基礎コンピュータ工学
第５章 機械語プログラミング
（パート７）

https://github.com/tctsigemura/TecTextBook

本スライドの入手：

ジャンプ命令（７種類）の残り３種類

無条件ジャンプ命令： **プログラムの流れ**を指定のアドレスに飛ばす.

- *JMP*（*Jump*）**命令**：いつもジャンプする.

条件ジャンプ命令： ある条件のときだけジャンプする.

- *JZ*（*Jump on Zero*）**命令**：$Z = 1$ ならジャンプ
- *JC*（*Jump on Carry*）**命令**：$C = 1$ ならジャンプ
- *JM*（*Jump on Minus*）**命令**：$S = 1$ ならジャンプ
- *JNZ*（*Jump on Not Zero*）**命令**：$Z = 0$ ならジャンプ
- *JNC*（*Jump on Not Carry*）**命令**：$C = 0$ ならジャンプ
- *JNM*（*Jump on Not Minus*）**命令**：$S = 0$ ならジャンプ

JNZ（Jump on Not Zero）命令

Z フラグが 0 なら（計算結果が 0 でないなら）ジャンプする.

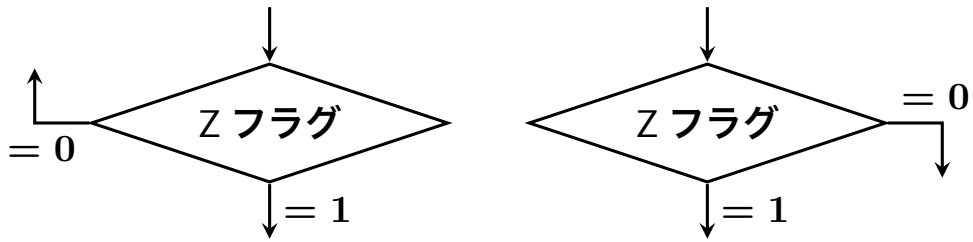
フラグ： 変化しない.

ニーモニック： `JNZ EA`　　(`if(Z=0) PC ← EA`)

命令フォーマット： 2 バイトの長さを持つ.

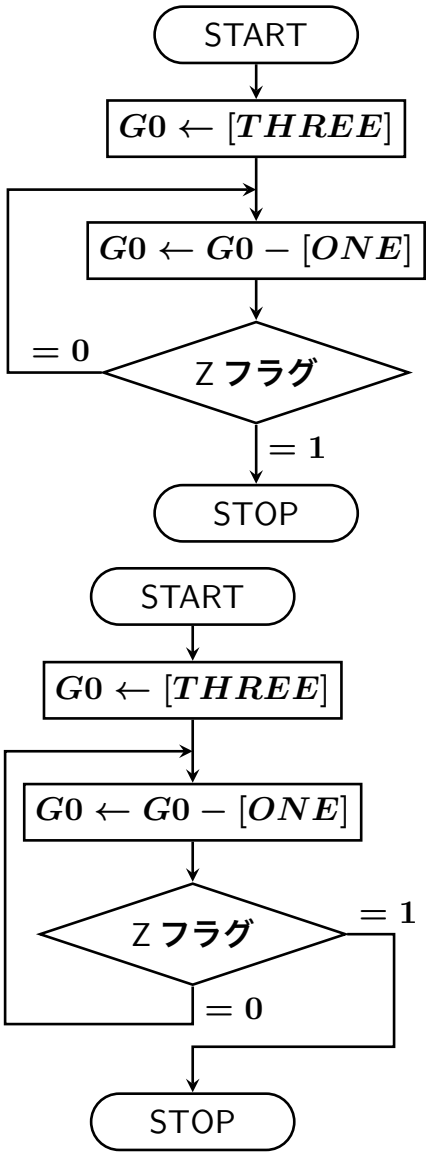
第 1 バイト		第 2 バイト
OP	GR XR	
1011_2	01_2 *XR*	*aaaa aaaa*

フローチャート： ある程度，自由にアレンジしてよい.



JNZ 命令の使用例（JZ 命令との比較）

ループを 3 回，繰り返すプログラム



番地	機械語	ラベル	ニーモニック	
00	10 07		LD	G0,THREE
02	40 08	LOOP	SUB	G0,ONE
04	B4 02		JNZ	LOOP
06	FF		HALT	
07	03	THREE	DC	3
08	01	ONE	DC	1

番地	機械語	ラベル	ニーモニック	
00	10 09		LD	G0,THREE
02	40 0A	LOOP	SUB	G0,ONE
04	A4 08		JZ	STOP
06	A0 02		JMP	LOOP
08	FF	STOP	HALT	
09	03	THREE	DC	3
0A	01	ONE	DC	1

JNC（Jump on Not Carry）命令

C フラグが 0 なら（オーバーフローしていないなら）ジャンプする.

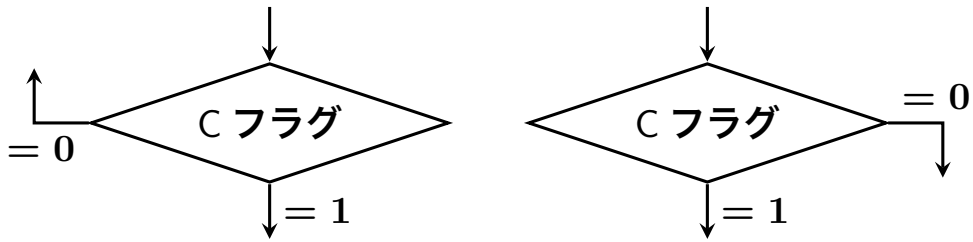
フラグ： 変化しない.

ニーモニック： `JNC EA`　　(`if(C=0) PC ← EA`)

命令フォーマット： 2 バイトの長さを持つ.

第 1 バイト		第 2 バイト
OP	GR XR	
1011_2	10_2 *XR*	*aaaa aaaa*

フローチャート： ある程度，自由にアレンジしてよい.



JNM（Jump on Not Minus）命令

S フラグが 0 なら（正かゼロなら）ジャンプする.

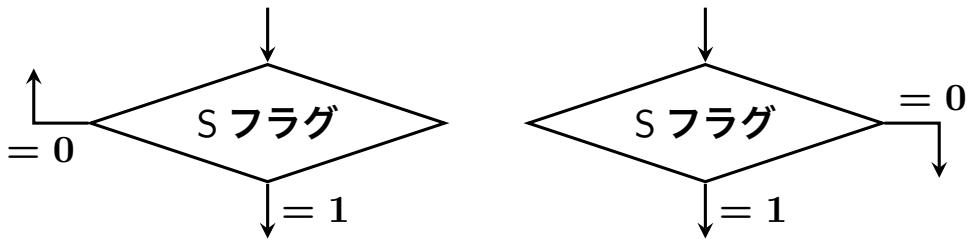
フラグ： 変化しない.

ニーモニック： `JNM EA`　　（`if(S=0) PC ← EA`）

命令フォーマット： 2 バイトの長さを持つ.

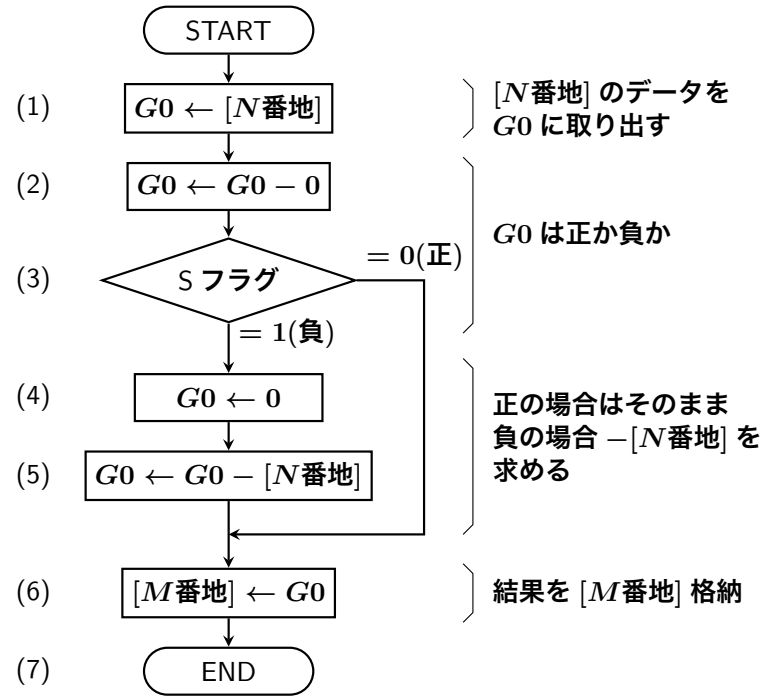
第 1 バイト		第 2 バイト
OP	GR XR	
1011_2	11_2 *XR*	*aaaa aaaa*

フローチャート： ある程度，自由にアレンジしてよい.



条件判断の例（JNM 使用）

絶対値を求めるプログラム（例題 5-1 の改良）



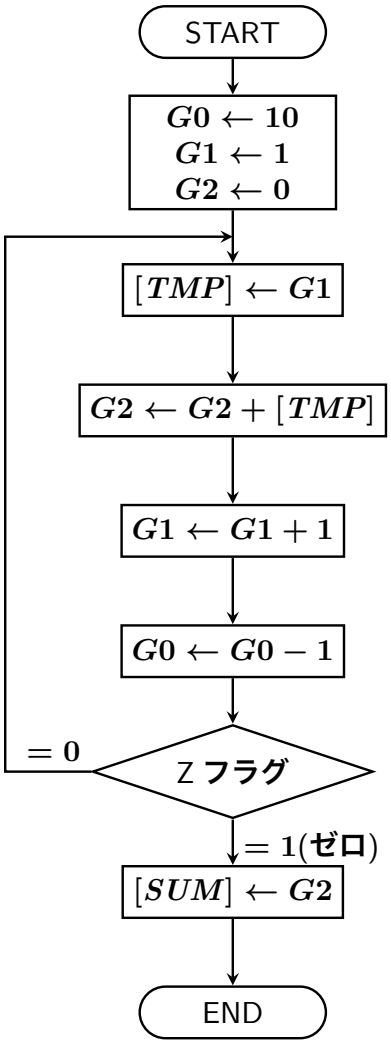
注意：$[N番地]$ は、N 番地に格納されているデータのこと

番地	機械語	ラベル	ニーモニック	
00	10 0E	START	LD	G0,N
02	40 0D		SUB	G0,ZERO
04	BC 0A		JNM	L1
06	10 0D		LD	G0,ZERO
08	40 0E		SUB	G0,N
0A	20 0F	L1	ST	G0,M
0C	FF		HALT	
0D	00	ZERO	DC	0
0E	FF	N	DC	-1
0F	00	M	DS	1

- JNM を使用したほうが短くなる.

繰り返しの例

$1+2+3+\ldots+10$ を計算する．（例題 5-2 の改良）



JZ 使用（以前の例）

```
        LD    G0, N
        LD    G1, ONE
        LD    G2, ZERO
LOOP    ST    G1, TMP
        ADD   G2, TMP
        ADD   G1, ONE
        SUB   G0, ONE
        JZ    STOP
        JMP   LOOP
STOP    ST    G2, SUM
        HALT
N       DC    10
ONE     DC    1
ZERO    DC    0
TMP     DS    1
SUM     DS    1
```

JNZ 使用（改良版）

```
        LD    G0, N
        LD    G1, ONE
        LD    G2, ZERO
LOOP    ST    G1, TMP
        ADD   G2, TMP
        ADD   G1, ONE
        SUB   G0, ONE
        JNZ   LOOP
        ST    G2, SUM
        HALT
N       DC    10
ONE     DC    1
ZERO    DC    0
TMP     DS    1
SUM     DS    1
```

CMP（Compare）命令（比較命令）

レジスタの値とメモリの値を比較しフラグを変化させる.
比較には引き算を使用する.

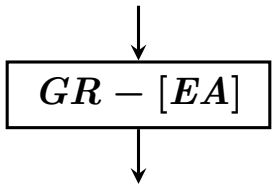
フラグ： 計算結果により変化する.

ニーモニック： `CMP GR,EA` （GR - [EA]）

命令フォーマット： 2バイトの長さを持つ.

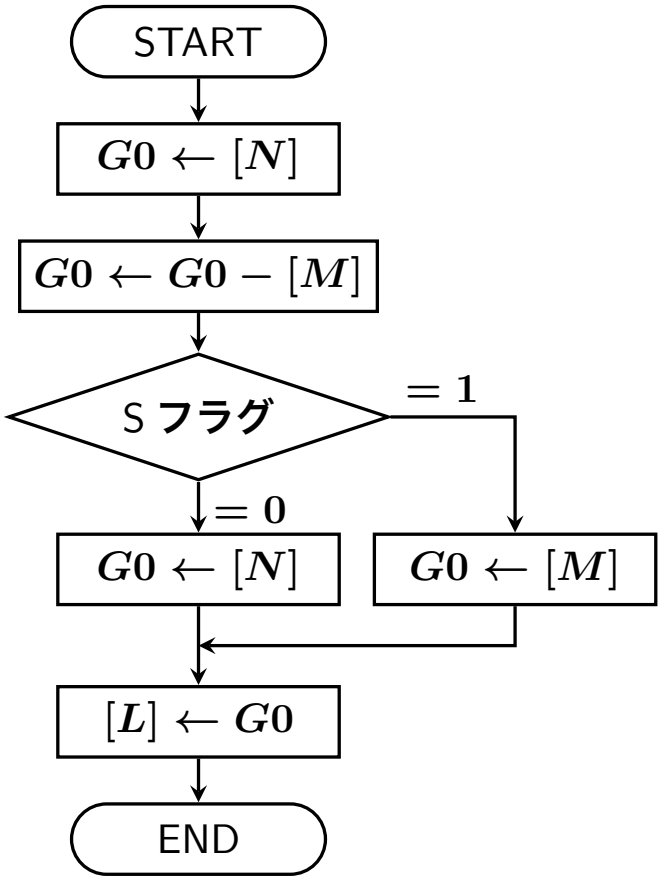
第1バイト		第2バイト
OP	GR XR	
0101_2	*GR XR*	*aaaa aaaa*

フローチャート： 値を保存しない引き算の意味.



大小比較（演習問題の解答と同じ）

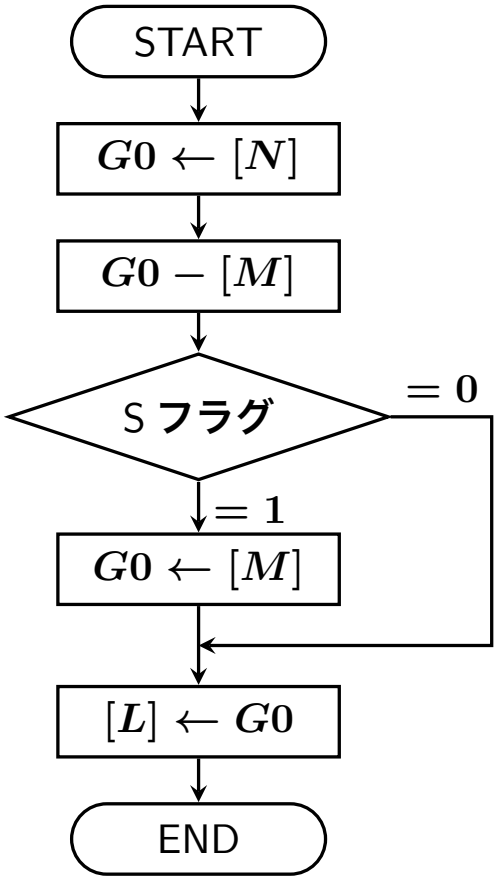
M と N 大きい方を選択して L に格納する.




```
     LD    G0, N
     SUB   G0, M
     JM    L1

     LD    G0, N
     JMP   L2

L1   LD    G0, M

L2   ST    G0, L
     HALT

N    DS    1
M    DS    1
L    DS    1
```

大小比較（CMP，JNM 命令を使用して改良）

M と N 大きい方を選択して L に格納する.




```
      LD    G0, N
      CMP   G0, M
      JNM   L1

      LD    G0, M

L1    ST    G0, L
      HALT

N     DS    1
M     DS    1
L     DS    1
```

まとめ

学んだこと

- 今回の命令は，必須ではないが，あると便利なもの．
- 残りの条件ジャンプ命令（JNZ，JNC，JNM）
- 比較命令（CMP）
- 新しい命令を使用して改良したプログラム

演習（宿題）

- **割り算プログラム：**M 番地のデータを N 番地のデータで割り，商を K 番地，余りを L 番地に格納するプログラム
- データはどれも符号なし整数とする．
- 割り算は引き算の繰り返しでできる．