

CASL II 基本テクニック 25 選

テーマ 01 副プログラムにパラメータを渡す

■ 値渡しと参照渡し

```
TECH01    START
           LAD    GR1,PARAM          ; パラメータリストのアドレス設定
           CALL  SUB
           RET

PARAM     DC    100                  ; 値渡し
           DC    DATA                ; 参照渡し

DATA     DC    10,20,30

;

SUB       START
           LD     GR2,0,GR1           ; GR2 = 100
           LD     GR3,1,GR1           ; GR3 = DATA のアドレス
           LD     GR4,0,GR3           ; GR4 = 10
           LD     GR5,1,GR3           ; GR5 = 20
           LD     GR6,2,GR3           ; GR6 = 30
           RET
           END
```

テーマ 02 レジスタのクリアと負数化

■ 方法は様々

```
TECH02    START
           LAD    GR1,0                ; FR を設定しない
           LD     GR2,=0                ; ZF = 1
           SUBA   GR3,GR3                ; ZF = 1
           XOR    GR4,GR4                ; ZF = 1

;

           SUBA   GR5,GR5
           SUBA   GR5,N                  ; 0 - N
           LD     GR6,N
           XOR    GR6,=#FFFF            ; 2 の補数化
           ADDA   GR6,=1
           RET

N         DC    100
           END
```

テーマ 03 配列の順次参照

■ 先頭アドレスをラベルで指定してオフセットを操作する

```
TECH03  START
        LD      GR2, =0
LOOP    LD      GR1, SRC, GR2
        ST      GR1, DES, GR2
        LAD     GR2, 1, GR2          ; 次のオフセット
        CPA     GR2, =5
        JMI     LOOP
        RET
SRC     DC      10, 20, 30, 40, 50
DES     DS      5                    ; 10, 20, 30, 40, 50 が格納される
        RET
        END
```

テーマ 04 配列の順次参照

■ アドレスを直接操作する

```
TECH04  START
        LAD     GR2, SRC              ; 転送元アドレス設定
        LAD     GR3, DES              ; 転送先アドレス設定
        LD      GR4, =0              ; 転送数の初期値
LOOP    LD      GR1, 0, GR2
        ST      GR1, 0, GR3
        ADDL   GR2, =1                ; 次のアドレス
        ADDL   GR3, =1                ; 次のアドレス
        LAD     GR4, 1, GR4          ; 転送数のインクリメント
        CPA     GR4, =5
        JMI     LOOP
        RET
SRC     DC      10, 20, 30, 40, 50
DES     DS      5                    ; 10, 20, 30, 40, 50 が格納される
        END
```

テーマ 05 数値から文字へ変換

■ 文字の利用

```
TECH05  START
        LD      GR1, X
        ADDA   GR1, = '0'            ; 文字へ変換
        ST      GR1, PRN
        OUT    PRN, LEN
        RET
X       DC      7
PRN     DS      1
LEN     DC      1
        END
```

テーマ 06 数値から文字へ変換

■ 文字コードの利用 算術演算

```
TECH06    START
           LD      GR1,X
           ADDA   GR1,=#0030      ; 文字へ変換
           ST      GR1,PRN
           OUT    PRN,LEN        ; 文字の 7 が出力される
           RET
X          DC      7
PRN        DS      1
LEN        DC      1
           END
```

テーマ 07 数値から文字へ変換

■ 文字コードの利用 ビット演算

```
TECH07    START
           LD      GR1,X
           OR      GR1,=#0030      ; 文字へ変換
           ST      GR1,PRN
           OUT    PRN,LEN        ; 文字の 7 が出力される
           RET
X          DC      7
PRN        DS      1
LEN        DC      1
           END
```

テーマ 08 数値から文字へ変換

■ 文字列テーブルの利用

```
TECH08    START
           LD      GR1,X
           LD      GR1,NUM,GR1    ; テーブルから抽出
           ST      GR1,PRN
           OUT    PRN,LEN        ; 文字の 7 が出力される
           RET
X          DC      7
NUM        DC      '0123456789'
PRN        DS      1
LEN        DC      1
           END
```

テーマ 09 文字から数値へ変換

■ 文字の利用

```
TECH09    START
           LD      GR1,X
           SUBA   GR1,='0'           ; 数値へ変換
           ST      GR1,PRN
           RET
X          DC      '7'
PRN        DS      1                 ; 数値の 7 が格納される
           END
```

テーマ 10 文字から数値へ変換

■ 文字コードの利用 算術演算

```
TECH10    START
           LD      GR1,X
           SUBA   GR1,=#0030         ; 数値へ変換
           ST      GR1,PRN
           RET
X          DC      '7'
PRN        DS      1                 ; 数値の 7 が格納される
           END
```

テーマ 11 文字から数値へ変換

■ 文字コードの利用 ビット演算

```
TECH11    START
           LD      GR1,X
           AND    GR1,=#000F         ; 数値へ変換
           ST      GR1,PRN
           RET
X          DC      '7'
PRN        DS      1                 ; 数値の 7 が格納される
           END
```

テーマ 12 指定したビット位置の反転

■ シフトによるビットパターンの作成

```
TECH12    START
           LD      GR1,X
           LD      GR2,N
           LD      GR3,BIT
           SLL    GR3,-1,GR2         ; N-1 ビット左シフト
           XOR    GR1,GR3           ; GR1 = 1010 1011 1110 1101
           RET
X          DC      #ABCD
N          DC      6                 ; 右から 6 ビット目を反転
BIT        DC      #0001
           END
```

テーマ 13 上位 8 ビットと下位 8 ビットの入替

■ 定数によるビットの抽出

```
TECH13    START
           LD      GR1,X
           LD      GR2,GR1          ; データを別のレジスタに保存
           AND     GR1,=#00FF      ; 下位 8 ビットの抽出
           SLL    GR1,8            ; 上位 8 ビットに移動
           AND     GR2,=#FF00      ; 上位 8 ビットの抽出
           SRL    GR2,8            ; 下位 8 ビットに移動
           OR     GR1,GR2          ; GR1 = #CDAB
X          DC      #ABCD
           RET
           END
```

テーマ 14 ビット 1 の数を数える

■ データの右シフト

```
TECH14    START
           LD      GR1,X
           JZE     FIN
           LD      GR2,=0          ; カウンタの 0 クリア
LOOP       LD      GR3,GR1        ; データを別のレジスタに保存
           AND     GR3,=#0001     ; ビット判定
           JZE     SKIP
           ADDA   GR2,=1          ; ビットが 1 ならばカウンタのインクリメント
SKIP       SRL    GR1,1          ; データの 1 ビット右論理シフト
           JZE     FIN            ; ビット 1 が残っていなければ終了
           JUMP   LOOP
FIN        RET
X          DC      #ABCD          ; GR2 = 10
           END
```

テーマ 15 ビット 1 の数を数える

■ 検査用ビット列の左シフト

```
TECH15    START
           LD      GR1,X
           JZE     FIN
           LD      GR2,=0          ; カウンタの 0 クリア
           LD      GR4,BIT        ; 検査用ビット列の初期値設定
LOOP       LD      GR3,GR1        ; データを別のレジスタに保存
           AND     GR3,GR4        ; ビット判定
           JZE     SKIP
           ADDA   GR2,=1          ; ビットが 1 ならばカウンタのインクリメント
SKIP       SLL    GR4,1          ; 検査用ビット列の 1 ビット左シフト
           JZE     FIN            ; 16 ビット検査したら終了
           JUMP   LOOP
FIN        RET
X          DC      #ABCD          ; GR2 = 10
BIT        DC      #0001
           END
```

テーマ 16 ビット 1 の数を数える

■ 検査用ビット列の右シフト

```
TECH16    START
          LD      GR1,X
          JZE     FIN
          LD      GR2,=0          ; カウンタの 0 クリア
          LD      GR4,BIT        ; 検査用ビット列の初期値設定
LOOP      LD      GR3,GR1        ; データを別のレジスタに保存
          AND     GR3,GR4        ; ビット判定
          JZE     SKIP
          ADDA   GR2,=1          ; ビットが 1 ならばカウンタのインクリメント
SKIP      SRL    GR4,1          ; 検査用ビット列の 1 ビット右論理シフト
          JZE     FIN            ; 16 ビット検査したら終了
          JUMP   LOOP
FIN       RET
X         DC     #ABCD          ; GR2 = 10
BIT       DC     #8000
          END
```

テーマ 17 ビット 1 の数を数える

■ 符号ビットの利用

```
TECH17    START
          LD      GR1,X
          JZE     FIN
          LD      GR2,=0          ; カウンタの 0 クリア
LOOP      LD      GR1,GR1        ; GR1 のフラグをセット
          JZE     FIN
          JPL    SKIP
          ADDA   GR2,=1          ; SF = 1 ならばカウンタのインクリメント
SKIP      SLL   GR1,1          ; データの 1 ビット左シフト
          JUMP   LOOP
FIN       RET
X         DC     #ABCD          ; GR2 = 10
          END
```

テーマ 18 ビット 1 の数を数える

■ オーバフローフラグの利用 左シフト インクリメントがフラグに影響する

```
TECH18    START
          LD      GR1,X
          JZE     FIN
          LD      GR2,=0          ; カウンタの 0 クリア
LOOP      SLL    GR1,1          ; データの 1 ビット左シフト
          JOV    CNT
          JUMP   LOOP
CNT       ADDA   GR2,=1          ; OF = 1 ならばカウンタのインクリメント
          LD     GR1,GR1        ; GR1 のフラグセット
          JZE     FIN
          JUMP   LOOP
FIN       RET
X        DC     #ABCD          ; GR2 = 10
          END
```

テーマ 19 ビット 1 の数を数える

■ オーバフローフラグの利用 左シフト インクリメントがフラグに影響しない

```
TECH19    START
          LD      GR1,X
          JZE     FIN
          LD      GR2,=0          ; カウンタの 0 クリア
LOOP      SLL    GR1,1          ; データの 1 ビット左シフト
          JOV    CNT
          JUMP   LOOP
CNT       LAD    GR2,1,GR2      ; フラグに影響を与えない
          JZE     FIN
          JUMP   LOOP
FIN       RET
X        DC     #ABCD          ; GR2 = 10
          END
```

テーマ 20 ビット 1 の数を数える

■ オーバフローフラグの利用 右シフト

```
TECH20    START
           LD      GR1,X
           JZE     FIN
           LD      GR2,=0          ; カウンタの 0 クリア
LOOP      SRL     GR1,1          ; データの 1 ビット右論理シフト
           JOV     CNT
           JUMP    LOOP
CNT       LAD     GR2,1,GR2      ; フラグに影響を与えない
           JZE     FIN
           JUMP    LOOP
FIN       RET
X         DC      #ABCD          ; GR2 = 10
           END
```

テーマ 21 左から指定した数だけビット 1 を入れる

■ 右シフトの利用

```
TECH21    START
           LD      GR1,=#8000
           LD      GR2,N
           SRA     GR1,-1,GR2     ; N-1 ビット右算術シフト
           RET     ; GR1 = 1111 1100 0000 0000
N         DC      6
           END
```

テーマ 22 右から指定した数だけビット 1 を入れる

■ 右シフトの利用

```
TECH22    START
           LD      GR1,=#8000
           LD      GR2,N
           SRA     GR1,-1,GR2     ; N-1 ビット右算術シフト
           LD      GR2,=16
           SUBA    GR2,N          ; GR2 = 16-N
           SRL     GR1,0,GR2     ; 16-N ビット右論理シフト
           RET     ; GR1 = 0000 0000 0011 1111
N         DC      6
           END
```


テーマ 23 左から指定した数だけビット 0、そこから指定した数だけビット 1 を入れる

■ 右シフトの利用

```
TECH23    START
          LD      GR1,=#8000
          LD      GR2,N
          SRA    GR1,-1,GR2          ; N-1 ビット右算術シフト
          LD      GR2,M
          SRL    GR1,0,GR2          ; M ビット右論理シフト
          RET    GR1 = 0001 1111 1000 0000
N         DC      6
M         DC      3
          END
```

テーマ 24 データの右に指定したビット数だけ 0 を入れる

■ 右シフトと左シフトの利用

```
TECH24    START
          LD      GR1,DATA
          LD      GR2,N
          SRL    GR1,0,GR2          ; N ビット右論理シフト
          SLL    GR1,0,GR2          ; N ビット左シフト
          RET    GR1 = 1010 1011 1100 0000
DATA      DC      #ABCD
N         DC      5
          END
```

テーマ 25 データの左に指定したビット数だけ 0 を入れる

■ 左シフトと右シフトの利用

```
TECH25    START
          LD      GR1,DATA
          LD      GR2,N
          SLL    GR1,0,GR2          ; N ビット左シフト
          SRL    GR1,0,GR2          ; N ビット右論理シフト
          RET    GR1 = 0000 0011 1100 1101
DATA      DC      #ABCD
N         DC      5
          END
```