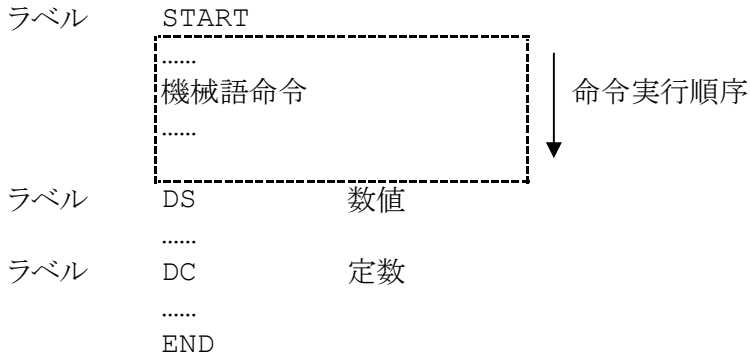


# CASL II プログラミング

## プログラムの記述スタイル



各行の命令は以下の3つの部分から構成される。

ラベル      命令コード      オペランド

ラベルはプログラム中でその行を参照する際に利用され、値はそのアドレスである。

START に付けるラベルはプログラム名としての意味がある。

オペランドは命令の作用対象を表す。

### アセンブラ命令

START      プログラムの開始

END        プログラムの終り

DS        領域の確保 (Define Storage)、数値は確保する語数

DC        定数の定義 (Define Constant)

          数値のみは10進数 例) 12      10進数の12

          #に続けて4桁は16進数 例) #000F      16進数のF

          ' で囲むと文字定数 例) 'A'      文字の A

### マクロ命令

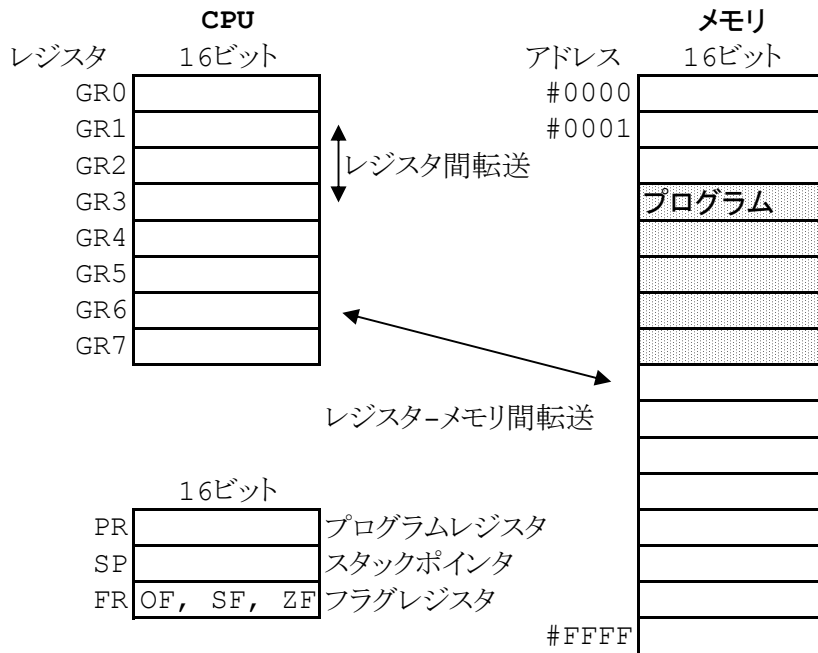
IN        読み込み命令 試験に出題されたことはない。

OUT      書出し命令 試験に出題されたことはない。

RPUSH    レジスタの一括退避 (Register PUSH)

RPOP     レジスタの一括復帰 (Register POP)

## 処理の基本概念



レジスタを介さないメモリ-メモリ間転送は出来ない。  
 フラグレジスタはオーバーフロー、サイン、ゼロの3ビットである。  
 プログラムレジスタとスタックポインタに関して試験で出題されたことはない。

## 機械語命令

### データ転送命令

LD GRn, GRm ; GRn ← GRm レジスタ-レジスタ間  
 LD GRn, アドレス ; GRn ← アドレスの内容 レジスタ-メモリ間  
 LD GRn, アドレス, GRm ; GRn ← 実効アドレスの内容  
 ; 実効アドレスは アドレス + GRm(m≠0)の内容  
 LD GRn, =数値 ; GRn ← 定数

この命令はアセンブラによって以下のように翻訳される。

```
LD GRn, ラベル
.....
ラベル DC 数値
```

したがって、レジスタ-メモリ間転送になるが、試験ではアセンブラについて出題されることはないので、定数をレジスタに設定する命令と理解しておけばよい。

ST GRn, アドレス ; GRn → アドレスの内容 レジスタ-メモリ間  
 ST GRn, アドレス, GRm ; 実効アドレスは アドレス + GRm(m≠0)の内容

### アドレス転送命令

LAD GRn, アドレス ; GRn ← アドレス  
 LAD GRn, アドレス, GRm ; GRn ← 実効アドレス  
 LAD GR1, 1 ; GR1 ← 1  
 LAD GR1, 1, GR1 ; GR1 ← GR1+1  
 LAD GR1, -1, GR1 ; GR1 ← GR1-1

### 算術加算命令

ADDA	GRn, GRm	; GRn ← GRn + GRm
ADDA	GRn, アドレス	; GRn ← GRn + アドレスの内容
ADDA	GRn, アドレス, GRm	; GRn ← GRn + 実効アドレスの内容
ADDA	GRn, =数値	; GRn ← GRn + 定数

### 論理加算命令

ADDL	GRn, GRm	; GRn ← GRn + GRm
ADDL	GRn, アドレス	; GRn ← GRn + アドレスの内容
ADDL	GRn, アドレス, GRm	; GRn ← GRn + 実効アドレスの内容
ADDL	GRn, =数値	; GRn ← GRn + 定数

### 算術減算命令

SUBA	GRn, GRm	; GRn ← GRn - GRm
SUBA	GRn, アドレス	; GRn ← GRn - アドレスの内容
SUBA	GRn, アドレス, GRm	; GRn ← GRn - 実効アドレスの内容
SUBA	GRn, =数値	; GRn ← GRn - 定数

### 論理減算命令

SUBL	GRn, GRm	; GRn ← GRn - GRm
SUBL	GRn, アドレス	; GRn ← GRn - アドレスの内容
SUBL	GRn, アドレス, GRm	; GRn ← GRn - 実効アドレスの内容
SUBL	GRn, =数値	; GRn ← GRn - 定数

### 論理積命令

AND	GRn, GRm	; GRn ← GRn AND GRm
AND	GRn, アドレス	; GRn ← GRn AND アドレスの内容
AND	GRn, アドレス, GRm	; GRn ← GRn AND 実効アドレスの内容
AND	GRn, =数値	; GRn ← GRn AND 定数

### 論理和命令

OR	GRn, GRm	; GRn ← GRn OR GRm
OR	GRn, アドレス	; GRn ← GRn OR アドレスの内容
OR	GRn, アドレス, GRm	; GRn ← GRn OR 実効アドレスの内容
OR	GRn, =数値	; GRn ← GRn OR 定数

### 排他的論理和命令

XOR	GRn, GRm	; GRn ← GRn XOR GRm
XOR	GRn, アドレス	; GRn ← GRn XOR アドレスの内容
XOR	GRn, アドレス, GRm	; GRn ← GRn XOR 実効アドレスの内容
XOR	GRn, =数値	; GRn ← GRn XOR 定数

### 算術左シフト命令

SLA GRn, 数値 ; GRn を数値ビットだけ算術左シフト  
SLA GRn, 数値, GRm ; GRn を(数値+GRmの内容)ビットだけ算術左シフト

### 算術右シフト命令

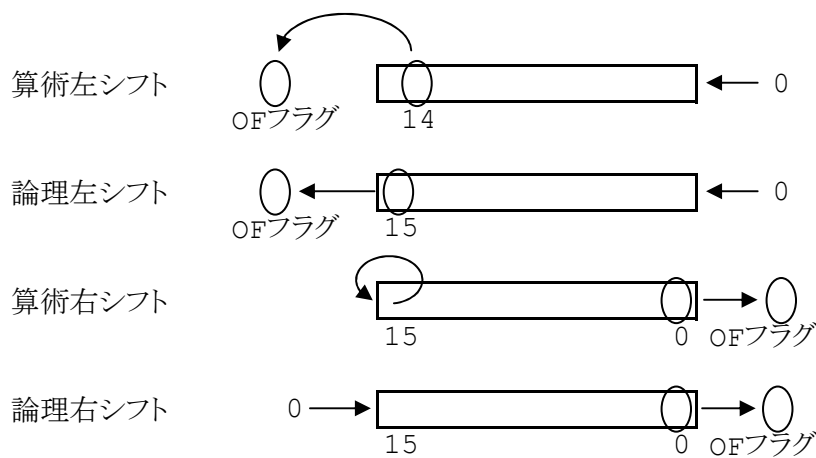
SRA GRn, 数値 ; GRn を数値ビットだけ算術右シフト  
SRA GRn, 数値, GRm ; GRn を(数値+GRmの内容)ビットだけ算術右シフト

### 論理左シフト命令

SLL GRn, 数値 ; GRn を数値ビットだけ論理左シフト  
SLL GRn, 数値, GRm ; GRn を(数値+GRmの内容)ビットだけ論理左シフト

### 論理右シフト命令

SRL GRn, 数値 ; GRn を数値ビットだけ論理右シフト  
SRL GRn, 数値, GRm ; GRn を(数値+GRmの内容)ビットだけ論理右シフト



シフトによって第15ビットが1になればサインフラグがセットされる。  
シフトによって全ビットが0になればゼロフラグがセットされる。

### 算術比較命令

-32768~32767

CPA GRn, GRm ; GRn - GRm  
CPA GRn, アドレス ; GRn - アドレスの内容  
CPA GRn, アドレス, GRm ; GRn - 実効アドレスの内容  
CPA GRn, =数値 ; GRn - 定数

### 論理比較命令

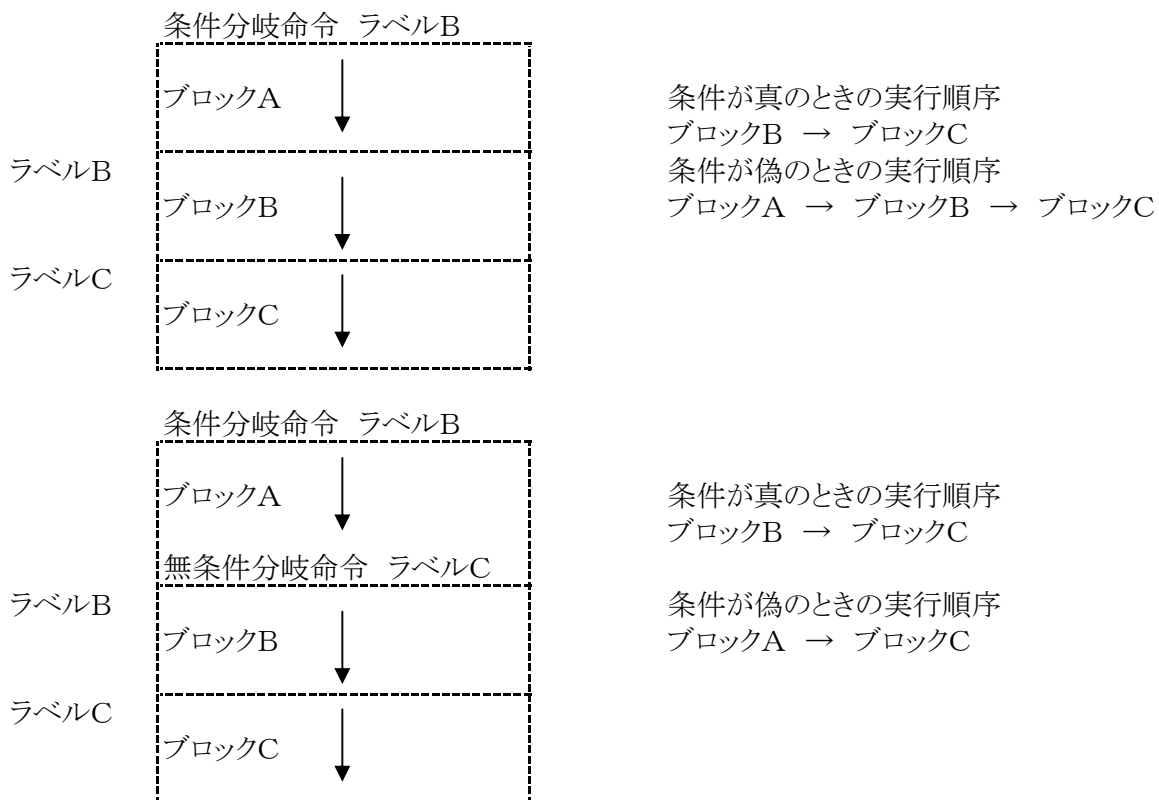
0~65535

CPL GRn, GRm ; GRn - GRm  
CPL GRn, アドレス ; GRn - アドレスの内容  
CPL GRn, アドレス, GRm ; GRn - 実効アドレスの内容  
CPL GRn, =数値 ; GRn - 定数

比較命令によってフラグレジスタが変化する。

> のとき サインフラグ = 0, ゼロフラグ = 0  
= のとき ゼロフラグ = 1  
< のとき サインフラグ = 1

## 条件分岐の基本



### 正分岐命令

JPL アドレス ; SF=0, ZF=0 のときアドレスに分岐  
 JPL アドレス, GRm ; SF=0, ZF=0 のとき実効アドレスに分岐

### 負分岐命令

JMI アドレス ; SF=1 のときアドレスに分岐  
 JMI アドレス, GRm ; SF=1 のとき実効アドレスに分岐

### 非零分岐命令

JNZ アドレス ; ZF=0 のときアドレスに分岐  
 JNZ アドレス, GRm ; ZF=0 のとき実効アドレスに分岐

### 零分岐命令

JZE アドレス ; ZF=1 のときアドレスに分岐  
 JZE アドレス, GRm ; ZF=1 のとき実効アドレスに分岐

### オーバーフロー分岐命令

JOV アドレス ; OF=1 のときアドレスに分岐  
 JOV アドレス, GRm ; OF=1 のとき実効アドレスに分岐

### 無条件分岐命令

JUMP アドレス ; 無条件でアドレスに分岐  
 JUMP アドレス, GRm ; 無条件で実効アドレスに分岐

GRmでアドレスを修飾する際、アドレスを 0 とすれば GRm の内容に分岐させることが出来る。

レジスタ退避命令

PUSH 0, GRn

; GRn の内容をスタックに退避する。

レジスタ復帰命令

POP GRn

; スタックの先頭を GRn に復帰する。

プログラム呼出し命令

CALL アドレス

; 戻り番地(CALL命令の次)をスタックに退避してから  
; プログラムレジスタにアドレスをセットする。

プログラム復帰命令

RET

; スタックの先頭をプログラムレジスタにセットする。  
; 呼出し元に戻る。

スーパーバイザコール命令

SVC アドレス

; OSの機能呼出し 試験に出題されたことはない。

ノーオペレーション命令

NOP

; 動作なし 試験に出題されたことはない。