

## 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4：分岐命令)

<https://github.com/tctsigemura/TecTextBook>

本スライドの入手：



基礎コンピュータ工学第5章 機械語プログラミング

1 / 10

## プログラムの流れ

- プログラムは番地順に実行される（逐次実行）。
- 実行が進んでいく流れを「プログラムの流れ」と呼ぶ。
- 「プログラムの流れ」は PC によって管理されている。
- 通常、PC は増加していく。
- 「プログラムの流れ」を別のアドレスに変えることも必要。
  - 条件によって処理内容を変更したい場合。
  - 同じ処理内容を繰り返したい場合。
- 「プログラムの流れ」を変える命令を**ジャンプ命令**と呼ぶ。  
「プログラムの流れ」を飛ばす = PC にアドレスをロードする

基礎コンピュータ工学第5章 機械語プログラミング

2 / 10

## ジャンプ命令（7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。

**条件ジャンプ命令：** 条件が成立したときだけジャンプする。

無条件ジャンプ命令（JMP 命令）の役割イメージ

| 番地 | 機械語   | ラベル | ニーモニック     |
|----|-------|-----|------------|
| 00 | 10 08 |     | LD GO,08H  |
| 02 | 30 09 |     | ADD GO,09H |
| 04 | 20 0A |     | ST GO,0AH  |
| 06 | A0 0B |     | JMP 0BH    |
| 08 | 12    |     | データ        |
| 09 | 34    |     | データ        |
| 0A | 00    |     | データ        |
| 0B | 30 09 |     | ADD GO,09H |
| 0D | ...   |     | ...        |

基礎コンピュータ工学第5章 機械語プログラミング

3 / 10

## JMP (Jump) 命令（ニーモニックと命令フォーマット）

**無条件ジャンプ命令：** JMP (Jump) 命令

**ニーモニック：** JMP EA (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              |           |
| 1010 <sub>2</sub> | 00 <sub>2</sub> XR | aaaa aaaa |

例：メモリの 10<sub>16</sub> 番地へ飛ぶ（ジャンプする）。

**ニーモニック：** JMP 10H 動作：PC に 10<sub>16</sub> をロードする。

**命令フォーマット：** 10H を反映する。

| 第1バイト             |                                 | 第2バイト                  |
|-------------------|---------------------------------|------------------------|
| OP                | GR XR                           |                        |
| 1010 <sub>2</sub> | 00 <sub>2</sub> 00 <sub>2</sub> | 0001 0000 <sub>2</sub> |

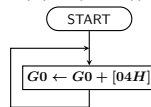
基礎コンピュータ工学第5章 機械語プログラミング

4 / 10

## JMP (Jump) 命令（フローチャートとプログラム例）

**JMP 命令のフローチャート：** ←, →, ↑, ↓ など

**フローチャートの例：** ADD 命令を永遠に繰り返す。（無限ループ）



**プログラムの例：** 0 番地の ADD 命令を永遠に繰り返す。（無限ループ）

| 番地 | 機械語   | ラベル | ニーモニック     |
|----|-------|-----|------------|
| 00 | 30 04 |     | ADD G0,04H |
| 02 | A0 00 |     | JMP 00H    |

**演習（1）：** 上のプログラムを 4 番地に 1 を格納した状態で実行する。  
STOP ボタンでプログラムを停止し G0 の値を確認する。

基礎コンピュータ工学第5章 機械語プログラミング

5 / 10

## ラベル

ニーモニックだけでプログラムを完結させるために使用する。

- JMP 命令のプログラム例では、ジャンプ先のアドレスをニーモニックに中に数値で書いた。
- 機械語の番地が決まらなるとニーモニックが完成しない。一方で、ニーモニックを書かないと機械語が完成しない。
- ニーモニックだけでプログラムを完結させる必要がある。  
→ 場所（アドレス）に名前（ラベル）を付ける。

前のプログラムをラベルを使って書き直したもの。

| 番地 | 機械語   | ラベル  | ニーモニック     |
|----|-------|------|------------|
| 00 | 30 04 | LOOP | ADD G0,04H |
| 02 | A0 00 |      | JMP LOOP   |

LOOP = 「輪」 ... 意味を持った名前を付けるとより良い。

基礎コンピュータ工学第5章 機械語プログラミング

6 / 10

## DC (Define Constant) 命令

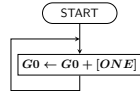
まだ、データ部分がニーモニックで表現できていない。

- データもニーモニックで表現できる必要がある。
- DC 命令はデータを記述するための疑似命令 (≠ 機械語命令)
- ニーモニック: DC データの値
- 前のプログラムを DC 命令を使って書き直したもの。

| 番地 | 機械語   | ラベル  | ニーモニック      |
|----|-------|------|-------------|
| 00 | 30 04 | LOOP | ADD G0, ONE |
| 02 | A0 00 |      | JMP LOOP    |
| 04 | 01    | ONE  | DC 1        |

データの番地 (04H) もラベル (ONE) で参照できる。

- フローチャートの例



## DS (Define Storage) 命令

結果を格納する領域を作るための疑似命令。

ニーモニック: DS 領域の大きさ (バイト数)

プログラムの例: X番地とY番地のデータの和をZ番地に求める。  
(7番地と8番地のデータの和を9番地に求めると同じ)

| 番地 | 機械語   | ラベル | ニーモニック    |
|----|-------|-----|-----------|
| 00 | 10 07 |     | LD G0, X  |
| 02 | 30 08 |     | ADD G0, Y |
| 04 | 20 09 |     | ST G0, Z  |
| 06 | FF    |     | HALT      |
| 07 | 12    | X   | DC 12H    |
| 08 | 34    | Y   | DC 34H    |
| 09 | 00    | Z   | DS 1      |

## DC 命令と DS 命令の使い分け

入力となるデータを色々変化させたい場合。

プログラムの例: X番地のデータに1を加えたものをZ番地に求める。

| 番地 | 機械語   | ラベル | ニーモニック      |
|----|-------|-----|-------------|
| 00 | 10 08 |     | LD G0, X    |
| 02 | 30 07 |     | ADD G0, ONE |
| 04 | 20 09 |     | ST G0, Z    |
| 06 | FF    |     | HALT        |
| 07 | 01    | ONE | DC 1        |
| 08 | 00    | X   | DS 1        |
| 09 | 00    | Z   | DS 1        |

**DC と DS の区別:** 値が変化しないものを DC で準備する。  
入力になるものは、典型的な値を DC で準備する。  
入力になるものは、後で決めるので DS で場所を確保する。  
出力は、DS で場所を確保する。

## まとめ

学んだこと

- 無条件ジャンプ命令 (JMP 命令)
- ラベル
- データを表現する命令 (DC 命令)
- データ領域を予約する命令 (DS 命令)

演習 (2) (以下の目的で演習を行う)

- PC の役割を再確認する。
- PC と JMP 命令の関係を調べる。
- 計算結果とフラグの関係を調べる。 ※
- ステップモード実行の練習をする。
- ブレークモード実行の練習をする。

※次回はフラグの値を条件にするジャンプ命令を学ぶ。