2,DAT         ; DAT 番地の内容を GR2 に
                  ; 即ち, <DAT> → GR2
LD GR3,DAT,GR1     ; <e> (= <DAT> + <GR1>) → GR3
ADDA GR2,GR3       ; <GR2> + <GR3> → GR2
ST GR2,ADR1        ; <GR2> → ADR1
RET
DAT DC #0002       ; 定数#0002 をここに配置
DC #0003           ; 定数#0003 をここに配置
ADR1 DS 1          ; 1語分の領域 をここに確保
END
```

CASLシミュレータで実行する前に、結果を予想して見よう！

15

アセンブリ言語CASL II

- プログラム例 (SMPL3)

GR0 に#01C5 をロードした後、COUNT番地の #1F9A と算術加算し、結果の値とFRの値を確認せよ

```
SMPL3 START
LAD GR0,#01C5      ; #01C5 を GR0 に
                  ; <GR0> + <COUNT> → GR0
COUNT DC #1F9A    ; 定数#1F9A をここに配置
END
```

同様に、GR0 に#6C25 をロードした後、COUNT番地の #1EB7 と算術加算し、結果の値とFRの値を確認せよ

同様に、GR0 に#6C25 をロードした後、COUNT番地の #1EB7 と論理加算し、結果の値とFRの値を確認せよ

16

COMET II マシンの機械命令

- 算術減算 (subtract arithmetic) 命令 (教科書 p.19)

[書き方] SUBA gr,adr[,xr]

[機能] $\langle \text{GRn} \rangle - \langle e \rangle \rightarrow \text{T};$

$\langle \text{T} \rangle \rightarrow \text{GRn};$

if $\langle \text{T} \rangle > 32767 = 2^{15} - 1$

or $\langle \text{T} \rangle < -32768$

then $1 \rightarrow \text{OF}$ else $0 \rightarrow \text{OF};$

SZ設定

if $\langle \text{GRn} \rangle = 0$ then $1 \rightarrow \text{ZF}$

else $0 \rightarrow \text{ZF}$

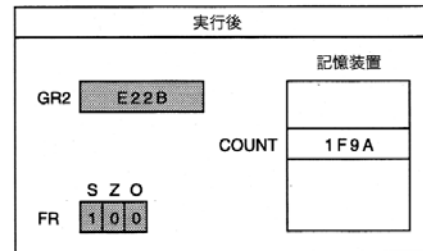
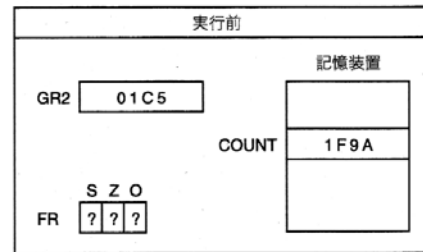
$\langle \text{GRn}[15] \rangle \rightarrow \text{SF};$

(1語命令)

[書き方] SUBA gr1,gr2

[機能] 上記と同様

[実行例] SUBA GR2,COUNT



17

COMET II マシンの機械命令

- 論理減算 (subtract logical) 命令 (教科書 p.19)

[書き方] SUBL gr,adr[,xr]

[機能] $\langle \text{GRn} \rangle - \langle e \rangle \rightarrow \text{T};$

$\langle \text{T} \rangle \rightarrow \text{GRn};$

if $\langle \text{T} \rangle < 0$

then $1 \rightarrow \text{OF}$ else $0 \rightarrow \text{OF};$

SZ設定

if $\langle \text{GRn} \rangle = 0$ then $1 \rightarrow \text{ZF}$

else $0 \rightarrow \text{ZF}$

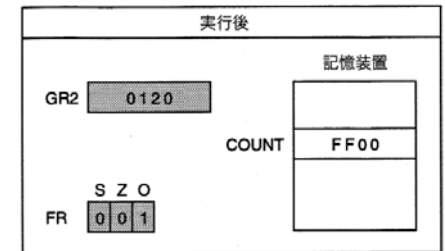
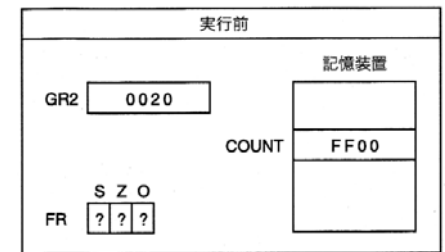
$\langle \text{GRn}[15] \rangle \rightarrow \text{SF};$

(1語命令)

[書き方] SUBL gr1,gr2

[機能] 上記と同様

[実行例] SUBL GR2,COUNT



18

演習

- 下記を行う SMPL4 を作成し、実行して見よ

DAT番地のデータ #0001 と (DAT+1)番地のデータ #8000
を算術加算し、(DAT+2)番地に格納せよ

```

SMPL4 START
    LAD GR1,1          ; 1 → GR1
                        ; <DAT> → GR2
                        ; <e> (= <DAT> + <GR1>) → GR3
                        ; <GR2> + <GR3> → GR2
                        ; 2 → GR1
                        ; <GR2> → DAT + <GR1>

    ; 定数#0001 をここに配置
    ; 定数#8000 をここに配置
    ; 1語分の領域 をここに確保

END
    
```

19

演習

- SMPL4 の実行の結果、SF, ZF, OF はどうなったか？

その理由も考察せよ

FR (フラグレジスタ)

ヒント:

- SZ設定する命令は FR を3ビットまとめて上書きする
- 最後の SZ設定 が有効となる

解答:

- 最後に SZ設定 したのは 命令で、結果は #
- その時の結果は、SF = , ZF = , OF =

- 算術加算の代わりに算術減算とすると、SF, ZF, OF はどうなるか？

解答:

-

20

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

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
```

21

宿題: LD,LAD,ADDA,ST命令等を含むプログラムの実行結果を求めよ
(それぞれ結果予想を行った上で, WCASL IIで実行し, 確かめよ)

- 授業HP「第4回宿題」のPDFを各自印刷し, 次回授業の冒頭に副手に提出すること

22