

プロセッサと機械語

2年次前期（第8回）

中島 克人

情報メディア学科

nakajima@im.dendai.ac.jp

1

第1回テストの解答

2

問題1 解答例

- プログラム **OTT** はラベル **DAT** 番地の内容(この例では10進数で123であるが、任意の数)を1倍した値(**DAT** 番地の内容そのまま)、2倍した値、3倍した値をラベル **RSLT** 番地から続く3語の領域に格納するプログラムである。
 部分を埋めてプログラムを完成させよ。

```
OTT  START
      LD GR0,DAT
      ST GR0,RSLT
      SLA GR0,1
      LAD GR1,1
      ST GR0,RSLT,GR1
      ADDA GR0,DAT
      LAD GR1,2
      ST GR0,RSLT,GR1
      RET
DAT   DC   123
RSLT  DS   3
      END
```

インテックスレジスタを+1するには次の方法もある。

```
LAD GR1,1,GR1
```

また、1の入ったレジスタを予め用意しておき、このレジスタの内容を加算する方法もある。

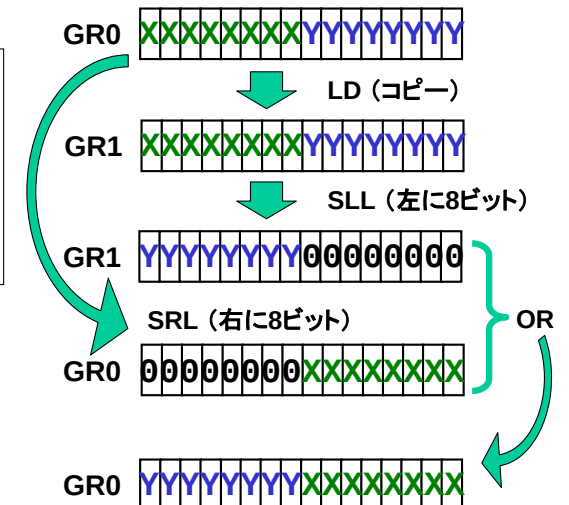
```
LAD GR7,1 ; 1のレジスタ用意
      ⋮
      ADDA GR1,GR7
```

3

問題2 解答例

- プログラム **REVRL** は GR0 の内容の上位8ビットと下位8ビットを入れ替えて GR1 に格納するプログラムである。
 部分を埋めてプログラムを完成させよ。

```
REVRL START
      LD GR1,GR0
      SLL GR1,8
      SRL GR0,8
      OR GR1,GR0
      RET
      END
```



4

問題3 解答

- 以下は同じ数を表す 2進数 (8bit, 負数は2の補数表現), 16進数 (2桁), および, 符号付10進数を各行に示している。
 部分を埋めて表を完成させよ。
 (計算の仕方は自由である。プログラムを書き, CASL-IIシミュレータで求めても良い。)

2進数	16進数 (2桁)	10進数 (符号付任意桁)
00111010 ₂	= 3A<sub>16</sub>	= 58<sub>10</sub>
11101100 ₂	= EC<sub>16</sub>	= -20<sub>10</sub>
00111011<sub>2</sub>	= 3B<sub>16</sub>	= 59<sub>10</sub>
10110100<sub>2</sub>	= B4<sub>16</sub>	= -76<sub>10</sub>
11010011<sub>2</sub>	= D3<sub>16</sub>	= -45<sub>10</sub>
01111110<sub>2</sub>	= 7E<sub>16</sub>	= 126<sub>10</sub>

5

問題4 解答

- 下記の左側の10進数の加減算の結果を求め, 4桁の16進数 (2進数16ビット相当)で答えよ。更に, これを COMET-IIマシンで実行した場合のフラグレジスタ FR=(S,Z,O)の結果も合わせて答えよ。
 (計算の仕方は自由である。プログラムを書き, CASL-IIシミュレータで求めても良い。)

10進数加減算	結果の16進数 (4桁) (S,Z,O)
10 ₁₀ + 20 ₁₀	= #001E (001E<sub>16</sub>) 0,0,0
10 ₁₀ - 20 ₁₀	= #FFF6 (FFF6<sub>16</sub>) 1,0,0
20 ₁₀ - 40 ₁₀	= #FFEC (FFEC<sub>16</sub>) 1,0,0

プログラムを書き, CASL2
シミュレータで求める場合,
右図を実行し, GR0の16進
数の値を読み取れば良い。

```

ADSB START
LD    GR0,A
ADDA  GR0,B
RET
A     DC 10 ; 10-20の場合
B     DC -20 ; 10-20の場合
END
    
```

6

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
    
```

7

アセンブリ言語CASL II

宿題

- 演習 JUF2 ... JUF を比較命令を使用せずに書き直してみよ

```

JUF2 START
 ; 学籍番号のロード
 ; <GR0>-50 → GR0
 ; 正ならもう一度減算,
 ; 零でももう一度減算
UDR50 ADDA  GR0,CMP1 ; <GR0>+50 → GR0
STR    ST    GR0,RSLT ; <GR0> → RSLT
RET
DAT    DC    105 ; 自分の学籍番号下3桁を10進数で入力
CMP1   DC    50
RSLT   DS    
END
    
```

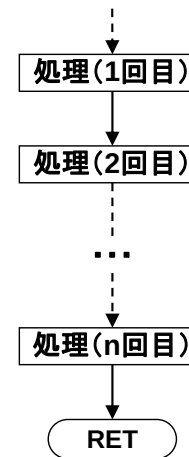
8

ループの構成

9

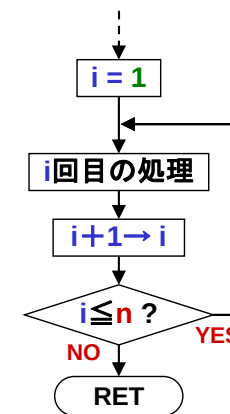
- 比較(または演算)命令と条件分岐命令によるループの実現

同じ処理をn回繰り返す

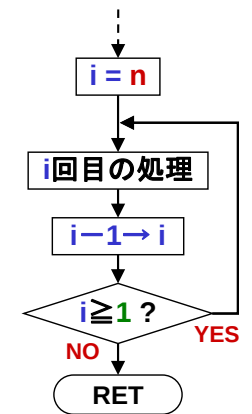


ループカウンタ(i: ループインデックスとも)で何回目かを制御

i = 1 ~ n の場合



i = n ~ 1 の場合



10

アセンブリ言語CASL II

- 演習 STNUM ... 1,2,3 ~ N を ADR ~ ADR+(N-1) 番地に格納するプログラム作成し, WCASL II で実行せよ

- 繰り返し処理をまず考える
- ループカウンタやインデックス・レジスタを導入する
(兼用できないか検討する)
- ループする(同じ所に戻る)か, 終了するための条件分岐法を決定する
- ループ回数に留意しながら, 初期化部分を追加する

格納すべき数 K(1,2,3 ~ N) は初めに 1 にしておく(初期化)
インデックス・レジスタ i は初めに 0 にしておく(初期化)
(ループカウンタは K を兼用する)

```

K → ADR + <i>
K番目の値 → ADR + (K-1)番地
if (K > N) then 終了
K + 1 → K
i + 1 → i
  
```

これを K=1 から K=N
まで繰り返す

11

アセンブリ言語CASL II

- 演習 STNUM ... 1,2,3 ~ N を ADR ~ ADR+(N-1) 番地に格納するプログラム作成し, WCASL II で実行せよ

```

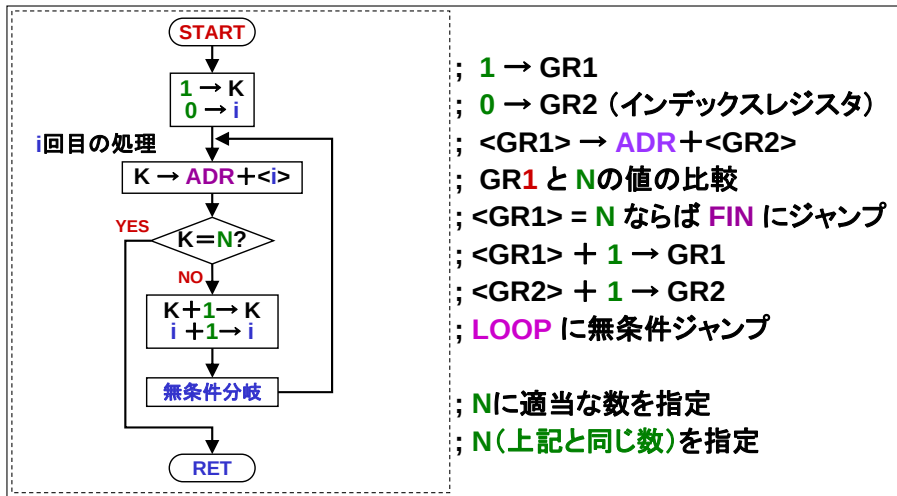
STNUM START
; 1 → GR1
; 0 → GR2 (インデックスレジスタ)
; <GR1> → ADR + <GR2>
; GR1 と N の値の比較
; <GR1> = N ならば FIN にジャンプ
; <GR1> + 1 → GR1
; <GR2> + 1 → GR2
; LOOP に無条件ジャンプ
LOOP
JUMP LOOP
FIN
RET
DAT DC N
ADR DS N
END
  
```

; N に適当な数を指定
; N(上記と同じ数)を指定

12

アセンブリ言語CASL II

- 演習 STNUM ... 1,2,3~N を ADR ~ ADR+(N-1) 番地に格納するプログラム作成し, WCASL II で実行せよ

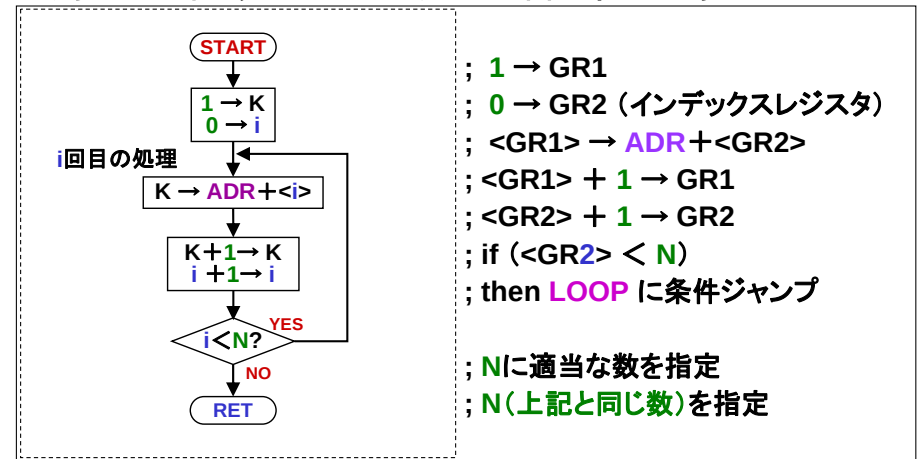


13

アセンブリ言語CASL II

宿題

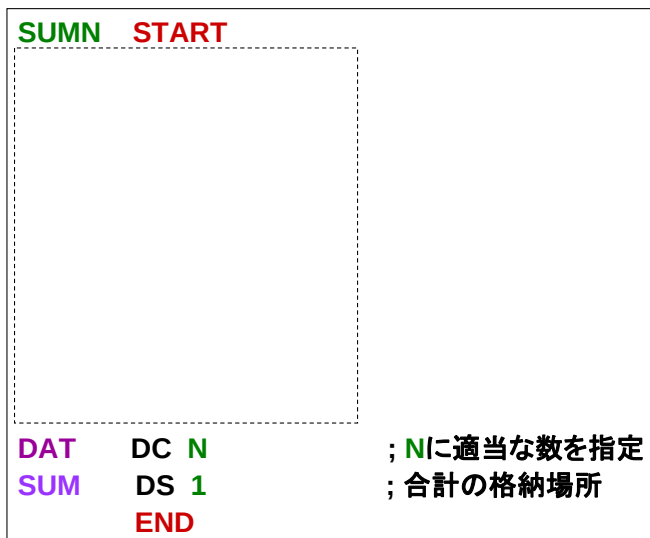
- 演習 STNUM ... ジャンプ命令を1箇所だけに用いて, より少ない行数のプログラムに書き直して見よ.



14

アセンブリ言語CASL II

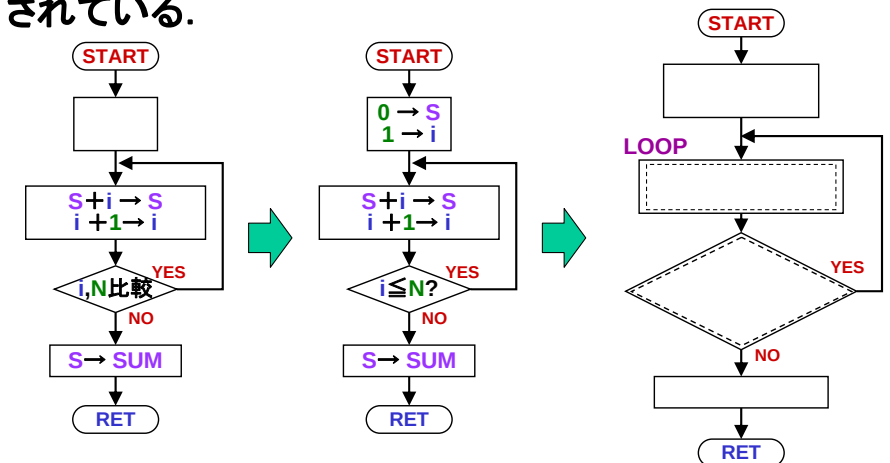
- 演習 SUMN ... 1,2,3~N の合計を SUM 番地に格納するプログラム作成せよ. なお, N は DAT 番地に格納されている.



15

アセンブリ言語CASL II

- 演習 SUMN ... 1,2,3~N の合計を SUM 番地に格納するプログラム作成せよ. なお, N は DAT 番地に格納されている.



16

アセンブリ言語CASL II

- 例題 **MULT0** ... ラベル**DAT**番地の内容(**m**とする)をラベル**N**番地の内容(**n**とする)倍して、ラベル**RSLT**番地に格納する。

```

MULT0 START
    LAD GR0,0      ; 0 → GR0 (結果の格納用レジスタを初期化)
    LD  GR1,N      ; n → GR1 (ループの回数を数えるカウンタ i)
    LAD GR2,1      ; カウンタ i の減算用の 1
LOOP   ADDA GR0,DAT ; 「処理=GR0にmを足し込む」
    SUBA GR1,GR2    ; カウンタ i の減算 <GR1>-1→GR1
    JNZ LOOP        ; カウンタ i が 0 でなければ LOOPに繰り返し
    ST  GR0,RSLT    ; <GR0> → RSLT
    RET
DAT    DC 100      ; 被乗数 m
N      DC 3        ; 乗数 n (ループ回数)
RSLT   DS 1
END
    
```

- ◆ このプログラムで LOOP 番地の ADDA は何回実行されるか？
まずはプログラムを読んで予想し、次に WCASL-IIで実行して確かめよ。
- ◆ カウンタ $i \geq 1$ の判定の代りに JNZ, 即ち $i \neq 0$ の判定を行っている事に注意
- ◆ DATやNを変えて、実行して見よ。

17

アセンブリ言語CASL II

- 演習 **MULTN** ... プログラム**MULT0**を 0 から $n-1$ まで(n は **N**番地の内容)ループするように書き換えよ。

```

MULTN START
    LAD GR0,0      ; 0 → GR0 (結果の格納用レジスタを初期化)
    LAD GR2,1      ; カウンタ i の加算用の 1
LOOP   ADDA GR0,DAT ; 「処理=GR0にmを足し込む」
    ST  GR0,RSLT    ; <GR0> → RSLT
    RET
DAT    DC 100      ; 被乗数 m
N      DC 3        ; 乗数 n (ループ回数)
RSLT   DS 1
END
    
```

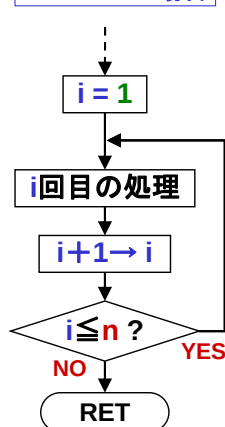
- ◆ JMIを使ったカウンタ $i < N$ の判定の代りに JNZ, 即ち $i \neq N$ の判定を行っても良い

18

COMET II マシンの機械命令

- ループするプログラムの準備

$i = 1 \sim n$ の場合



GRx ... ループカウンタ i のためのレジスタ
 GRy ... 値1の入ったレジスタ (i を $+1$ するため)
 label ... ループ処理の先頭の命令に付与
 分岐方法 ... どんな演算で FR をセットし、
 どの条件分岐命令でそれをテストするか？

ヒント(1) : CPL i, n ($i - n$ に相当)の後、
 SF,ZF=0,0ならば $i > n$
 SF,ZF=0,1ならば $i = n$
 SF,ZF=1,0ならば $i < n$

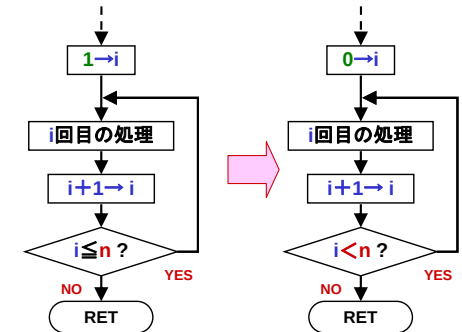
ヒント(2) : JZE ... ZF=1 ならば jump
 JNZ ... ZF=0 ならば jump
 JMI ... SF=1 ならば jump
 JPL ... SF,ZF=0,0 ならば jump

19

条件分岐命令の使い分け

- ◆ COMET II が備える条件分岐命令の分岐条件

- JPL (jump on plus)
直前の演算結果 > 0
直前の比較 $\langle gr1 \rangle > \langle gr2 \rangle \mid \langle e \rangle$
- JMI (jump on minus)
直前の演算結果 < 0
直前の比較 $\langle gr1 \rangle < \langle gr2 \rangle \mid \langle e \rangle$
- JZE (jump on zero)
直前の演算結果 $= 0$
直前の比較 $\langle gr1 \rangle = \langle gr2 \rangle \mid \langle e \rangle$
- JNZ (jump on non zero)
直前の演算結果 $\neq 0$
直前の比較 $\langle gr1 \rangle \neq \langle gr2 \rangle \mid \langle e \rangle$



- ◆ 実現したいアルゴリズムで $\leq$ や $\geq$ がある時、ループカウンタ値をずらして $<$ や $>$ に置き換えられることもある。

可能なら、ループカウンタ値の増減を逆に使用しても良い。

20

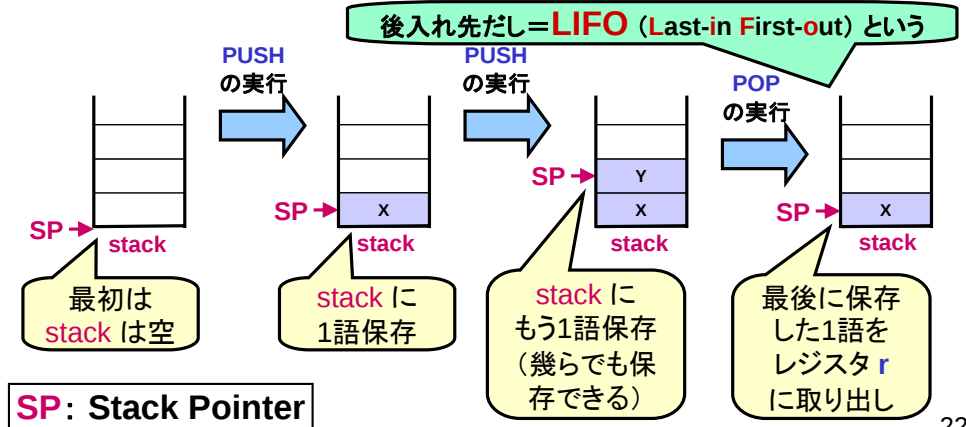
スタックとサブルーチン

21

COMET II マシンの機械命令

• PUSH と POP による stack 操作

- PUSH adr[,xr] ... 実効番地(adr[,xr]) (X,Y等とする)を stack の先頭に push
- POP gr ... stack の先頭の内容をレジスタ gr に pop



22

COMET II マシンの機械命令

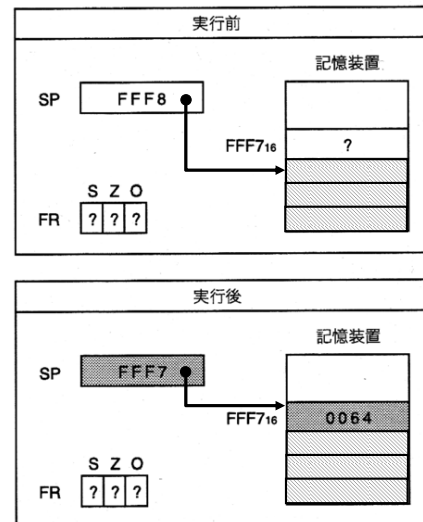
(教科書 p.26)

• プッシュ(push)命令

[書き方] PUSH adr[,xr]

[機能] $\langle SP \rangle - 1 \rightarrow SP;$
 $e \rightarrow \langle SP \rangle$

[実行例] PUSH 100



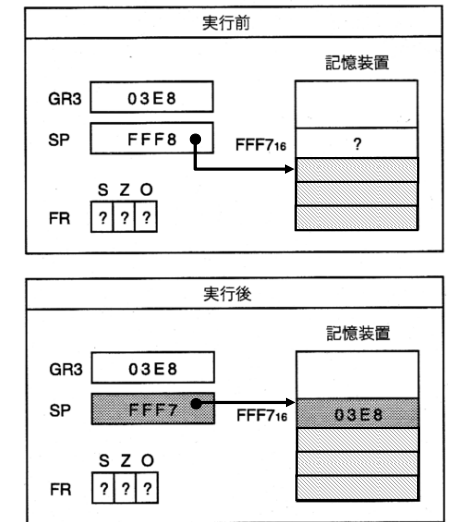
23

COMET II マシンの機械命令

(教科書 p.26)

• プッシュ(push)命令

[実行例] PUSH 0,GR3



24

COMET II マシンの機械命令

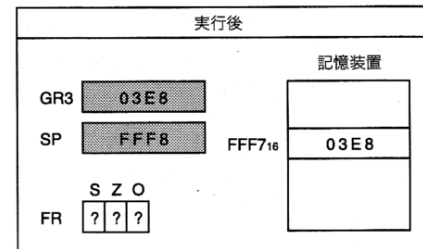
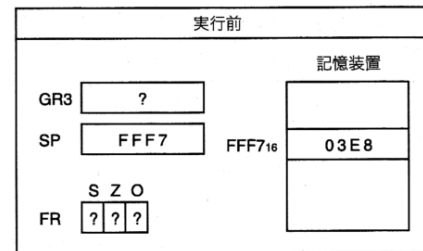
(教科書 p.27)

• ポップ(pop)命令

[書き方] POP gr

[機能] <<SP>> → GRn;
<SP>+1 → SP

[実行例] POP GR3



25

お知らせ

休講

★来週**6月6日**は休講にさせていただきます。

★次回(第9回)は6月13日です。

宿題

「第8回宿題」を次回(6/13)授業の冒頭に副手に提出すること

第2回テスト

★次回**6月13日**の授業時間の冒頭

★ループ, シフト等の論理演算, 他全範囲

26

お知らせ(その2)

補習

• 第2回

- 日時: **6/8(金)5限**(16:30~18:00)(延長有?)
- 場所: **2号館10階2103室**(第1回補習の場所と同じ)

• 講師: 副手の増田君

• 希望者:

- 直接上記の日時・場所に集合して下さい
- できれば事前に「どこを詳しく説明して欲しい」かをメールすること

To: nakajima@im.dendai.ac.jp

CC: 11fmi43@ms.dendai.ac.jp

27