資　料

# 8080 — 6800　双　方　向
# ソースプログラムコンバータ
# 取　扱　い　説　明　書

昭 和 56 年 3 月

JIPDEC

財団法人　日本情報処理開発協会

この報告書は、日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて、昭和５５年度に実施した「マイクロコンピュータの応用に関する調査研究」の一環としてとりまとめたものであります。

# 目　　次

# 1. 概　　要

## 1.1　システムの目的

本「8080－6800 双方向ソースプログラム・コンバータ」は8ビットマイクロ・コンピュータの代表機種であるインテル社製8080 CPUとモトローラ社製6800 CPU のアセンブリ言語プログラムをソースプログラム・レベルにて双方向に変換するものである。

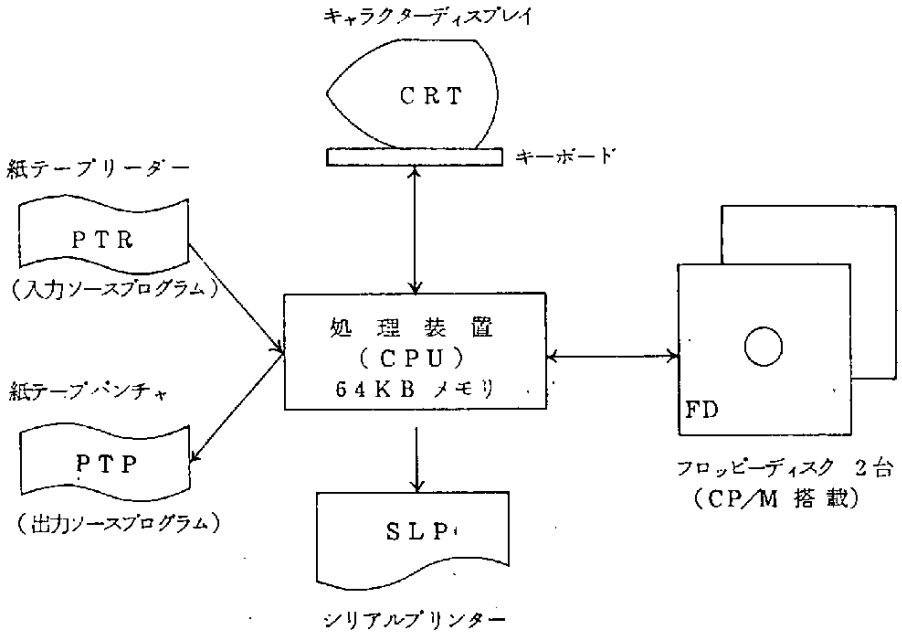
8080ソースプログラム紙テープを入力して変換後6800ソースプログラム紙テープを出力する。又、6800ソースプログラム紙テープを入力して、変換後8080ソースプログラム紙テープを得るシステムである。

8080 CPUと6800 CPU は同じ8ビット系であるとはいえ、レジスターの個数、性質、演算フラグの立ち方、アドレッシング方法、ROMとRAMの配置法、入出力処理と大巾にアーキテクチュアが異なっており、まったく完全な変換は不可能に近い。

本システムにおいては変換後プログラムが効率的で実用にたえられるようにメモリサイズ・処理スピードが過大にならないように変換している。すなわち、命令変換の前にプログラムチェックリストを出力し、演算フラグの立ち方、変換不能命令の表示をしており、オペレータの介入によってターゲット CPU に合わせたプログラム修正を可能とし、その後、命令変換を行ない、プログラム変換リストとターゲット・プログラム紙テープを出力するようにしている。

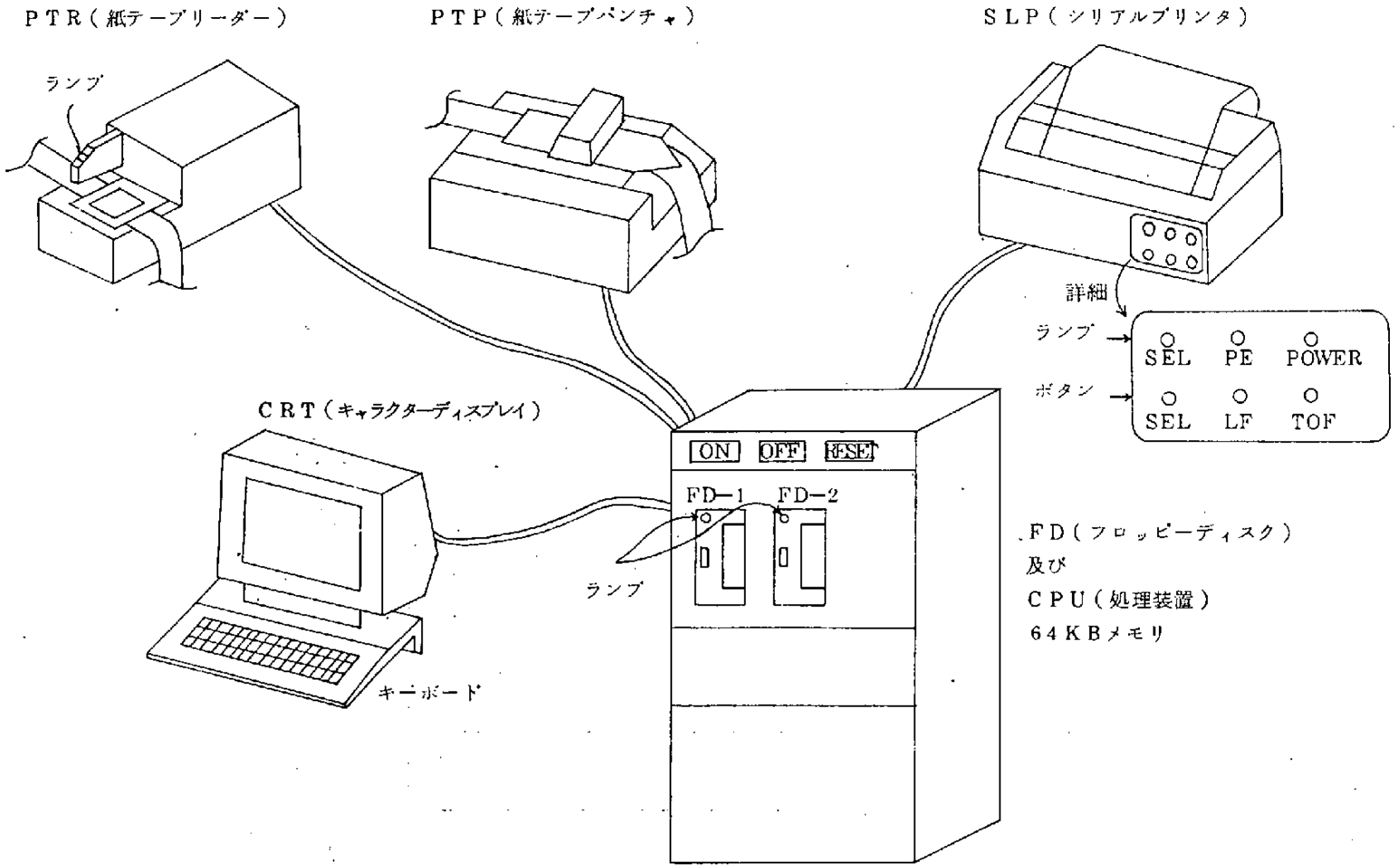
## 1.2　ハードウェア構成

ＣＰ／Ｍを搭載している一般的マイコンプログラム開発システムであり、次のような構成となる。




（注）　フロッピーディスクは単密度と倍密度の両方が扱える。

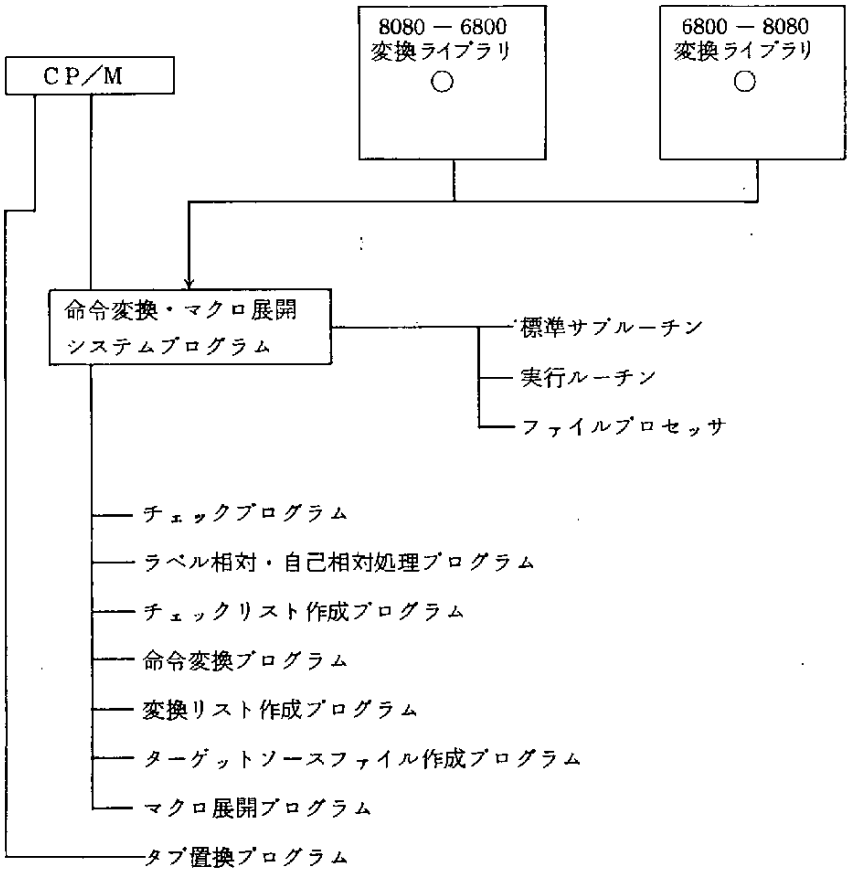
次図にその外観図を示す。


ＰＴＲ（紙テープリーダー）
ＰＴＰ（紙テープパンチャ）
ＳＬＰ（シリアルプリンタ）
ランプ
詳細
ランプ
SEL
PE
POWER
ボタン
SEL
LF
TOF
ＣＲＴ（キャラクターディスプレイ）
ON
OFF
RESET
FD−1
FD−2
ランプ
キーボード
ＦＤ（フロッピーディスク）
及び
ＣＰＵ（処理装置）
６４ＫＢメモリ

外　観　図

## 1.3　ソフトウェア構成

ＣＰ／Ｍの下にて動作し、8080←→6800 変換ライブラリ・ファイルを基本ファイルとして命令変換・マクロ展開システムプログラムを中心にして、次のような構成となっている。




命令変換・マクロ展開システムプログラムは最適化ロジックの入りこんだ変換ライブラリを読みこんで、命令の変換とマクロの展開を基本とする一種のコンパイル＆ゴーシステムであり、作表とかファイル処理もできる言語プロセッサである。

# 2. 変　換　概　要

## 2.1　変換処理の範囲

1. 8080系は8080と8085を対象としたインテル社言語、6800系は6800を対象とするモトローラ社言語を扱う。
2. 入出力、割込関係を除くプログラムモジュールの変換を扱う。
3. 入出力媒体は紙テープとし、変換後のリストを出力する機能を持つ。
4. ソースプログラムとしては完成品を扱う。
5. ソースマシン1命令よりターゲットマシン命令を作り出す方式を原則とする。
6. 変換後命令の最適化処理を行う。

## 2.2　8080 と 6800 の相違

大きな相違点をあげると次のものがある。

| 項　　目 | 8080 | 6800 |
|---|---|---|
| アドレス空間 | レジスタ・メモリ・I/O | レジスタ・メモリ |
| レジスタ数 | 7　（ACC　1） | 3　（ACC　2） |
| •演算フラグ | CZSPA | HINZVC |
| •開始アドレス | $(0000)_{16}$ 番地 | $(FFFE)_{16}$ 番地 |
| スタック | 2バイト単位 | 1バイト単位 |
| •アドレスデータ | LH格納 | HL格納 |
| アドレッシング | レジスタインダイレクト<br>（MOV　A, M） | インデックス<br>（LDA　8, X） |
| *割込 | ユーザにて<br>レジスタの退避回復 | ハードにて<br>レジスタの退避回復 |
| *I/O 命令 | I/O命令あり | メモリ　リード/ライト |

*印　　変換対象外としているところ

•印　　ロジック上注意を要するところ

（注1）　演算フラグのCr（キャリー）は必ず保証される。

（注2）　アドレスデータはターゲット CPUに合わせられる。

アセンブリ言語の違いには次のものがある。

| 項　　目 | 8080 表現 | 6800 表現 |
|---|---|---|
| 16進 データ | 0ABH | $AB |
| 10進 データ | 10<br>10D | 10 |
| 8進 データ | 70Q<br>70O | @70 |
| 2進 データ | 1010B | %1010 |
| 自 己 相 対 | $+6 | *+6 |
| ASCII データ | 'A' | 'A |
| データ エリア | DS 10 | RMB 10 |
| データ定数<br>（バイト単位） | DB 'TIME'<br>DB 0ABH | FCC /TIME/<br>FCC 4, TIME<br>FCB $AB |
| データ定数<br>（2バイト） | DW LABEL<br>DW 'AB'<br>（データはLH形式） | FDB LABEL<br>FDB $4142<br>（データはHL形式） |
| コメント文 | ;<br>（ラベル第1文字） | *<br>（ラベル第1文字） |
| ラ　ベ　ル | : で区切り | スペースで区切り |

## 2.3　変換における注意事項

ターゲット・プログラム（変換後プログラム）を完全にするためには、次のところを入力ソース上で訂正を要する場合がある。

(1)　8080←→6800変換の全般

- メモリ空間の配置

  ＲＡＭ／ＲＯＭの割当て方が異なるので、ＯＲＧ命令の訂正を行う。

- Ｉ／Ｏ命令の書き直し

  入出力処理は８０８０ではＩ／Ｏ命令を用い、６８００ではメモリー・リード／ライトを用いる。ハードの接続法とあわせて、この部分の書き直しを行う。

- ＲＥＧ命令の投入

  変換時のレジスタの対応づけにおけるレジスタ不足をメモリにて代替するため、ワークエリアの一部にＲＥＧ命令が必要となる。

8080 → 6800 変換時

入力REG

```
-          REG
+*** REG 8080 TO 6800 CONVERT
+*         A    --> A
+*         B,C  --> MEMORY B,C (NOT C,B)
+*         D,E  --> MEMORY D,E (NOT E,D)
+*         H,L  --> MEMORY H,L (NOT L,H)
+*         F    --> CCR
+*                   B,X WORK REG
+MB        RMB       1          * B REG
+MC        RMB       1          * C REG
+MD        RMB       1          * D REG
+ME        RMB       1          * E REG
+MH        RMB       1          * H REG
+ML        RMB       1          * L REG
+MSP       RMB       2          * SP
+**** ---
```

変換後

6800 → 8080 変換時

入力REG

```
-          REG
+;** REG 6800 TO 8080 CONVERT
+;         A    --> B
+;         B    --> C
+;         X    --> D,E
+;         CCR  --> F
+;                   A,HL WORK REG
+;***---
```

変換後

（注）　6800―8080変換時はレジスタをメモリにて代替はしていないので、必ずしもＲＥＧ命令の投入は必要ではない。

- 割込処理

  割込処理の機構が異なるので、その部分のロジック訂正を行う。

- リエントラントサブルーチン形式

  タスク制御時のリエントラントサブルーチンは変換後、その性質が保証されない場合がある。その時は書き直す必要がある。

- プログラムラベル

  判定分岐命令におけるラベル相対（ＬＡＢＥＬ±ｉ）、自己相対（$\overset{\$}{*}$±ｉ）の形式は、プログラムステップがかわることから、０で初まる数字４桁のシーケンス番号にて置きかえている。そのため、入力ソース上０で初まる数字４桁のプログラムラベルは二重定義になる恐れがあるので訂正する必要がある。

  (注) ラベル相対・自己相対処理では最大１０００ステートメント（コメントを除く）まで許される。

(2) 8080 → 6800変換

- パリティフラグとその判定命令

  6800 ターゲットプログラムの方で代替サブルーチンを作成する。

- アドレスデータの扱い

  8080 ＬＨは6800 ＨＬにかわるので、２バイトデータを、１バイト単位で取扱う時は注意を要する。

(3) 6800 → 8080変換

- オーバーフローフラグとその判定命令

  8080 ターゲット・プログラムの方で判定命令の変更を行う。

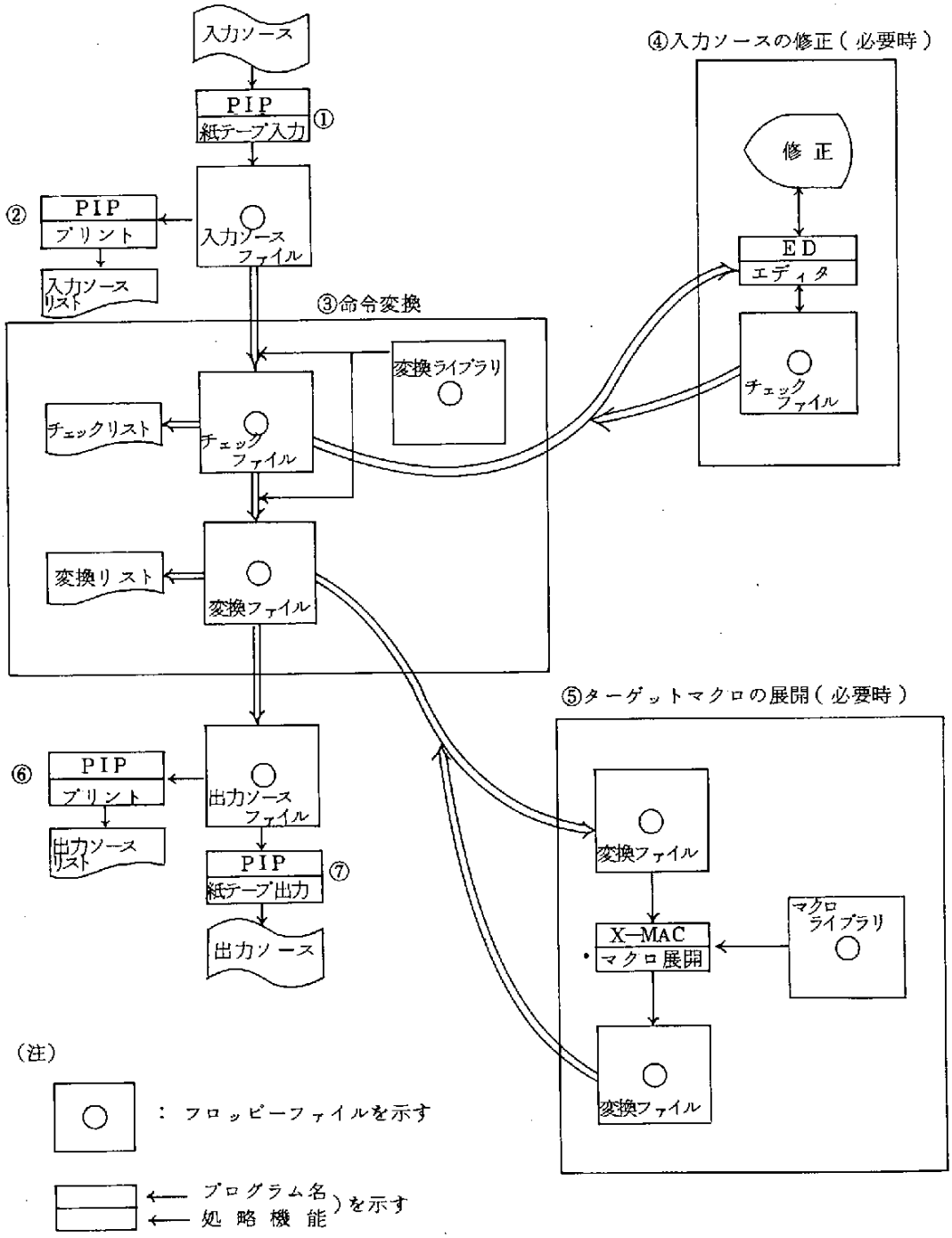
- アドレスデータの扱い

  6800 ＨＬは8080 ＬＨにかわるので、２バイトデータを１バイト単位で取扱う時は注意を要する。

  （注）詳細は、添付資料２、機能仕様書“言語変換編”参照のこと。

## 2.4 変 換 方 式

システム的には紙テープ入力、紙テープ出力となるが、本質的にはフロッピー・ディスク入力、フロッピー・ディスク出力になり、変換ライブラリを基本とした変換方式となる。以下に概略を示す。



① 入力ソースプログラム紙テープを、フロッピー・ディスクへ格納する。

② フロッピー・ディスク内の入力ソースプログラムをリスティングする。

③ 入力ソースと変換ライブラリと突き合せ、不良・不能命令の検出を行ない、各命令の演算フラグを表示したプログラムチェックリストを作成する。

その後、同じ変換ライブラリを用いて命令変換を行い、変換リストを作成し、ターゲット命令のみの出力ソースプログラムを作成する。

④ チェックリストを調査し、必要あれば入力ソースの修正を行う。

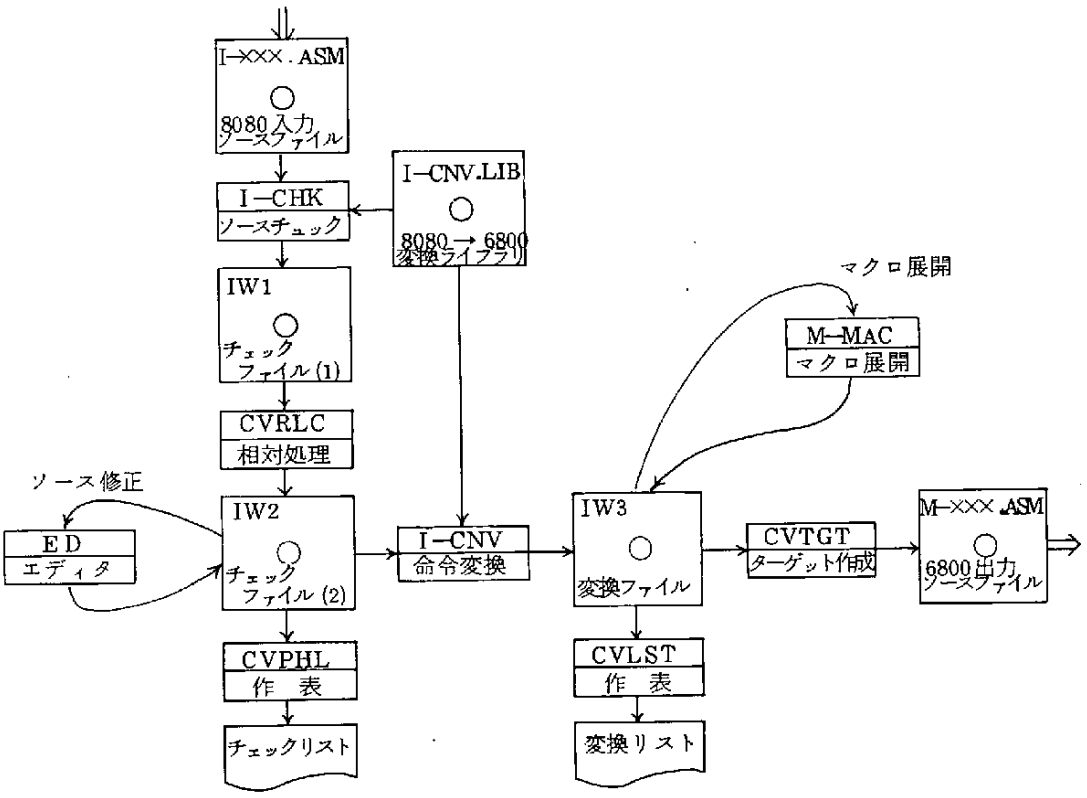
⑤ ターゲットマシンにて標準サブルーチン・マクロが用意されている時、ターゲットマクロの展開を行うことができる。

⑥ ターゲット命令の出力ソースのプログラムをリスティングする。

⑦ 出力ソースプログラムをフロッピー・ディスクより紙テープにパンチアウトする。

## 2.5　8080 ⟶ 6800 命令変換

入力ソースファイル（オリジナル・ソース）より出力ソース・ファイル（ターゲット・ファイル）を得るまでのファイルの流れ（システム・フロー）は次の様になる。




（注１）　ファイル名は、ＣＰ／Ｍ の書き方に従う。ただし、入力ソース・ファイル、チェック・ファイル、変換ファイルの頭１文字はＩであること。

（注２）　ソース修正は小さなサブルーチンの時にはほとんど必要ないが、複雑なプログラムには必要となる。

（注３）　変換対象として多数のプログラムを扱う時にはＭＯＶＥ、ＣＬＥＡＲ、Ｉ／Ｏ処理などはマクロとして登録し、その展開を行わせるとターゲットの効率が上る。

以下に変換の実例を示す。

2.5.1 入力ソース・ファイル・リスト ( 1/1 )

```
;*********************************************
;                                             *
;    8080 TO 6800 TEST    F=I-01.ASM          *
;         MOVE SUBROUTINE TEST                *
;                                             *
;*********************************************
;*
          ORG       1000H
;*
          LXI       SP,WSP+128
;***---
          LXI       D,T1
          LXI       H,W1
          MVI       A,16
          CALL      AMOVE             ; TITLE SET
;*
          LDA       WCNT              ; COUNTER
          ORI       30H
          STA       W1+6
;***---
          HLT
;*******
T1:       DB        'TOTAL X GROUPS'
          DB        0DH,0AH
;*
W1:       DS        30
WCNT:     DB        8
WSP:      DS        128
;*
;*********************************************
;*        DATA AREA MOVE (REENTRANT)
;*
;*                  DE=SND ADR
;*                  HL=RSV ADR
;*                  A=BYTE SIZE
;*        CALL      AMOVE    (A,BC SAVE)
;*                  DE=SND.END+1
;*                  HL=RSV.END+1
;*
AMOVE:    ORI       0
          JZ        AMOVE2
          PUSH      PSW
          PUSH      B
          MOV       B,A
;*
AMOVE1:   LDAX      D
          INX       D
          MOV       M,A
          INX       H
          DCR       B
          JNZ       AMOVE1
;*
          POP       B
          POP       PSW
AMOVE2:   RET
;******* ------
          END
```

2.5.2 チェック リスト ( 1／1 )

```
  ID:IW1          ***  8080 PROGRAM CHECK LIST  ***          56/ 2/20

SEQ  ER CZSPA%  LABEL   CODE    OPERAND          COMMENT

0001            ;*****************************************
0002            ;                                         *
0003            ;   8080 TO 6800 TEST   F=I-01.ASM        *
0004            ;       MOVE SUBROUTINE TEST              *
0005            ;                                         *
0006            ;*****************************************
0007            ;*
0008                    ORG     1000H
0009            ;*
0010    .....3          LXI     SP,WSP+128
0011            ;***---
0012    .....3          LXI     D,T1
0013    .....3          LXI     H,W1
0014    .....2          MVI     A,16
0015    .....3          CALL    AMOVE           ; TITLE SET
0016            ;*
0017    .....3          LDA     WCNT            ; COUNTER
0018    RVVVR2          ORI     30H
0019    .....3          STA     W1+6
0020            ;***---
0021    .....1          HLT
0022            ;*******
0023            T1:     DB      'TOTAL X GROUPS'
0024                    DB      0DH,0AH
0025            ;*
0026            W1:     DS      30
0027            WCNT:   DB      8
0028            WSP:    DS      128
0029            ;*
0030            ;*****************************************
0031            ;*      DATA AREA MOVE (REENTRANT)
0032            ;*
0033            ;*              DE=SND ADR
0034            ;*              HL=RSV ADR
0035            ;*              A=BYTE SIZE
0036            ;*      CALL    AMOVE   (A,BC SAVE)
0037            ;*              DE=SND.END+1
0038            ;*              HL=RSV.END+1
0039            ;*
0040    RVVVR2  AMOVE:  ORI     0
0041    .U...3          JZ      AMOVE2
0042    .....1          PUSH    PSW
0043    .....1          PUSH    B
0044    .....1          MOV     B,A
0045            ;*
0046    .....1  AMOVE1: LDAX    D
0047    .....1          INX     D
0048    .....1          MOV     M,A
0049    .....1          INX     H
0050    .VVVV1          DCR     B
0051    .U...3          JNZ     AMOVE1
0052            ;*
0053    .....1          POP     B
0054    VVVVV1          POP     PSW
0055    .....1  AMOVE2: RET
0056            ;******* ------
0057                    REG
0058                    END
```

2.5.3 変換リスト ( 1/3 )

```
  ID:IW2        ***  8080-6800 PROGRAM CONVERT LIST  ***   56/ 2/20

SEQ SEQ CZSPA%  LABEL   CODE    OPERAND          COMMENT

00010001        ********************************************
00020002        *                                          *
00030003        *   8080 TO 6800 TEST   F=I-01.ASM         *
00040004        *       MOVE SUBROUTINE TEST               *
00050005        *                                          *
00060006        ********************************************
00070007        **
0008           -        ORG     1000H
    0008       +        ORG     $1000

00090009        **
0010    .....3 -        LXI     SP,WSP+128
    0010       +        LDX     #WSP+128
    0011       +        TXS

00110012        ****---
0012    .....3 -        LXI     D,T1
    0013       +        LDX     #T1
    0014       +        STX     MD

0013    .....3 -        LXI     H,W1
    0015       +        LDX     #W1
    0016       +        STX     MH

0014    .....2 -        MVI     A,16
    0017       +        LDAA    #16

0015    .....3 -        CALL    AMOVE           ; TITLE SET
    0018       +        JSR     AMOVE   * TITLE SET

00160019        **
0017    .....3 -        LDA     WCNT            ; COUNTER
    0020       +        LDAA    WCNT    * COUNTER

0018    RVVVR2 -        ORI     30H
    0021       +        CLC
    0022       +        ORAA    #$30

0019    .....3 -        STA     W1+6
    0023       +        STAA    W1+6

00200024        ****---
0021    .....1 -        HLT
    0025       +        WAI

00220026        ********
0023           -T1:     DB      'TOTAL X GROUPS'
    0027       +T1      FCC     /TOTAL X GROUPS/

0024           -        DB      0DH,0AH
    0028       +        FCB     $D,$A

00250029        **
0026           -W1:     DS      30
    0030       +W1      RMB     30
```

```
  ID:IW2          ***  8080-6800 PROGRAM CONVERT LIST  ***    56/ 2/20
SEQ SEQ CZSPA%  LABEL    CODE     OPERAND          COMMENT


0027            -WCNT:   DB       8
    0031        +WCNT    FCB      8

0028            -WSP:    DS       128
    0032        +WSP     RMB      128

00290033         **
00300034         ********************************************
00310035         **       DATA AREA MOVE (REENTRANT)
00320036         **
00330037         **                DE=SND ADR
00340038         **                HL=RSV ADR
00350039         **                A=BYTE SIZE
00360040         **       CALL     AMOVE    (A,BC SAVE)
00370041         **                DE=SND.END+1
00380042         **                HL=RSV.END+1
00390043         **
0040    RVVVR2  -AMOVE:   ORI      0
    0044        +AMOVE    CLC
    0045        +         ORAA     #0

)041    .U...3  -         JZ       AMOVE2
    0046        +         BNE      *+5
    0047        +         JMP      AMOVE2

0042    .....1  -         PUSH     PSW
    0048        +         TAB
    0049        +         PSHA
    0050        +         TPA
    0051        +         PSHA
    0052        +         TBA

0043    .....1  -         PUSH     B
    0053        +         LDAB     MB
    0054        +         PSHB
    0055        +         LDAB     MB+1
    0056        +         PSHB

0044    .....1  -         MOV      B,A
    0057        +         STAA     MB

00450058         **
0046    .....1  -AMOVE1:  LDAX     D
    0059        +AMOVE1   LDX      MD.
    0060        +         LDAA     0,X

0047    .....1  -         INX      D
    0061        +         INX
    0062        +         STX      MD

0048    .....1  -         MOV      M,A
    0063        +         LDX      MH
    0064        +         STAA     0,X
```

```
  ID:IW2        ***  8080-6800 PROGRAM CONVERT LIST  ***   56/ 2/20

SEQ SEQ CZSPA%  LABEL   CODE    OPERAND          COMMENT

0049     .....1 -        INX     H
    0065        +        INX
    0066        +        STX     MH

0050     .VVVV1 -        DCR     B
    0067        +        DEC     MB

0051     .U...3 -        JNZ     AMOVE1
    0068        +        BEQ     *+5
    0069        +        JMP     AMOVE1

00520070         **
0053     .....1 -        POP     B
    0071        +        PULB
    0072        +        STAB    MB+1
    0073        +        PULB
    0074        +        STAB    MB

0054    VVVVV1 -        POP     PSW
    0075        +        PULA
    0076        +        TAP
    0077        +        PULA

0055     .....1 -AMOVE2: RET
    0078        +AMOVE2  RTS

00560079         ******* ------
0057             -       REG
    0080        +*** REG 8080 TO 6800 CONVERT
    0081        +*       A    --> A
    0082        +*       B,C  --> MEMORY B,C (NOT C,B)
    0083        +*       D,E  --> MEMORY D,E (NOT E,D)
    0084        +*       H,L  --> MEMORY H,L (NOT L,H)
    0085        +*       F    --> CCR
    0086        +*                B,X WORK REG
    0087        +MB      RMB     1        * B REG
    0088        +MC      RMB     1        * C REG
    0089        +MD      RMB     1        * D REG
    0090        +ME      RMB     1        * E REG
    0091        +MH      RMB     1        * H REG
    0092        +ML      RMB     1        * L REG
    0093        +MSP     RMB     2        * SP
    0094        +*** ---

00580095                 END
```

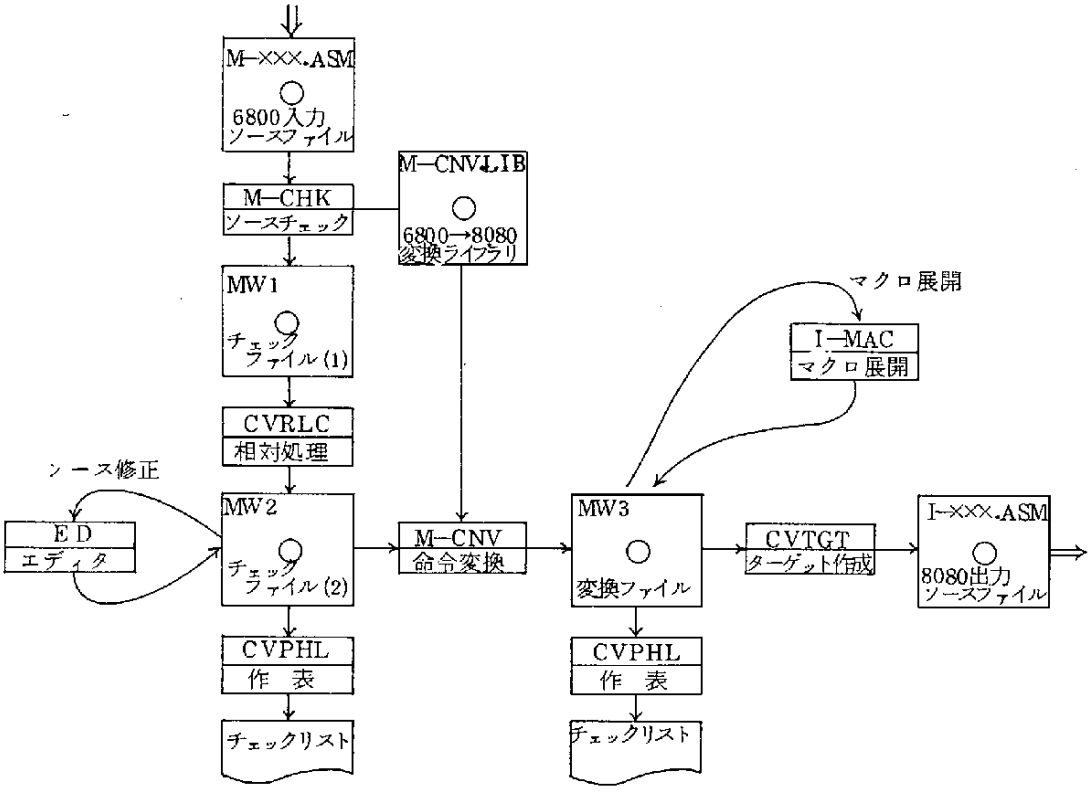
2.5.4 出力ソース・ファイル・リスト ( 1/2 )

```
 1: ********************************************
 2: *                                          *
 3: *   8080 TO 6800 TEST   F=I-01.ASM         *
 4: *       MOVE SUBROUTINE TEST               *
 5: *                                          *
 6: ********************************************
 7: **
 8:          ORG     $1000
 9: **
10:          LDX     #WSP+128
11:          TXS
12: ******---
13:          LDX     #T1
14:          STX     MD
15:          LDX     #W1
16:          STX     MH
17:          LDAA    #16
18:          JSR     AMOVE    * TITLE SET
19: **
20:          LDAA    WCNT     * COUNTER
21:          CLC
22:          ORAA    #$30
23:          STAA    W1+6
24: ******---
25:          WAI
26: ********
27: T1       FCC     /TOTAL X GROUPS/
28:          FCB     $D,$A
29: **
30: W1       RMB     30
31: WCNT     FCB     8
32: WSP      RMB     128
33: **
34: ********************************************
35: **       DATA AREA MOVE (REENTRANT)
36: **
37: **                DE=SND ADR
38: **                HL=RSV ADR
39: **                A=BYTE SIZE
40: **       CALL     AMOVE    (A,BC SAVE)
41: **                DE=SND.END+1
42: **                HL=RSV.END+1
43: **
44: AMOVE    CLC
45:          ORAA    #0
46:          BNE     *+5
47:          JMP     AMOVE2
48:          TAB
49:          PSHA
50:          TPA
51:          PSHA
52:          TBA
53:          LDAB    MB
54:          PSHB
55:          LDAB    MB+1
56:          PSHB
57:          STAA    MB
58: **
59: AMOVE1   LDX     MD
60:          LDAA    0,X
```

```
61:             INX
62:             STX     MD
63:             LDX     MH
64:             STAA    0,X
65:             INX
66:             STX     MH
67:             DEC     MB
68:             BEQ     *+5
69:             JMP     AMOVE1
70: **
71:             PULB
72:             STAB    MB+1
73:             PULB
74:             STAB    MB
75:             PULA
76:             TAP
77:             PULA
78: AMOVE2      RTS
79: ******* ------
80: *** REG 8080 TO 6800 CONVERT
81: *           A   --> A
82: *           B,C --> MEMORY B,C (NOT C,B)
83: *           D,E --> MEMORY D,E (NOT E,D)
84: *           H,L --> MEMORY H,L (NOT L,H)
85: *           F   --> CCR
86: *                   B,X WORK REG
87: MB          RMB     1       * B REG
88: MC          RMB     1       * C REG
89: MD          RMB     1       * D REG
90: ME          RMB     1       * E REG
91: MH          RMB     1       * H REG
92: ML          RMB     1       * L REG
93: MSP         RMB     2       * SP
94: *** ---
95:             END
```

## 2.6　6800 ― 8080 命令変換

入力ソース・ファイル（オリジナル・ソース）より出力ソース・ファイル（ターゲットファイル）を得るまでのファイルの流れ（システム・フロー）は次の様になる。




（注１）　ファイル名はＣＰ／Ｍの書き方に従う。ただし、入力ソース・ファイル、チェック・ファイル、変換ファイルの頭１文字はＭであること。

（注２）　ソース修正は小さなサブルーチンの時にはほとんど必要ないが、複雑なプログラムには必要となる。

（注３）　変換対象として多数のプログラムを扱う時にはＭＯＶＥ、ＣＬＥＡＲ、Ｉ／Ｏ処理などはマクロとして登録し、その展開を行わせるとターゲットの効率が上る。

以下に変換の実例を示す。

2.6.1 入力ソース・ファイル・リスト ( 1/1 )

```
 1: ************************************************
 2: *                                              *
 3: *   6800 TO 8080 TEST   F=M-01.ASM             *
 4: *       MOVE SUBROUTINE TEST                   *
 5: *                                              *
 6: ************************************************
 7: *
 8:         ORG     $1000
 9: *
10:         LDS     #WSP+127
11: ****----
12:         LDX     #T1
13:         STX     BWRK1
14:         LDX     #W1
15:         STX     BWRK2
16:         LDAA    #16
17:         JSR     BMOVE           * TITLE SET
18: *
19:         LDAA    WCNT            * COUNTER
20:         ORAA    #$30
21:         STAA    W1+6
22: ****----
23:         WAI
24: *******
25: T1      FCC     /TOTAL X GROUPS/
26:         FCB     $0D,$0A
27: *
28: W1      RMB     30
29: WCNT    FCB     8
30: WSP     RMB     128
31: *
32: ******  B ROUTINE WORK AREA *******
33: BWRK1   RMB     2
34: BWRK2   RMB     2
35: BWRK3   RMB     2
36: ******  ------
37: ********************************************
38: *       DATA AREA MOVE
39: *
40: *               BWRK1=SND ADR
41: *               BWRK2=RSV ADR
42: *               A=BYTE SIZE
43: *       JSR     BMOVE   (A,B SAVE)
44: *               BWRK1=SND.END+1
45: *               BWRK2=RSV.END+1
46: *
47: BMOVE   TSTA
48:         BEQ     BMOVE2
49:         PSHA
50:         PSHB
51: *
52: BMOVE1  LDX     BWRK1
53:         LDAB    0,X
54:         INX
55:         STX     BWRK1
56:         LDX     BWRK2
57:         STAB    0,X
58:         INX
59:         STX     BWRK2
60:         DECA
61:         BNE     BMOVE1
62: *
63:         PULB
64:         PULA
65: BMOVE2  RTS
66: ******  -------
67:         END
```

```
  ID:MW1          ***  6800 PROGRAM CHECK LIST  ***          56/ 2/20

SEQ  ER HNZVC%  LABEL   CODE    OPERAND          COMMENT

0001            ************************************************
0002            *                                              *
0003            *   6800 TO 8080 TEST   F=M-01.ASM             *
0004            *       MOVE SUBROUTINE TEST                   *
0005            *                                              *
0006            ************************************************
0007            *
0008                    ORG     $1000
0009            *
0010    .VVR.3          LDS     #WSP+127
0011            ********--------
0012    .VVR.3          LDX     #T1
0013    .VVR.3          STX     BWRK1
0014    .VVR.3          LDX     #W1
0015    .VVR.3          STX     BWRK2
0016    .VVR.2          LDAA    #16
0017    .....3          JSR     BMOVE           * TITLE SET
0018            *
0019    .VVR.3          LDAA    WCNT            * COUNTER
0020    .VVR.3          ORAA    #$30
0021    .VVR.3          STAA    W1+6
0022            ********--------
0023    .....1          WAI
0024            *******
0025            T1      FCC     /TOTAL X GROUPS/
0026                    FCB     $0D,$0A
0027            *
0028            W1      RMB     30
0029            WCNT    FCB     8
0030            WSP     RMB     128
0031            *
0032            ******  B ROUTINE WORK AREA *******
0033            BWRK1   RMB     2
0034            BWRK2   RMB     2
0035            BWRK3   RMB     2
0036            ******  ------
0037            ********************************************
0038            *       DATA AREA MOVE
0039            *
0040            *               BWRK1=SND ADR
0041            *               BWRK2=RSV ADR
0042            *               A=BYTE SIZE
0043            *       JSR     BMOVE   (A,B SAVE)
0044            *               BWRK1=SND.END+1
0045            *               BWRK2=RSV.END+1
0046            *
0047    .VVRR1  BMOVE   TSTA
0048    ..U..2          BEQ     BMOVE2
0049    .....1          PSHA
0050    .....1          PSHB
0051            *
0052    .VVR.3  BMOVE1  LDX     BWRK1
0053    .VVR.2          LDAB    0,X
0054    ..V..1          INX
0055    .VVR.3          STX     BWRK1
```

```
  ID:MW1          ***  6800 PROGRAM CHECK LIST  ***          56/ 2/20

SEQ   ER HNZVC%  LABEL    CODE     OPERAND          COMMENT

0056     .VVR.3           LDX      BWRK2
0057     .VVR.2           STAB     0,X
0058     ..V..1           INX
0059     .VVR.3           STX      BWRK2
0060     .VVV.1           DECA
0061     ..U..2           BNE      BMOVE1
0062              *
0063     .....1           PULB
0064     .....1           PULA
0065     .....1  BMOVE2   RTS
0066             ******   -------
0067                      REG
0068                      END
```

2.6.3 変換リスト ( 1/4 )

```
  ID:MW2         ***  6800-8080 PROGRAM CONVERT LIST  ***    56/ 2/20

SEQ SEQ HNZVC%  LABEL   CODE    OPERAND         COMMENT

00010001        ;*************************************************
00020002        ;                                                *
00030003        ;   6800 TO 8080 TEST   F=M-01.ASM               *
00040004        ;       MOVE SUBROUTINE TEST                     *
00050005        ;                                                *
00060006        ;*************************************************
00070007        ;
0008            -       ORG     $1000
    0008        +       ORG     1000H

00090009        ;
0010    .VVR.3  -       LDS     #WSP+127
    0010        +       LXI     SP,WSP+127

00110011        ;***----
0012    .VVR.3  -       LDX     #T1
    0012        +       LXI     D,T1

0013    .VVR.3  -       STX     BWRK1
    0013        +       XCHG
    0014        +       SHLD    BWRK1

0014    .VVR.3  -       LDX     #W1
    0015        +       LXI     D,W1

0015    .VVR.3  -       STX     BWRK2
    0016        +       XCHG
    0017        +       SHLD    BWRK2

0016    .VVR.2  -       LDAA    #16
    0018        +       MVI     B,16

0017    .....3  -       JSR     BMOVE           * TITLE SET
    0019        +       CALL    BMOVE   ; TITLE SET

00180020        ;
0019    .VVR.3  -       LDAA    WCNT            * COUNTER
    0021        +       LXI     H,WCNT  ; COUNTER
    0022        +       MOV     B,M

0020    .VVR.3  -       ORAA    #$30
    0023        +       PUSH    PSW
    0024        +       MVI     A,30H
    0025        +       ORA     B
    0026        +       MOV     B,A
    0027        +       XTHL
    0028        +       MOV     A,L
    0029        +       RRC
    0030        +       POP     H

0021    .VVR.3  -       STAA    W1+6
    0031        +       MOV     A,B
    0032        +       STA     W1+6

00220033        ;***----
```

```
  ID:MW2         ***  6800-8080 PROGRAM CONVERT LIST  ***    56/ 2/20

SEQ SEQ HNZVC%  LABEL    CODE     OPERAND          COMMENT

0023     .....1 -         WAI
    0034        +         HLT

00240035         ;******
0025            -T1       FCC      /TOTAL X GROUPS/
    0036        +T1:      DB       'TOTAL X GROUPS'

0026            -         FCB      $0D,$0A
    0037        +         DB       0DH,0AH

00270038         ;
0028            -W1       RMB      30
    0039        +W1:      DS       30

0029            -WCNT     FCB      8
    0040        +WCNT:    DB       8

0030            -WSP      RMB      128
    0041        +WSP:     DS       128

00310042         ;
00320043         ;*****  B ROUTINE WORK AREA *******
0033            -BWRK1    RMB      2
    0044        +BWRK1:   DS       2

0034            -BWRK2    RMB      2
    0045        +BWRK2:   DS       2

0035            -BWRK3    RMB      2
    0046        +BWRK3:   DS       2

00360047         ;*****  ------
00370048         ;****************************************
00380049         ;         DATA AREA MOVE
00390050         ;
00400051         ;                  BWRK1=SND ADR
00410052         ;                  BWRK2=RSV ADR
00420053         ;                  A=BYTE SIZE
00430054         ;         JSR      BMOVE    (A,B SAVE)
00440055         ;                  BWRK1=SND.END+1
00450056         ;                  BWRK2=RSV.END+1
00460057         ;
0047    .VVRR1  -BMOVE    TSTA
    0058        +BMOVE:   MOV      A,B
    0059        +         CPI      0

0048     ..U..2 -         BEQ      BMOVE2
    0060        +         JZ       BMOVE2

0049     .....1 -         PSHA
    0061        +         PUSH     B
    0062        +         INX      SP

0050     .....1 -         PSHB
    0063        +         MOV      A,C
```

```
  ID:MW2          ***  6800-8080 PROGRAM CONVERT LIST  ***    56/ 2/20

SEQ SEQ HNZVC%  LABEL    CODE     OPERAND          COMMENT

    0064        +        PUSH     PSW
    0065        +        INX      SP

00510066        ;
0052    .VVR.3 -BMOVE1   LDX      BWRK1
    0067       +BMOVE1:  LHLD     BWRK1
    0068        +        XCHG

0053    .VVR.2 -         LDAB     0,X
    0069        +        LXI      H,0
    0070        +        PUSH     PSW
    0071        +        DAD      D
    0072        +        POP      PSW
    0073        +        MOV      C,M

0054    ..V..1 -         INX
    0074        +        RAL
    0075        +        MOV      H,A
    0076        +        INX      D
    0077        +        MOV      A,D
    0078        +        ORA      E
    0079        +        MOV      A,H
    0080        +        RAR

0055    .VVR.3 -         STX      BWRK1
    0081        +        XCHG
    0082        +        SHLD     BWRK1

0056    .VVR.3 -         LDX      BWRK2
    0083        +        LHLD     BWRK2
    0084        +        XCHG

0057    .VVR.2 -         STAB     0,X
    0085        +        LXI      H,0
    0086        +        PUSH     PSW
    0087        +        DAD      D
    0088        +        POP      PSW
    0089        +        MOV      M,C

0058    ..V..1 -         INX
    0090        +        RAL
    0091        +        MOV      H,A
    0092        +        INX      D
    0093        +        MOV      A,D
    0094        +        ORA      E
    0095        +        MOV      A,H
    0096        +        RAR

0059    .VVR.3 -         STX      BWRK2
    0097        +        XCHG
    0098        +        SHLD     BWRK2

0060    .VVV.1 -         DECA
    0099        +        DCR      B
```

```
  ID:MW2         ***  6800-8080 PROGRAM CONVERT LIST  ***    56/ 2/20

SEQ SEQ HNZVC%  LABEL   CODE     OPERAND          COMMENT

0061    ..U..2 -        BNE      BMOVE1
    0100       +        JNZ      BMOVE1

00620101        ;
0063    .....1 -        PULB
    0102       +        DCX      SP
    0103       +        POP      H
    0104       +        MOV      C,H

0064    .....1 -        PULA
    0105       +        DCX      SP
    0106       +        POP      H
    0107       +        MOV      B,H

0065    .....1 -BMOVE2  RTS
    0108       +BMOVE2: RET

00660109        ;*****  -------
0067           -        REG
    0110       +;** REG 6800 TO 8080 CONVERT
    0111       +;       A    --> B
    0112       +;       B    --> C
    0113       +;       X    --> D,E
    0114       +;       CCR  --> F
    0115       +;                A,HL WORK REG
    0116       +;***---

00680117                END
```

2.6.4 出力ソース・リスト ( 1/2 )

```
 1: ;*************************************************
 2: ;                                                *
 3: ;   6800 TO 8080 TEST   F=M-01.ASM               *
 4: ;       MOVE SUBROUTINE TEST                     *
 5: ;                                                *
 6: ;*************************************************
 7: ;
 8:         ORG     1000H
 9: ;
10:         LXI     SP,WSP+127
11: ;***----
12:         LXI     D,T1
13:         XCHG
14:         SHLD    BWRK1
15:         LXI     D,W1
16:         XCHG
17:         SHLD    BWRK2
18:         MVI     B,16
19:         CALL    BMOVE   ; TITLE SET
20: ;
21:         LXI     H,WCNT  ; COUNTER
22:         MOV     B,M
23:         PUSH    PSW
24:         MVI     A,30H
25:         ORA     B
26:         MOV     B,A
27:         XTHL
28:         MOV     A,L
29:         RRC
30:         POP     H
31:         MOV     A,B
32:         STA     W1+6
33: ;***----
34:         HLT
35: ;******
36: T1:     DB      'TOTAL X GROUPS'
37:         DB      0DH,0AH
38: ;
39: W1:     DS      30
40: WCNT:   DB      8
41: WSP:    DS      128
42: ;
43: ;*****  B ROUTINE WORK AREA *******
44: BWRK1:  DS      2
45: BWRK2:  DS      2
46: BWRK3:  DS      2
47: ;*****  ------
48: ;*******************************************
49: ;       DATA AREA MOVE
50: ;
51: ;               BWRK1=SND ADR
52: ;               BWRK2=RSV ADR
53: ;               A=BYTE SIZE
54: ;       JSR     BMOVE   (A,B SAVE)
55: ;               BWRK1=SND.END+1
56: ;               BWRK2=RSV.END+1
57: ;
58: BMOVE:  MOV     A,B
59:         CPI     0
60:         JZ      BMOVE2
```

```
61:          PUSH    B
62:          INX     SP
63:          MOV     A,C
64:          PUSH    PSW
65:          INX     SP
66: ;
67: BMOVE1:  LHLD    BWRK1
68:          XCHG
69:          LXI     H,0
70:          PUSH    PSW
71:          DAD     D
72:          POP     PSW
73:          MOV     C,M
74:          RAL
75:          MOV     H,A
76:          INX     D
77:          MOV     A,D
78:          ORA     E
79:          MOV     A,H
80:          RAR
81:          XCHG
82:          SHLD    BWRK1
83:          LHLD    BWRK2
84:          XCHG
85:          LXI     H,0
86:          PUSH    PSW
87:          DAD     D
88:          POP     PSW
89:          MOV     M,C
90:          RAL
91:          MOV     H,A
92:          INX     D
93:          MOV     A,D
94:          ORA     E
95:          MOV     A,H
96:          RAR
97:          XCHG
98:          SHLD    BWRK2
99:          DCR     B
100:         JNZ     BMOVE1
101: ;
102:         DCX     SP
103:         POP     H
104:         MOV     C,H
105:         DCX     SP
106:         POP     H
107:         MOV     B,H
108: BMOVE2: RET
109: ;*****  -------
110: ;** REG 6800 TO 8080 CONVERT
111: ;       A    --> B
112: ;       B    --> C
113: ;       X    --> D,E
114: ;       CCR  --> F
115: ;                A,HL WORK REG
116: ;***---
117:         END
```

# 3　システム開始・終了の操作

## 3.1　電源投入前の確認

(1)　ＰＴＰ（紙テープパンチャ）

ＰＴＰに紙テープをセットする。

(2)　ＳＬＰ（シリアルプリンタ）

ＳＬＰに９インチ巾の用紙をセットする。

(3)　ＦＤ（フロッピーディスク）

ＦＤ－１（Ａ面）にＣＰ／Ｍシステム・ディスクを挿入する。

ＦＤ－２（Ｂ面）にコンバータ・システム・ディスクを挿入する。

## 3.2　電源投入法とデバイス確認

(1)　ＣＲＴのＯＮ、ＯＦＦ　ＳＷ、ＳＬＰのＯＮ、ＯＦＦ　ＳＷを［ＯＮ］にする。

（この時点でのランプの点灯はなく、システム電源［ＯＮ］まで待たされる）

(2)　パネルシステム電源の［ＯＮ］ＳＷを押す。ＦＤ－１のランプ点灯を確認して、ＣＲＴに以下のメッセージ出力を確認する。

```
＊＊　　ＳＵＧＩ　　ＥＬＥＣＴＲＩＣ　　ＣＯ．ＬＴＤ．　　＊＊
＊＊　　　　　　ＳＭＣ－２　　ＳＹＳＴＥＭ　　＊＊
Ａ＞
```

また、ＳＬＰのＰＯＷＥＲ ランプ点灯及び、ＰＴＲのランプ点灯を確認する。ＳＬＰのＳＥＬランプ点灯していない時は［ＳＥＬ］ＳＷを押し、ランプ点灯を確認する。

(3)　紙テープ読込みが必要な時はＰＴＲに、入力紙テープをセットする。

(4)　ＳＬＰの用紙のライン・フィード必要時は［ＳＥＬ］ＳＷ押し、［ＬＦ］ＳＷを押すことにより、ライン・フィードする。同様にトップフォーム必要時は［ＴＯＦ］ＳＷを押すことにより、トップフォームする。そして、再度［ＳＥＬ］ＳＷ押し、ＳＥＬランプ点灯を確認する。

(5)　再度リセットして行ないたい時はパネルの［ＲＥＳＥＴ］　ＳＷを押す。その時もＣＲＴに以下のメッセージが出力される。

```
＊＊　　ＳＵＧＩ　　ＥＬＥＣＴＲＩＣ　　ＣＯ．ＬＴＤ．　　＊＊
＊＊　　　　　　ＳＭＣ－２　　ＳＹＳＴＥＭ　　＊＊
Ａ＞
```

## 3.3 フロッピーメディアの確認

(1) ディレクトリー確認を以下の様に行う。

A＞ DIR A: (Cr) ← A面のディレクトリー確認時

or

DIR B: (Cr) ← B面 〃

上記の操作により、ファイル名がＣＲＴに出力される

(2) 空エリア調査時は以下の様に行う。

A＞ STAT A: (Cr) ← A面の空エリア調査時

or

STAT B: (Cr) ← B面 〃

上記の操作により、空エリアサイズがＣＲＴに出力される。

詳細はＣＰ／Ｍマニュアル参照のこと。

## 3.4 終了操作

フロッピーディスクとのＩ／Ｏ処理が動いていないことを確認して、パネルシステム電源のOFF SWを押すことによって電源を切る。

そして、フロッピーディスクを抜き取り保管する。

# 4. コンバージョン操作

## 4.1 共通操作事項

各プログラムにて共通となっている操作を説明する。

(1) プログラムのローディング

B＞[MPL] (Cr)

／MPL PROGRAM＝ [プログラム名]

ここで、プログラム名はコンバージョン各操作におけるプログラム名である。

本システムにおいてはマクロプロセッサ用の言語を用いて開発され、オブジェクト効率、ファイル効率を考えて、コンパイル&ゴー形式となっているため、マクロプロセッサ・システムを呼出し、次にプログラム名を与えて初めて目的とするプログラムをメモリーに呼出せる様になっている。

（注１） MPLシステム、コンバータプログラム、命令変換ライブラリ、マクロライブラリは同じユニット上で操作する。

（注２） キーインミスの時は [DEL] を押すことにより、前に投入された文字が画面表示され、消されたことになる。

（注３） B＞[MPL] (Cr) の様にコントロール [P] を投入するとそのＣＲＴ表示分がすべてプリンタに出され、ハードコピーが取れることになる。

└ コントロール [P]

(2) ファイル名称の投入

例の様に入出力ファイルを必要とする時はメッセージが出るのでユニット名、ファイル名を入力する。

／INPUT FILE＝ [B：×××．ASM] (Cr)

ユニット名（A，Bのみ）
ユニット名省略時は現在の選択Diskとなる。

ファイル主要名
（最大８文字まで可）

ファイル従属名
（最大３文字まで可、省略可）

（注１） 不良の時は再度メッセージが印字される。

（注２） [DEL] コードは有効である。

(3) 日付の投入

日付を必要とする時は以下のメッセージが出力されるので、８文字の文字列で年月日を入力する。

／DATE：YY／MM／DD＝[56／△2／20] (Cr)

（注１） ８文字以外の時は再度メッセージが出力されるので、再度入力する。

(4) ファイルエラーに関して

ファイルエラー発生時は以下のエラーメッセージが出力される。

／FILE ERR ××
　　└エラーコード

| 項　　目 | エラーコード | 原　因、対　策 |
|---|---|---|
| オープン時 | O1 | モード設定（入力 or 出力）エラー |
| | O2 | ユニット名、FILE名が未定義 |
| | O3 | モード入力用でFILEが存在しない |
| | O4 | モード出力用でFILEが存在する。<br>続いて以下のメッセージが出力される。<br>IN＝[X]← 0 or 0以外　入力<br>{ 0入力時　FILEをデリート後出力用に使用する。<br>{ 0以外入力時　CP／Mへリターンする。 |
| | O5 | モード出力用でFILEの作成不可 |
| クローズ時 | C1 | クローズ不可 |
| 読　込　時 | R1 | 読込不可 |
| | R2 | 1行読込時128バイトに(Cr) なし |
| 書　込　時 | W1 | 書込不可 |
| | W2 | 書込時拡張FILEエラー |
| | W3 | Diskデータエリア満杯 |
| | W4 | ディレクトリーエリア満杯 |
| | W5 | 1行書込時128バイトに(Cr) なし |

エラー発生後 O4　IN＝ [0] 入力以外はすべてCP／Mへリターンするので、再度(1)のプログラムのローディングから始める。

(5) プログラム入力ユニットの変更

B面を選択Diskとして使用する時

A＞[B：](Cr)

と入力すると

B＞

と出力され、B面が選択されたことになる。

同様にA面を選択Diskとして使用する時

B＞[A：](Cr)

と入力すると

A＞

と出力され、A面が選択されたことになる。

(6) プログラム・コピー

以下の方法によりプログラム・コピーを行う。

A＞PIP [A：×××.×××＝B：XXX.×××]

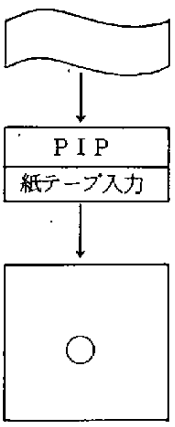
B面の×××.×××ファイルが、A面に×××.×××ファイルとしてコピーされる。

A＞PIP [B：＝A：＊.＊]

A面のすべてのファイルがB面にコピーされる。

## 4.2　入力ソース紙テープ（オリジナル紙テープ）の読み込み

変換しようとする8080アセンブリ言語プログラム又は、6800アセンブリ言語プログラムの紙テープをフロッピーディスクに入れる。




8080 オリジナルテープ
or
6800　　〃

I－×××. ASM
or
M－×××. ASM

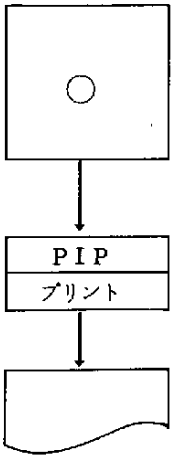
（8080→6800 の時）

A> PIP B:I－×××. ASM＝RDR: (Cr)
or
A> PIP B:M－×××. ASM＝RDR: (Cr)

（6800→8080 の時）

## 4.3　入力ソースファイルのプリント

これから変換をしようとする入力ファイルのリストを取る。




I－×××. ASM
or
M－×××. ASM

入力ソースリスト

（8080→6800 の時）

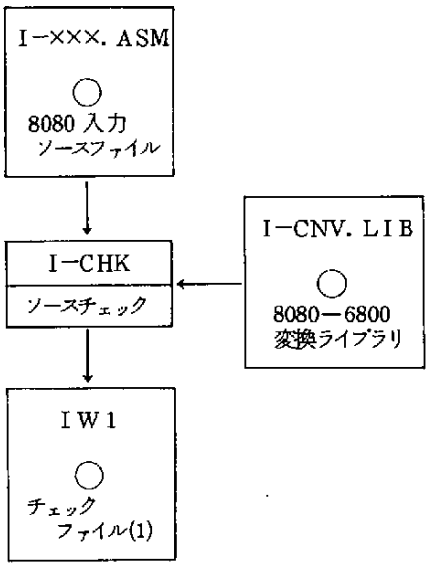
A> PIP PRN:＝B:I－×××. ASM (Cr)
or
A> PIP PRN:＝B:M－×××. ASM (Cr)

（6800→8080 の時）

## 4.4　入力ソースの命令チェック

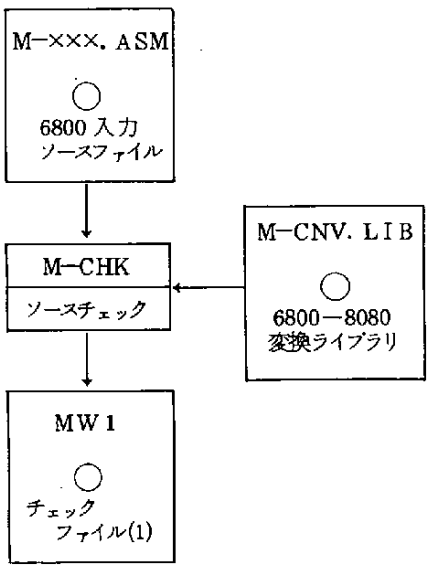
入力ソースの各命令について変換ライブラリを使用し、ソースマシンのＣＰＵフラグとそのバイト数を取り出す。

又変換不能な命令を検出する。




(1)　8080 → 6800 の時

```
B>MPL (Cr)
/MPL PROGRAM  = I−CHK (Cr)
/INPUT FILE   = I−×××. ASM (Cr)
/OUTPUT FILE  = IW1 (Cr)
```

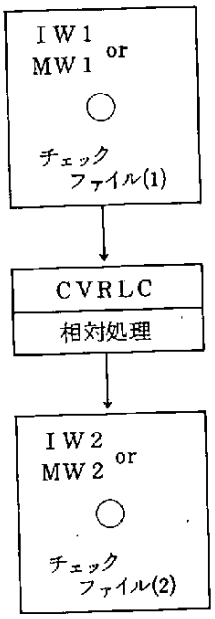



(2)　6800 → 8080 の時

```
B>MPL (Cr)
/MPL PROGRAM  = M−CHK (Cr)
/INPUT FILE   = M−×××. ASM (Cr)
/OUTPUT FILE  = MW1 (Cr)
```

## 4.5　相対アドレス処理

ブランチ命令におけるラベル相対（ＬＡＢＥＬ±ｉ）、自己相対（＊±ｉ，＄±ｉ）をターゲットマシンに合わせるべく飛び先命令にラベル付を行う。




(1)　8080 → 6800 の時

```
B> MPL (Cr)
/MPL  PROGRAM  = CVRLC (Cr)
/INPUT  FILE   = IW1 (Cr)
/OUTPUT FILE   = IW2 (Cr)
```

(2)　6800 → 8080 の時

```
B> MPL (Cr)
/MPL  PROGRAM  = CVRLC (Cr)
/INPUT  FILE   = MW2 (Cr)
/OUTPUT FILE   = MW2 (Cr)
```

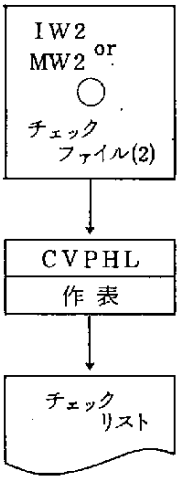
（注１）　本操作は、ブランチ命令に相対形式のないときは省略できる。

（注２）　本処理では1000ステートメント（コメントを除く）までのプログラムを扱うことができる。

1000ステートメントをオーバーすると、ＴＡＢＬＥ　ＯＶＥＲ　ＥＲＲＯＲ　のメッセージを出力して処理を終了する。

## 4.6　チェックリストの作成

前記入力ソース命令チェック、相対アドレス処理されたチェック・ファイルのリスティングを行う。




(1)　8080 → 6800 の時

```
B>MPL (Cr)
/MPL PROGRAM =CVPHL (Cr)
/DATE:YY/MM/DD=56/△2/20 (Cr)
/INPUT FILE =IW2 (Cr)
```

(2)　6800 → 8080 の時

```
B>MPL (Cr)
/MPL PROGRAM =CVPHL (Cr)
/DATE:YY/MM/DD=56/△2/20 (Cr)
/INPUT FILE =MW2 (Cr)
```

8080 → 6800 時フォーマット

```
   ID:×××××××   *** 8080 PROGRAM CHECK LIST ***   YY/MM/DD   PAGE=×××
SEQ  ER CZSPA%    LABEL   CODE     OPERAND       COMMENT
```

6800 → 8080 時フォーマット

```
   ID:×××××××   *** 6800 PROGRAM CHECK LIST ***   YY/MM/DD   PAGE=×××
SEQ  ER HNZVC%    LABEL   CODE     OPERAND       COMMENT
```

ＩＤ　：ファイル名
ＳＥＱ：シーケンス番号
ＥＲ　：エラー表示

- ＮＣ　変換不能コード
- ＥＲ　オペランド不正
- ＡＥ　アドレスエラー

ＣＺＳＰＡ：8080フラグ
ＨＮＺＶＣ：6800フラグ

- •　変化なし
- Ｖ　変化する
- Ｒ　リセットされる（０になる）
- Ｓ　セットされる（１になる）
- Ｕ　ブランチ命令にて参照される
- ％　バイト数を示す

- エラー表示のあるところを次のエディターにて修正する。
- ブランチ命令のUフラグをもとに、本当に変換可能かどうか検討する。

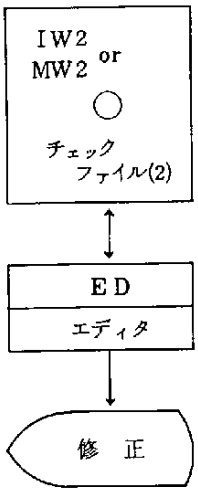
  8080のP(パリティフラグ)は保証されていない。

  6800のV(オーバーフローフラグ)は保証されていない。

  キャリーを除くフラグは完全には保証されていない。

## 4.7 チェックファイルの修正

エラー表示のステートメントを削除して、ターゲットのマシン命令にて置換える時、ターゲットの標準サブルーチンが使用可能な時等はCP/MのED(テキスト・エディター)を用いてチェック・ファイルの修正を行う。




(8080 → 6800 の時)

A> ED B:IW2 (Cr)

or

A> ED B:MW2 (Cr)

(6800 → 8080 の時)

修正の詳細はCP/M マニュアル参照のこと。

主な機能として次のコマンドがある。

I : 追加　　L, T : 表示

K : 削除

F : 探索

S : 置換

(1) REG命令の投入

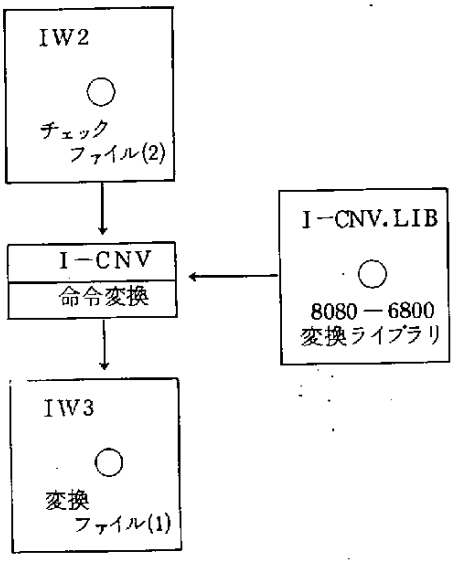
(ENDの前に)必ず挿入すること。

(2) オリジナル命令とターゲット命令が同じ命令(DAA命令)については16バイト目に[+]を投入すること。

その他は (TAB)(TAB) LABEL (TAB) CODE (TAB) OPERAND (TAB) COMMENT (Cr) の様に投入する。

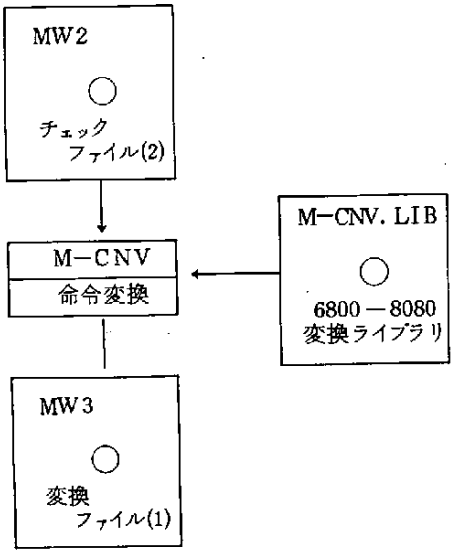
## 4.8 命令変換

チェック・ファイルを入力として変換ライブラリを参照して、ターゲットマシン命令への命令変換を行う。この操作の出力ファイルとして入力命令には [−]、出力命令には [+] が、16バイト目に付けられる。




(1) 8080 → 6800 の時

```
B>[MPL] (Cr)
/MPL PROGRAM = [I−CNV] (Cr)
/INPUT FILE  = [IW2] (Cr)
/OUTPUT FILE = [IW3] (Cr)
```

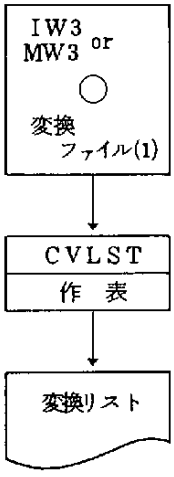



(2) 6800 → 6800 の時

```
B>[MPL] (Cr)
/MPL PROGRAM = [M−CNV] (Cr)
/INPUT FILE  = [MW2] (Cr)
/OUTPUT FILE = [MW3] (Cr)
```

## 4.9　変換リストの形成

前記命令変換を行った変換ファイルのリスティングを行う。




```
B>MPL (Cr)
/MPL PROGRAM   = CVLST (Cr)
/DATE:YY/MM/DD = 56/△2/20 (Cr)
/INPUT FILE    = IW3 (Cr)
                 (8080 → 6800 の時)
                 or
                 MW3 (Cr)
                 (6800 → 8080 の時)
```

8080 → 6800 時フォーマット

```
   ID:×××××××   *** 8080-6800 PROGRAM CONVERT LIST *** YY/MM/DD PAGE=×××

SEQ SEQ   CZSPA%   LABEL  CODE      OPERAND    COMMENT
```

6800 → 8080 時フォーマット

```
   ID:×××××××   *** 6800-8080 PROGRAM CONVERT LIST *** YY/MM/DD PAGE=×××

SEQ SEQ   HNZVC%   LABEL  CODE      OPERAND    COMMENT
```

左ＳＥＱ：入力ソースのシーケンスNO

右ＳＥＱ：出力　　〃

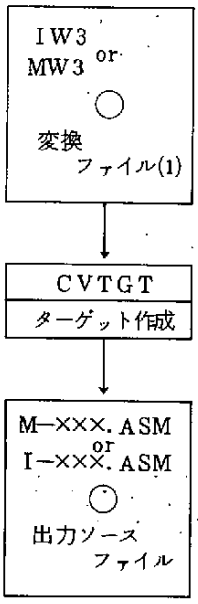
ラベルの前の特殊文字は変換区分を示し、次の意味をもつ

- －　：ソース命令であり不要となるもの
- ＋　：ターゲット命令であり展開されたもの
- (sp)：コメントはマクロ展開不能であったもの

ここで、エラー表示がまだあれば、4.7チェック・ファイルの修正に戻って再度この流れを行う。

## 4.10　ターゲットソースファイルの作成

変換後のターゲットファイルを作成する。変換ファイルの１６バイト目が＋、(sp) である命令に関してのみ取出す。




(1)　8080 → 6800 の時

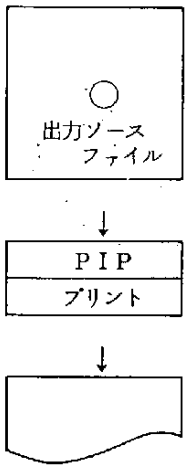
```
B>MPL (Cr)
/MPL PROGRAM = CVTGT (Cr)
/INPUT FILE  = IW3 (Cr)
/OUTPUT FILE = M−×××. ASM (Cr)
```

(2)　6800 → 8080 の時

```
B>MPL (Cr)
/MPL PROGRAM = CVTGT (Cr)
/INPUT FILE  = MW3 (Cr)
/OUTPUT FILE = I−×××. ASM (Cr)
```

## 4.11　ターゲットソースファイルのプリント

変換後のファイルのリストを取る。




M－×××. ASM
or
I－×××. ASM

出力ソースリスト

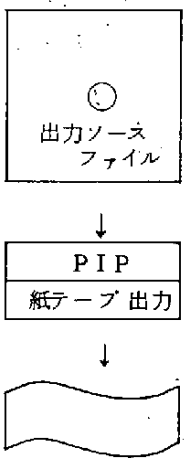
（8080 → 6800 の時）

A> PIP PRN：＝B：M－×××. ASM (C_r)
or
A> PIP PRN：＝B：I－×××. ASM (C_r)

（6800 → 8080 の時）

## 4.12　出力ソース紙テープ（ターゲットテープ）の書き出し

変換後のファイルの紙テープを作成する。




M－×××. ASM
or
I－×××. ASM

6800 ターゲットテープ
or
8080 ターゲットテープ

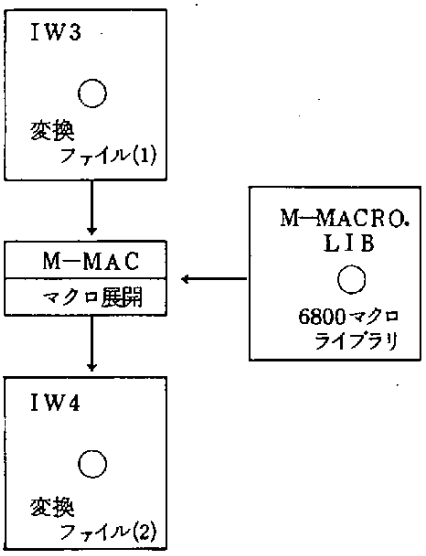
（8080 → 6800 の時）

A> PIP PUN：＝B：M－×××. ASM (C_r)
or
A> PIP PUN：＝B：I－×××. ASM (C_r)

（6800 → 8080 の時）

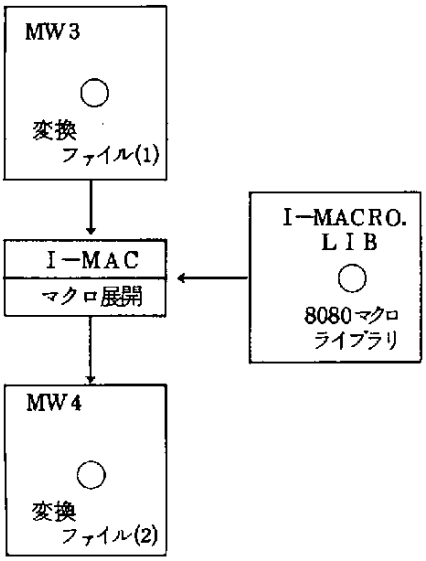
## 4.13　ターゲットマシンマクロ展開

ターゲットマシンにて標準サブルーチンが装備されている時、及びI／Oモジュールがある時は有効な処理であり、サブルーチンのリンケージとサブルーチンの本体及びワークエリアを展開する。




1）8080 → 6800 の時

```
B>MPL (Cr)
/MPL PROGRAM = M—MAC (Cr)
/LEFT 16 BYTE〔Y/N〕IO= YY (Cr)
/INPUT FILE  = IW3 (Cr)
/OUTPUT FILE = IW4 (Cr)
```




2）6800 → 8080 の時

```
B>MPL (Cr)
/MPL PROGRAM = I—MAC (Cr)
/LEFT 16 BYTE〔Y/N〕IO= YY (Cr)
/INPUT FILE  = MW3 (Cr)
/OUTPUT FILE = MW4 (Cr)
```

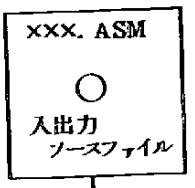
（注1）　コンバータとして用いているので左16バイト有となるので

/LEFT 16 BYTE〔Y/N〕IO=YY と入力する。

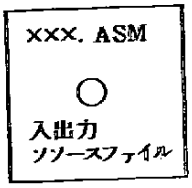
（注2）　マクロライブラリの言語は機能仕様書"マクロプロセッサ編"参照のこと。

## 4.14　タブ置換処理

オリジナルマシンの入力ソースがＣＰ／Ｍ標準の８桁毎に合っていない時、及びターゲットマシンの出力ソースがやはりＣＰ／Ｍ標準に合っていない時にタブ位置を合わせる操作である。

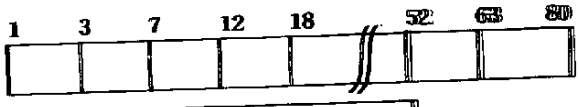



入力ファイル　　B>CVTAB (Cr)
入力(Tab)位置　　／INPUT FILE = XXX. ASM (Cr)
／TAB POSITION = 3, 7, 12, 18, ---, 52, 63 (Cr)
or
*
or
(Cr)
or
0


出力ファイル　　／OUTPUT FILE = XXX. ASM (Cr)
出力(Tab)位置　　／TAB POSITION = 2, 6, 11, 17, ---, 66, 80 (Cr)
or
*
or
(Cr)
or
0

カラム




の様なTab Position の時

入力は 3, 7, 12, 18, ……, 52, 63 と入力する。

- CP/M 標準時（ 9, 17, 25, 33, 41, 49, 57, 65, 73 ）は * を入力する。
- Tabなしでデータが (Cr) まで (sp) で置換える時は (Cr) を入力する。
- Tabなしでデータが (Cr) 以後８０カラムまで (sp) で置換える時は 0 を入力する。

（注１）(Tab)の個数は最大２０ケまで可能である。

# 添付資料

# 添付資料

## 1. 全命令展開例

# 目　　次

## 1. 8080 → 6800 展開

```
  ID:IALL        ***  8080-6800 PROGRAM CONVERT LIST  ***

SEQ SEQ CZSPA%  LABEL   CODE    OPERAND          COMMENT

00010001        ****************************************
00020002        *                                      *
00030003        *   8080 ASSEMBLER INSTRUCTION SET     *
00040004        *                FILE=I-ALL.ASM        *
00050005        *                                      *
00060006        ****************************************
00070007        *
0008            -       ORG     1000H
    0008        +       ORG     $1000

00090009        *
0010    .....1  -       NOP
    0010        +       NOP

0011    .....3  -       LXI     B,B1
    0011        +       LDX     #B1
    0012        +       STX     MB

0012    .....3  -       LXI     D,B1
    0013        +       LDX     #B1
    0014        +       STX     MD

0013    .....3  -       LXI     H,B1
    0015        +       LDX     #B1
    0016        +       STX     MH

0014    .....3  -       LXI     SP,B1
    0017        +       LDX     #B1
    0018        +       TXS

0015    V....1  -       DAD     B
    0019        +       LDAB    MB+1
    0020        +       ADDB    ML
    0021        +       STAB    ML
    0022        +       LDAB    MB
    0023        +       ADCB    MH
    0024        +       STAB    MH

0016    V....1  -       DAD     D
    0025        +       LDAB    MD+1
    0026        +       ADDB    ML
    0027        +       STAB    ML
    0028        +       LDAB    MD
    0029        +       ADCB    MH
    0030        +       STAB    MH

0017    V....1  -       DAD     H
    0031        +       LDAB    MH+1
    0032        +       ADDB    ML
    0033        +       STAB    ML
    0034        +       LDAB    MH
    0035        +       ADCB    MH
    0036        +       STAB    MH

0018    V....1  -       DAD     SP
    0037        +       TSX
    0038        +       STX     MSP
    0039        +       LDAB    MSP+1
    0040        +       ADDB    ML
    0041        +       STAB    ML
    0042        +       LDAB    MSP
    0043        +       ADCB    MH
    0044        +       STAB    MH

0019    .....1  -       STAX    B
    0045        +       LDX     MB
    0046        +       STAA    0,X
```

```
0020     .....1 -       STAX    D
    0047        +       LDX     MD
    0048        +       STAA    0,X

0021     .....1 -       LDAX    B
    0049        +       LDX     MB
    0050        +       LDAA    0,X

0022     .....1 -       LDAX    D
    0051        +       LDX     MD
    0052        +       LDAA    0,X

0023     .....3 -       SHLD    B1
    0053        +       LDX     MH
    0054        +       STX     B1

0024     .....3 -       STA     B1
    0055        +       STAA    B1

0025     .....3 -       LHLD    B1
    0056        +       LDX     B1
    0057        +       STX     MH

0026     .....3 -       LDA     B1
    0058        +       LDAA    B1

0027     .....1 -       INX     B
    0059        +       LDX     MB
    0060        +       INX
    0061        +       STX     MB

0028     .....1 -       INX     D
    0062        +       LDX     MD
    0063        +       INX
    0064        +       STX     MD

0029     .....1 -       INX     H
    0065        +       LDX     MH
    0066        +       INX
    0067        +       STX     MH

0030     .....1 -       INX     SP
    0068        +       INS

0031     .....1 -       DCX     B
    0069        +       LDX     MB
    0070        +       DEX
    0071        +       STX     MB

0032     .....1 -       DCX     D
    0072        +       LDX     MD
    0073        +       DEX
    0074        +       STX     MD

0033     .....1 -       DCX     H
    0075        +       LDX     MH
    0076        +       DEX
    0077        +       STX     MH

0034     .....1 -       DCX     SP
    0078        +       DES

0035     .VVVV1 -       INR     B
    0079        +       INC     MB

0036     .VVVV1 -       INR     C
    0080        +       INC     MC

0037     .VVVV1 -       INR     D
    0081        +       INC     MD

0038     .VVVV1 -       INR     E
    0082        +       INC     ME

0039     .VVVV1 -       INR     H
    0083        +       INC     MH

0040     .VVVV1 -       INR     L
    0084        +       INC     ML

0041     .VVVV1 -       INR     M
    0085        +       LDX     MH
    0086        +       INC     0,X

0042     .VVVV1 -       INR     A
    0087        +       INCA

0043     .VVVV1 -       DCR     B
    0088        +       DEC     MB

0044     .VVVV1 -       DCR     C
    0089        +       DEC     MC

0045     .VVVV1 -       DCR     D
    0090        +       DEC     MD

0046     .VVVV1 -       DCR     E
    0091        +       DEC     ME

0047     .VVVV1 -       DCR     H
    0092        +       DEC     MH

0048     .VVVV1 -       DCR     L
    0093        +       DEC     ML

0049     .VVVV1 -       DCR     M
    0094        +       LDX     MH
    0095        +       DEC     0,X

0050     .VVVV1 -       DCR     A
    0096        +       DECA

0051     V....1 -       RLC
    0097        +       ASRA
    0098        +       ROLA
    0099        +       ROLA

0052     V....1 -       RAL
    0100        +       ROLA

0053     V....1 -       RRC
    0101        +       PSHA
    0102        +       ASRA
    0103        +       PULA
    0104        +       RORA

0054     V....1 -       RAR
    0105        +       RORA

0055     VVVVV1 -       DAA
    0106        +       DAA

0056     S....1 -       STC
    0107        +       SEC

0057     .....1 -       CMA
    0108        +       RORB
    0109        +       COMA
    0110        +       ROLB

0058     V....1 -       CMC
    0111        +       PSHA
    0112        +       TPA
    0113        +       EORA    #1
    0114        +       TAP
    0115        +       PULA
```

```
0059     .....2 -        MVI     B,B1
0116            +        LDAB    #B1
0117            +        STAB    MB

0060     .....2 -        MVI     C,B1
0118            +        LDAB    #B1
0119            +        STAB    MC

0061     .....2 -        MVI     D,B1
0120            +        LDAB    #B1
0121            +        STAB    MD

0062     .....2 -        MVI     E,B1
0122            +        LDAB    #B1
0123            +        STAB    ME

0063     .....2 -        MVI     H,B1
0124            +        LDAB    #B1
0125            +        STAB    MH

0064     .....2 -        MVI     L,B1
0126            +        LDAB    #B1
0127            +        STAB    ML

0065     .....2 -        MVI     M,B1
0128            +        LDAB    #B1
0129            +        LDX     MH
0130            +        STAB    0,X

0066     .....2 -        MVI     A,B1
0131            +        LDAA    #B1

0067     .....1 -        MOV     B,B
0132            +        NOP

0068     .....1 -        MOV     B,C
0133            +        LDAB    MC
0134            +        STAB    MB

0069     .....1 -        MOV     B,D
0135            +        LDAB    MD
0136            +        STAB    MB

0070     .....1 -        MOV     B,E
0137            +        LDAB    ME
0138            +        STAB    MB

0071     .....1 -        MOV     B,H
0139            +        LDAB    MH
0140            +        STAB    MB

0072     .....1 -        MOV     B,L
0141            +        LDAB    ML
0142            +        STAB    MB

0073     .....1 -L:      MOV     B,M
0143            +L       LDX     MH
0144            +        LDAB    0,X
0145            +        STAB    MB

0074     .....1 -        MOV     B,A
0146            +        STAA    MB

0075     .....1 -        MOV     C,B
0147            +        LDAB    MB
0148            +        STAB    MC

0076     .....1 -        MOV     C,C
0149            +        NOP

0077     .....1 -        MOV     C,D
0150            +        LDAB    MD
0151            +        STAB    MC

0078     .....1 -        MOV     C,E
0152            +        LDAB    ME
0153            +        STAB    MC

0079     .....1 -        MOV     C,H
0154            +        LDAB    MH
0155            +        STAB    MC

0080     .....1 -        MOV     C,L
0156            +        LDAB    ML
0157            +        STAB    MC

0081     .....1 -L:      MOV     C,M
0158            +L       LDX     MH
0159            +        LDAB    0,X
0160            +        STAB    MC

0082     .....1 -        MOV     C,A
0161            +        STAA    MC

0083     .....1 -        MOV     D,B
0162            +        LDAB    MB
0163            +        STAB    MD

0084     .....1 -        MOV     D,C
0164            +        LDAB    MC
0165            +        STAB    MD

0085     .....1 -        MOV     D,D
0166            +        NOP

0086     .....1 -        MOV     D,E
0167            +        LDAB    ME
0168            +        STAB    MD

0087     .....1 -        MOV     D,H
0169            +        LDAB    MH
0170            +        STAB    MD

0088     .....1 -        MOV     D,L
0171            +        LDAB    ML
0172            +        STAB    MD

0089     .....1 -L:      MOV     D,M
0173            +L       LDX     MH
0174            +        LDAB    0,X
0175            +        STAB    MD

0090     .....1 -        MOV     D,A
0176            +        STAA    MD

0091     .....1 -        MOV     E,B
0177            +        LDAB    MB
0178            +        STAB    ME

0092     .....1 -        MOV     E,C
0179            +        LDAB    MC
0180            +        STAB    ME

0093     .....1 -        MOV     E,D
0181            +        LDAB    MD
0182            +        STAB    ME

0094     .....1 -        MOV     E,E
0183            +        NOP

0095     .....1 -        MOV     E,H
0184            +        LDAB    MH
0185            +        STAB    ME

0096     .....1 -        MOV     E,L
0186            +        LDAB    ML
0187            +        STAB    ME
```

```
0097      .....1 -L:    MOV     E,M
   0188          +L     LDX     MH
   0189          +      LDAB    0,X
   0190          +      STAB    ME

0098      .....1 -       MOV     E,A
   0191          +      STAA    ME

0099      .....1 -       MOV     H,B
   0192          +      LDAB    MB
   0193          +      STAB    MH

0100      .....1 -       MOV     H,C
   0194          +      LDAB    MC
   0195          +      STAB    MH

0101      .....1 -       MOV     H,D
   0196          +      LDAB    MD
   0197          +      STAB    MH

0102      .....1 -       MOV     H,E
   0198          +      LDAB    ME
   0199          +      STAB    MH

0103      .....1 -       MOV     H,H
   0200          +      NOP

0104      .....1 -       MOV     H,L
   0201          +      LDAB    ML
   0202          +      STAB    MH

0105      .....1 -       MOV     H,M
   0203          +      LDX     MH
   0204          +      LDAB    0,X
   0205          +      STAB    MH

0106      .....1 -       MOV     H,A
   0206          +      STAA    MH

0107      .....1 -       MOV     L,B
   0207          +      LDAB    MB
   0208          +      STAB    ML

0108      .....1 -       MOV     L,C
   0209          +      LDAB    MC
   0210          +      STAB    ML

0109      .....1 -       MOV     L,D
   0211          +      LDAB    MD
   0212          +      STAB    ML

0110      .....1 -       MOV     L,E
   0213          +      LDAB    ME
   0214          +      STAB    ML

0111      .....1 -       MOV     L,H
   0215          +      LDAB    MH
   0216          +      STAB    ML

0112      .....1 -       MOV     L,L
   0217          +      NOP

0113      .....1 -       MOV     L,M
   0218          +      LDX     MH
   0219          +      LDAB    0,X
   0220          +      STAB    ML

0114      .....1 -       MOV     L,A
   0221          +      STAA    ML

0115      .....1 -       MOV     M,B
   0222          +      LDAB    MB
   0223          +      LDX     MH
   0224          +      STAB    0,X

0116      .....1 -L:     MOV     M,C
   0225          +L     LDAB    MC
   0226          +      LDX     MH
   0227          +      STAB    0,X

0117      .....1 -L:     MOV     M,D
   0228          +L     LDAB    MD
   0229          +      LDX     MH
   0230          +      STAB    0,X

0118      .....1 -L:     MOV     M,E
   0231          +L     LDAB    ME
   0232          +      LDX     MH
   0233          +      STAB    0,X

0119      .....1 -L:     MOV     M,H
   0234          +L     LDAB    MH
   0235          +      LDX     MH
   0236          +      STAB    0,X

0120      .....1 -L:     MOV     M,L
   0237          +L     LDAB    ML
   0238          +      LDX     MH
   0239          +      STAB    0,X

0121      .....1 -L:     MOV     M,A
   0240          +L     LDX     MH
   0241          +      STAA    0,X

0122      .....1 -       MOV     A,B
   0242          +      LDAA    MB

0123      .....1 -       MOV     A,C
   0243          +      LDAA    MC

0124      .....1 -       MOV     A,D
   0244          +      LDAA    MD

0125      .....1 -       MOV     A,E
   0245          +      LDAA    ME

0126      .....1 -       MOV     A,H
   0246          +      LDAA    MH

0127      .....1 -       MOV     A,L
   0247          +      LDAA    ML

0128      .....1 -L:     MOV     A,M
   0248          +L     LDX     MH
   0249          +      LDAB    0,X
   0250          +      STAB    MA

0129      .....1 -       MOV     A,A
   0251          +      NOP

0130      .....1 -       HLT
   0252          +      WAI

0131      VVVVV1 -       ADD     B
   0253          +      ADDA    MB

0132      VVVVV1 -       ADD     C
   0254          +      ADDA    MC

0133      VVVVV1 -       ADD     D
   0255          +      ADDA    MD

0134      VVVVV1 -       ADD     E
   0256          +      ADDA    ME

0135      VVVVV1 -       ADD     H
   0257          +      ADDA    MH

0136      VVVVV1 -       ADD     L
   0258          +      ADDA    ML
```

```
0137    VVVVV1 -L:      ADD     M
    0259       +L       LDX     MH
    0260       +        ADDA    0,X

0138    VVVVV1 -        ADD     A
    0261       +        TAB
    0262       +        ABA

0139    VVVVV1 -        ADC     B
    0263       +        ADCA    MB

0140    VVVVV1 -        ADC     C
    0264       +        ADCA    MC

0141    VVVVV1 -        ADC     D
    0265       +        ADCA    MD

0142    VVVVV1 -        ADC     E
    0266       +        ADCA    ME

0143    VVVVV1 -        ADC     H
    0267       +        ADCA    MH

0144    VVVVV1 -        ADC     L
    0268       +        ADCA    ML

0145    VVVVV1 -L:      ADC     M
    0269       +L       LDX     MH
    0270       +        ADCA    0,X

0146    VVVVV1 -        ADC     A
    0271       +        PSHA
    0272       +        TSX
    0273       +        ADCA    0,X
    0274       +        PULB

0147    VVVVV1 -        SUB     B
    0275       +        SUBA    MB

0148    VVVVV1 -        SUB     C
    0276       +        SUBA    MC

0149    VVVVV1 -        SUB     D
    0277       +        SUBA    MD

0150    VVVVV1 -        SUB     E
    0278       +        SUBA    ME

0151    VVVVV1 -        SUB     H
    0279       +        SUBA    MH

0152    VVVVV1 -        SUB     L
    0280       +        SUBA    ML

0153    VVVVV1 -        SUB     M
    0281       +        LDX     MH
    0282       +        SUBA    0,X

0154    VVVVV1 -        SUB     A
    0283       +        TAB
    0284       +        SBA

0155    VVVVV1 -        SBB     B
    0285       +        SBCA    MB

0156    VVVVV1 -        SBB     C
    0286       +        SBCA    MC

0157    VVVVV1 -        SBB     D
    0287       +        SBCA    MD

0158    VVVVV1 -        SBB     E
    0288       +        SBCA    ME

0159    VVVVV1 -        SBB     H
    0289       +        SBCA    MH

0160    VVVVV1 -        SBB     L
    0290       +        SBCA    ML

0161    VVVVV1 -L:      SBB     M
    0291       +L       LDX     MH
    0292       +        SBCA    0,X

0162    VVVVV1 -        SBB     A
    0293       +        PSHA
    0294       +        TSX
    0295       +        SBCA    0,X
    0296       +        PULB

0163    RVVVV1 -        ANA     B
    0297       +        CLC
    0298       +        ANDA    MB

0164    RVVVV1 -        ANA     C
    0299       +        CLC
    0300       +        ANDA    MC

0165    RVVVV1 -        ANA     D
    0301       +        CLC
    0302       +        ANDA    MD

0166    RVVVV1 -        ANA     E
    0303       +        CLC
    0304       +        ANDA    ME

0167    RVVVV1 -        ANA     H
    0305       +        CLC
    0306       +        ANDA    MH

0168    RVVVV1 -        ANA     L
    0307       +        CLC
    0308       +        ANDA    ML

0169    RVVVV1 -        ANA     M
    0309       +        CLC
    0310       +        LDX     MH
    0311       +        ANDA    0,X

0170    RVVVV1 -        ANA     A
    0312       +        CLC
    0313       +        PSHA
    0314       +        TSX
    0315       +        ANDA    0,X
    0316       +        PULB

0171    RVVVR1 -        XRA     B
    0317       +        CLC
    0318       +        EORA    MB

0172    RVVVR1 -        XRA     C
    0319       +        CLC
    0320       +        EORA    MC

0173    RVVVR1 -        XRA     D
    0321       +        CLC
    0322       +        EORA    MD

0174    RVVVR1 -        XRA     E
    0323       +        CLC
    0324       +        EORA    ME

0175    RVVVR1 -        XRA     H
    0325       +        CLC
    0326       +        EORA    MH

0176    RVVVR1 -        XRA     L
    0327       +        CLC
    0328       +        EORA    ML
```

```
0177     RVVVR1 -        XRA     M
    0329        +        CLC
    0330        +        LDX     MH
    0331        +        EORA    0,X

0178     RVVVR1 -        XRA     A
    0332        +        CLC
    0333        +        PSHA
    0334        +        TSX
    0335        +        EORA    0,X
    0336        +        PULB

0179     RVVVR1 -        ORA     B
    0337        +        CLC
    0338        +        ORAA    MB

0180     RVVVR1 -        ORA     C
    0339        +        CLC
    0340        +        ORAA    MC

0181     RVVVR1 -        ORA     D
    0341        +        CLC
    0342        +        ORAA    MD

0182     RVVVR1 -        ORA     E
    0343        +        CLC
    0344        +        ORAA    ME

0183     RVVVR1 -        ORA     H
    0345        +        CLC
    0346        +        ORAA    MH

0184     RVVVR1 -        ORA     L
    0347        +        CLC
    0348        +        ORAA    ML

0185     RVVVR1 -        ORA     M
    0349        +        CLC
    0350        +        LDX     MH
    0351        +        ORAA    0,X

0186     RVVVR1 -        ORA     A
    0352        +        CLC
    0353        +        PSHA
    0354        +        TSX
    0355        +        ORAA    0,X
    0356        +        PULB

0187     VVVVV1 -        CMP     B
    0357        +        CMPA    MB

0188     VVVVV1 -        CMP     C
    0358        +        CMPA    MC

0189     VVVVV1 -        CMP     D
    0359        +        CMPA    MD

0190     VVVVV1 -        CMP     E
    0360        +        CMPA    ME

0191     VVVVV1 -        CMP     H
    0361        +        CMPA    MH

0192     VVVVV1 -        CMP     L
    0362        +        CMPA    ML

0193     VVVVV1 -        CMP     M
    0363        +        LDX     MH
    0364        +        CMPA    0,X

0194     VVVVV1 -        CMP     A
    0365        +        TAB
    0366        +        CAB

0195     VVVVV2 -        ADI     B1
    0367        +        ADDA    #B1

0196     VVVVV2 -        ACI     B1
    0368        +        ADCA    #B1

0197     VVVVV2 -        SUI     B1
    0369        +        SUBA    #B1

0198     VVVVV2 -        SBI     B1
    0370        +        SBCA    #B1

0199     RVVVV2 -        ANI     B1
    0371        +        CLC
    0372        +        ANDA    #B1

0200     RVVVR2 -        XRI     B1
    0373        +        CLC
    0374        +        EORA    #B1

0201     RVVVR2 -        ORI     B1
    0375        +        CLC
    0376        +        ORAA    #B1

0202     VVVVV2 -        CPI     B1
    0377        +        CMPA    #B1

0203 NC .....1 -         RST     B1
0204    .....3 -         CALL    B1
    0378        +        JSR     B1

0205    .U...3 -         CZ      B1
    0379        +        BNE     *+5
    0380        +        JSR     B1

0206    .U...3 -         CNZ     B1
    0381        +        BEQ     *+5
    0382        +        JSR     B1

0207    U....3 -         CC      B1
    0383        +        BCC     *+5
    0384        +        JSR     B1

0208    U....3 -         CNC     B1
    0385        +        BCS     *+5
    0386        +        JSR     B1

0209 NC ...U.3 -         CPE     B1
0210 NC ...U.3 -         CPO     B1
0211    ..U..3 -         CM      B1
    0387        +        BPL     *+5
    0388        +        JSR     B1

0212    ..U..3 -         CP      B1
    0389        +        BMI     *+5
    0390        +        JSR     B1

0213    .....3 -         JMP     B1
    0391        +        JMP     B1

0214    .U...3 -         JZ      B1
    0392        +        BNE     *+5
    0393        +        JMP     B1

0215    .U...3 -         JNZ     B1
    0394        +        BEQ     *+5
    0395        +        JMP     B1

0216    U....3 -         JC      B1
    0396        +        BCC     *+5
    0397        +        JMP     B1

0217    U....3 -         JNC     B1
    0398        +        BCS     *+5
```

```
    0399        +           JMP     B1

0218 NC ...U.3  -           JPE     B1
0219 NC ...U.3  -           JPO     B1
0220    ..U..3  -           JM      B1
    0400        +           BPL     **5
    0401        +           JMP     B1

0221    ..U..3  -           JP      B1
    0402        +           BMI     **5
    0403        +           JMP     B1

0222 NC .....2  -           IN      B1

0223 NC .....2  -           OUT     B1
0224    .....1  -           XTHL
    0404        +           TSX
    0405        +           LDAB    1,X
    0406        +           PSHB
    0407        +           LDAB    0,X
    0408        +           PSHB
    0409        +           LDAB    ML
    0410        +           STAB    1,X
    0411        +           LDAB    MH
    0412        +           STAB    0,X
    0413        +           PULB
    0414        +           STAB    MH
    0415        +           PULB
    0416        +           STAB    ML

0225    .....1  -           XCHG
    0417        +           LDX     MD
    0418        +           LDAB    MH
    0419        +           STAB    MD
    0420        +           LDAB    ML
    0421        +           STAB    ME
    0422        +           STX     MH

0226    .....1  -           EI
    0423        +           CLI

0227    .....1  -           DI
    0424        +           SEI

0228    .....1  -L:         PCHL
    0425        +L          LDX     MH
    0426        +           JMP     0,X

0229    .....1  -           SPHL
    0427        +           LDX     MH
    0428        +           TXS

0230    .....1  -           RET
    0429        +           RTS

0231    .U...1  -           RZ
    0430        +           BNE     **3
    0431        +           RTS

0232    .U...1  -           RNZ
    0432        +           BEQ     **3
    0433        +           RTS

0233    U....1  -           RC
    0434        +           BCC     **3
    0435        +           RTS

0234    U....1  -           RNC
    0436        +           BCS     **3
    0437        +           RTS

0235 NC ...U.1  -           RPE
0236 NC ...U.1  -           RPO
0237    ..U..1  -           RM
    0438        +           BPL     **3
    0439        +           RTS

0238    ..U..1  -           RP
    0440        +           BMI     **3
    0441        +           RTS

0239    .....1  -           POP     B
    0442        +           PULB
    0443        +           STAB    MB+1
    0444        +           PULB
    0445        +           STAB    MB

0240    .....1  -           POP     D
    0446        +           PULB
    0447        +           STAB    MD+1
    0448        +           PULB
    0449        +           STAB    MD

0241    .....1  -           POP     H
    0450        +           PULB
    0451        +           STAB    MH+1
    0452        +           PULB
    0453        +           STAB    MH

0242    VVVVV1  -           POP     PSW
    0454        +           PULA
    0455        +           TAP
    0456        +           PULA

0243    .....1  -           PUSH    B
    0457        +           LDAB    MB
    0458        +           PSHB
    0459        +           LDAB    MB+1
    0460        +           PSHB

0244    .....1  -           PUSH    D
    0461        +           LDAB    MD
    0462        +           PSHB
    0463        +           LDAB    MD+1
    0464        +           PSHB

0245    .....1  -           PUSH    H
    0465        +           LDAB    MH
    0466        +           PSHB
    0467        +           LDAB    MH+1
    0468        +           PSHB

0246    .....1  -           PUSH    PSW
    0469        +           TAB
    0470        +           PSHA
    0471        +           TPA
    0472        +           PSHA
    0473        +           TBA

02470474        ***

    0248        -B1:        DB      5       ; DECIMAL
        0475    +B1         FCB     5       * DECIMAL

    0249        -B2:        DB      5D
        0476    +B2         FCB     5
```

```
0250            -B3:    DB      0E5H    ; HEXA
    0477        +B3     FCB     $E5     * HEXA

0251            -       DB      0050    ; OCTAL
    0478        +       FCB     @005    * OCTAL

0252            -       DB      0050
    0479        +       FCB     @005

0253            -       DB      0101B   ; BINARY
    0480        +       FCB     %0101   * BINARY

0254            -       DB      'A      ; CHARACTER
    0481        +       FCC     /A/     * CHARACTER

0255            -       DB      'TEXT'
    0482        +       FCC     /TEXT/

0256            -LABEL: DW      1000H
    0483        +LABEL  FDB     $1000

0257            -       DW      LABEL
    0484        +       FDB     LABEL

0258            -       DW      LABEL+3
    0485        +       FDB     LABEL+3

0259            -       DW      $
    0486        +       FDB     *

0260            -       DW      $+13
    0487        +       FDB     *+13

0261            -       DW      'A'
    0488        +       FDB     $41

0262            -       DW      'AB'
    0489        +       FDB     $4142

02630490         *
0264            -       DS      38
    0491        +       RMB     38
02650492         ***
0266            -TAG1   EQU     LABEL
    0493        +TAG1   EQU     LABEL

0267            -TAG2   EQU     LABEL-3
    0494        +TAG2   EQU     LABEL-3

0268            -TAG3   EQU     $-6
    0495        +TAG3   EQU     *-6

02690496         *
0270            -       REG
    0497        +*** REG 8080 TO 6800 CONVERT
    0498        +*      A   --> A
    0499        +*      B,C --> MEMORY B,C (NOT C,B)
    0500        +*      D,E --> MEMORY D,E (NOT E,D)
    0501        +*      H,L --> MEMORY H,L (NOT L,H)
    0502        +*      F   --> CCR
    0503        +*              B,X WORK REG
    0504        +MB     RMB     1       * B REG
    0505        +MC     RMB     1       * C REG
    0506        +MD     RMB     1       * D REG
    0507        +ME     RMB     1       * E REG
    0508        +MH     RMB     1       * H REG
    0509        +ML     RMB     1       * L REG
    0510        +MSP    RMB     2       * SP
    0511        +*** ----

02710512         *
02720513                 END
```

## 2. 6800 → 8080 展開

```
  ID:MALL        ***  6800-8080 PROGRAM CONVERT LIST  ***
SEQ SEQ HNZVC%  LABEL   CODE    OPERAND          COMMENT

00010001        ;*****************************************
00020002        ;                                        *
00030003        ;   6800 ASSEMBLER INSTRUCTION SET      *
00040004        ;                FILE=M-ALL.ASM          *
00050005        ;                                        *
00060006        ;*****************************************
00070007        ;
0008          -         ORG     $1000
    0008      +         ORG     1000H

00090009        ;
0010    VVVVV2  -         ADD A   #B1
    0010        +         MVI     A,B1
    0011        +         ADD     B
    0012        +         MOV     B,A

0011    VVVVV3  -         ADD A   B1
    0013        +         LDA     B1
    0014        +         ADD     B
    0015        +         MOV     B,A

0012    VVVVV2  -         ADD A   B1,X
    0016        +         LXI     H,B1
    0017        +         DAD     D
    0018        +         MOV     A,M
    0019        +         ADD     B
    0020        +         MOV     B,A

0013    VVVVV1  -         ABA
    0021        +         MOV     A,C
    0022        +         ADD     B
    0023        +         MOV     B,A

0014    VVVVV2  -         ADD B   #B1
    0024        +         MVI     A,B1
    0025        +         ADD     C
    0026        +         MOV     C,A

0015    VVVVV3  -         ADD B   B1
    0027        +         LDA     B1
    0028        +         ADD     C
    0029        +         MOV     C,A

0016    VVVVV2  -         ADD B   B1,X
    0030        +         LXI     H,B1
    0031        +         DAD     D
    0032        +         MOV     A,M
    0033        +         ADD     C
    0034        +         MOV     C,A

0017    VVVVV2  -         ADC A   #B1
    0035        +         MVI     A,B1
    0036        +         ADC     B
    0037        +         MOV     B,A

0018    VVVVV3  -         ADC A   B1
    0038        +         LDA     B1
    0039        +         ADC     B
    0040        +         MOV     B,A

0019    VVVVV2  -         ADC A   B1,X
    0041        +         LXI     H,B1
    0042        +         PUSH    PSW
    0043        +         DAD     D
    0044        +         POP     PSW
    0045        +         MOV     A,M
    0046        +         ADC     B
    0047        +         MOV     B,A
```

```
0020     VVVVV2 -        ADC B   #B1
    0048        +        MVI     A,B1
    0049        +        ADC     C
    0050        +        MOV     C,A

0021     VVVVV3 -        ADC B   B1
    0051        +        LDA     B1
    0052        +        ADC     C
    0053        +        MOV     C,A

0022     VVVVV2 -        ADC B   B1,X
    0054        +        LXI     H,B1
    0055        +        PUSH    PSW
    0056        +        DAD     D
    0057        +        POP     PSW
    0058        +        MOV     A,M
    0059        +        ADC     C
    0060        +        MOV     C,A

0023     .VVR.2 -        AND A   #B1
    0061        +        PUSH    PSW
    0062        +        MVI     A,B1
    0063        +        ANA     B
    0064        +        MOV     B,A
    0065        +        XTHL
    0066        +        MOV     A,L
    0067        +        RRC
    0068        +        POP     H

0024     .VVR.3 -        AND A   B1
    0069        +        PUSH    PSW
    0070        +        LDA     B1
    0071        +        ANA     B
    0072        +        MOV     B,A
    0073        +        XTHL
    0074        +        MOV     A,L
    0075        +        RRC
    0076        +        POP     H

0025     .VVR.2 -        AND A   B1,X
    0077        +        PUSH    PSW
    0078        +        LXI     H,B1
    0079        +        DAD     D
    0080        +        MOV     A,M
    0081        +        ANA     B
    0082        +        MOV     B,A
    0083        +        XTHL
    0084        +        MOV     A,L
    0085        +        RRC
    0086        +        POP     H

0026     .VVR.2 -        AND B   #B1
    0087        +        PUSH    PSW
    0088        +        MVI     A,B1
    0089        +        ANA     C
    0090        +        MOV     C,A
    0091        +        XTHL
    0092        +        MOV     A,L
    0093        +        RRC
    0094        +        POP     H

0027     .VVR.3 -        AND B   B1
    0095        +        PUSH    PSW
    0096        +        LDA     B1
    0097        +        ANA     C
    0098        +        MOV     C,A
    0099        +        XTHL
    0100        +        MOV     A,L
    0101        +        RRC
    0102        +        POP     H

0028     .VVR.2 -        AND B   B1,X
    0103        +        PUSH    PSW
    0104        +        LXI     H,B1
    0105        +        DAD     D
    0106        +        MOV     A,M
    0107        +        ANA     C
    0108        +        MOV     C,A
    0109        +        XTHL
    0110        +        MOV     A,L
    0111        +        RRC
    0112        +        POP     H

0029     .VVR.2 -        BIT A   #B1
    0113        +        PUSH    PSW
    0114        +        MVI     A,B1
    0115        +        ANA     B
    0116        +        XTHL
    0117        +        MOV     A,L
    0118        +        RRC
    0119        +        POP     H

0030     .VVR.3 -        BIT A   B1
    0120        +        PUSH    PSW
    0121        +        LDA     B1
    0122        +        ANA     B
    0123        +        XTHL
    0124        +        MOV     A,L
    0125        +        RRC
    0126        +        POP     H

0031     .VVR.2 -        BIT A   B1,X
    0127        +        PUSH    PSW
    0128        +        LXI     H,B1
    0129        +        DAD     D
    0130        +        MOV     A,M
    0131        +        ANA     B
    0132        +        XTHL
    0133        +        MOV     A,L
    0134        +        RRC
    0135        +        POP     H

0032     .VVR.2 -        BIT B   #B1
    0136        +        PUSH    PSW
    0137        +        MVI     A,B1
    0138        +        ANA     C
    0139        +        XTHL
    0140        +        MOV     A,L
    0141        +        RRC
    0142        +        POP     H

0033     .VVR.3 -        BIT B   B1
    0143        +        PUSH    PSW
    0144        +        LDA     B1
    0145        +        ANA     C
    0146        +        XTHL
    0147        +        MOV     A,L
    0148        +        RRC
    0149        +        POP     H

0034     .VVR.2 -        BIT B   B1,X
    0150        +        PUSH    PSW
    0151        +        LXI     H,B1
    0152        +        DAD     D
    0153        +        MOV     A,M
    0154        +        ANA     C
    0155        +        XTHL
    0156        +        MOV     A,L
    0157        +        RRC
    0158        +        POP     H

0035     .RSRR2 -        CLR     B1,X
    0159        +        LXI     H,B1
    0160        +        DAD     D
    0161        +        XRA     A
    0162        +        MOV     M,A
```

```
0036    .RSRR3 -        CLR     B1
    0163       +        XRA     A
    0164       +        STA     B1

0037    .RSRR1 -        CLR A
    0165       +        XRA     A
    0166       +        MOV     B,A

0038    .RSRR1 -        CLR B
    0167       +        XRA     A
    0168       +        MOV     C,A

0039    .VVVV2 -        CMP A   #B1
    0169       +        MOV     A,B
    0170       +        CPI     B1

0040    .VVVV2 -        CMP B   #B1
    0171       +        MOV     A,C
    0172       +        CPI     B1

0041    .VVVV3 -        CMP A   B1
    0173       +        LXI     H,B1
    0174       +        MOV     A,B
    0175       +        CMP     M

0042    .VVVV3 -        CMP B   B1
    0176       +        LXI     H,B1
    0177       +        MOV     A,C
    0178       +        CMP     M

0043    .VVVV2 -        CMP A   B1,X
    0179       +        LXI     H,B1
    0180       +        DAD     D
    0181       +        MOV     A,B
    0182       +        CMP     M

0044    .VVVV2 -        CMP B   B1,X
    0183       +        LXI     H,B1
    0184       +        DAD     D
    0185       +        MOV     A,C
    0186       +        CMP     M

0045    .VVVV1 -        CBA
    0187       +        MOV     A,B
    0188       +        CMP     C

0046    .VVRS2 -        COM     B1,X
    0189       +        LXI     H,B1
    0190       +        DAD     D
    0191       +        MOV     A,M
    0192       +        CMA
    0193       +        MOV     M,A
    0194       +        STC                     ;SET CARRY

0047    .VVRS3 -        COM     B1
    0195       +        LXI     H,B1
    0196       +        MOV     A,M
    0197       +        CMA
    0198       +        MOV     M,A
    0199       +        STC                     ;SET CARRY

0048    .VVRS1 -        COM A
    0200       +        MOV     A,B
    0201       +        CMA
    0202       +        MOV     B,A
    0203       +        STC                     ;SET CARRY

0049    .VVRS1 -        COM B
    0204       +        MOV     A,C
    0205       +        CMA
    0206       +        MOV     C,A
    0207       +        STC                     ;SET CARRY
```

```
0050    .VVVV2 -        NEG     B1,X
    0208       +        LXI     H,B1
    0209       +        DAD     D
    0210       +        XRA     A
    0211       +        SUB     M
    0212       +        MOV     M,A

0051    .VVVV3 -        NEG     B1
    0213       +        LXI     H,B1
    0214       +        XRA     A
    0215       +        SUB     M
    0216       +        MOV     M,A

0052    .VVVV1 -        NEG A
    0217       +        XRA     A
    0218       +        SUB     B
    0219       +        MOV     B,A

0053    .VVVV1 -        NEG B
    0220       +        XRA     A
    0221       +        SUB     C
    0222       +        MOV     C,A

0054    .VVVV1 -        DAA
    0223       +        MOV     A,B
    0224       +        DAA
    0225       +        MOV     B,A

0055    .VVV.2 -        DEC     B1,X
    0226       +        LXI     H,B1
    0227       +        PUSH    PSW
    0228       +        DAD     D
    0229       +        POP     PSW
    0230       +        DCR     M

0056    .VVV.3 -        DEC     B1
    0231       +        LXI     H,B1
    0232       +        DCR     M

0057    .VVV.1 -        DEC A
    0233       +        DCR     B
0058    .VVV.1 -        DEC B
    0234       +        DCR     C

0059    .VVR.2 -        EOR A   #B1
    0235       +        PUSH    PSW
    0236       +        MVI     A,B1
    0237       +        XRA     B
    0238       +        MOV     B,A
    0239       +        XTHL
    0240       +        MOV     A,L
    0241       +        RRC
    0242       +        POP     H

0060    .VVR.3 -        EOR A   B1
    0243       +        PUSH    PSW
    0244       +        LDA     B1
    0245       +        XRA     B
    0246       +        MOV     B,A
    0247       +        XTHL
    0248       +        MOV     A,L
    0249       +        RRC
    0250       +        POP     H

0061    .VVR.2 -        EOR A   B1,X
    0251       +        PUSH    PSW
    0252       +        LXI     H,B1
    0253       +        DAD     D
    0254       +        MOV     A,M
    0255       +        XRA     B
    0256       +        MOV     B,A
    0257       +        XTHL
    0258       +        MOV     A,L
    0259       +        RRC
    0260       +        POP     H

0062    .VVR.3 -        EOR B   #B1
    0261       +        PUSH    PSW
    0262       +        MVI     A,B1
    0263       +        XRA     C
    0264       +        MOV     C,A
    0265       +        XTHL
    0266       +        MOV     A,L
    0267       +        RRC
    0268       +        POP     H

0063    .VVR.3 -        EOR B   B1
    0269       +        PUSH    PSW
    0270       +        LDA     B1
    0271       +        XRA     C
    0272       +        MOV     C,A
    0273       +        XTHL
    0274       +        MOV     A,L
    0275       +        RRC
    0276       +        POP     H

0064    .VVR.2 -        EOR B   B1,X
    0277       +        PUSH    PSW
    0278       +        LXI     H,B1
    0279       +        DAD     D
    0280       +        MOV     A,M
    0281       +        XRA     C
    0282       +        MOV     C,A
    0283       +        XTHL
    0284       +        MOV     A,L
    0285       +        RRC
    0286       +        POP     H

0065    .VVV.2 -        INC     B1,X
    0287       +        LXI     H,B1
    0288       +        PUSH    PSW
    0289       +        DAD     D
    0290       +        POP     PSW
    0291       +        INR     M

0066    .VVV.3 -        INC     B1
    0292       +        LXI     H,B1
    0293       +        INR     M

0067    .VVV.1 -        INC A
    0294       +        INR     B

0068    .VVV.1 -        INC B
    0295       +        INR     C

0069    .VVR.2 -        LDA A   #B1
    0296       +        MVI     B,B1

0070    .VVR.3 -        LDA A   B1
    0297       +        LXI     H,B1
    0298       +        MOV     B,M

0071    .VVR.2 -        LDA A   B1,X
    0299       +        LXI     H,B1
    0300       +        PUSH    PSW
    0301       +        DAD     D
    0302       +        POP     PSW
    0303       +        MOV     B,M

0072    .VVR.2 -        LDA B   #B1
    0304       +        MVI     C,B1

0073    .VVR.3 -        LDA B   B1
    0305       +        LXI     H,B1
    0306       +        MOV     C,M

0074    .VVR.2 -        LDA B   B1,X
    0307       +        LXI     H,B1
    0308       +        PUSH    PSW
    0309       +        DAD     D
```

```
    0310        +       POP     PSW
    0311        +       MOV     C,M

0075    .VVR.3  -       ORA A   #B1
    0312        +       PUSH    PSW
    0313        +       MVI     A,B1
    0314        +       ORA     B
    0315        +       MOV     B,A
    0316        +       XTHL
    0317        +       MOV     A,L
    0318        +       RRC
    0319        +       POP     H

0076    .VVR.3  -       ORA A   B1
    0320        +       PUSH    PSW
    0321        +       LDA     B1
    0322        +       ORA     B
    0323        +       MOV     B,A
    0324        +       XTHL
    0325        +       MOV     A,L
    0326        +       RRC
    0327        +       POP     H

0077    .VVR.3  -       ORA A   B1,X
    0328        +       PUSH    PSW
    0329        +       LXI     H,B1
    0330        +       DAD     D
    0331        +       MOV     A,M
    0332        +       ORA     B
    0333        +       MOV     B,A
    0334        +       XTHL
    0335        +       MOV     A,L
    0336        +       RRC
    0337        +       POP     H

0078    .VVR.2  -       ORA B   #B1
    0338        +       PUSH    PSW
    0339        +       MVI     A,B1
    0340        +       ORA     C
    0341        +       MOV     C,A
    0342        +       XTHL
    0343        +       MOV     A,L
    0344        +       RRC
    0345        +       POP     H

0079    .VVR.3  -       ORA B   B1
    0346        +       PUSH    PSW
    0347        +       LDA     B1
    0348        +       ORA     C
    0349        +       MOV     C,A
    0350        +       XTHL
    0351        +       MOV     A,L
    0352        +       RRC
    0353        +       POP     H

0080    .VVR.2  -       ORA B   B1,X
    0354        +       PUSH    PSW
    0355        +       LXI     H,B1
    0356        +       DAD     D
    0357        +       MOV     A,M
    0358        +       ORA     C
    0359        +       MOV     C,A
    0360        +       XTHL
    0361        +       MOV     A,L
    0362        +       RRC
    0363        +       POP     H

0081    .....1  -       PSH A
    0364        +       PUSH    B
    0365        +       INX     SP

0082    .....1  -       PSH B
    0366        +       MOV     A,C
    0367        +       PUSH    PSW
    0368        +       INX     SP

0083    .....1  -       PUL A
    0369        +       DCX     SP
    0370        +       POP     H
    0371        +       MOV     B,H

0084    .....1  -       PUL B
    0372        +       DCX     SP
    0373        +       POP     H
    0374        +       MOV     C,H

0085    .VVVV2  -       ROL     B1,X
    0375        +       LXI     H,B1
    0376        +       PUSH    PSW
    0377        +       DAD     D
    0378        +       POP     PSW
    0379        +       MOV     A,M
    0380        +       RAL
    0381        +       MOV     M,A

0086    .VVVV3  -       ROL     B1
    0382        +       LXI     H,B1
    0383        +       MOV     A,M
    0384        +       RAL
    0385        +       MOV     M,A

0087    .VVVV1  -       ROL A
    0386        +       MOV     A,B
    0387        +       RAL
    0388        +       MOV     B,A

0088    .VVVV1  -       ROL B
    0389        +       MOV     A,C
    0390        +       RAL
    0391        +       MOV     C,A

0089    ,VVVV2  -       ROR     B1,X
    0392        +       LXI     H,B1
    0393        +       PUSH    PSW
    0394        +       DAD     D
    0395        +       POP     PSW
    0396        +       MOV     A,M
    0397        +       RAR
    0398        +       MOV     M,A

0090    .VVVV3  -       ROR     B1
    0399        +       LXI     H,B1
    0400        +       MOV     A,M
    0401        +       RAR
    0402        +       MOV     M,A

0091    .VVVV1  -       ROR A
    0403        +       MOV     A,B
    0404        +       RAR
    0405        +       MOV     B,A

0092    .VVVV1  -       ROR B
    0406        +       MOV     A,C
    0407        +       RAR
    0408        +       MOV     C,A

0093    .VVVV2  -       ASL     B1,X
    0409        +       LXI     H,B1
    0410        +       DAD     D
    0411        +       MOV     A,M
    0412        +       ORA     A
    0413        +       RAL
    0414        +       MOV     M,A

0094    .VVVV3  -       ASL     B1
    0415        +       LXI     H,B1
    0416        +       MOV     A,M
    0417        +       ORA     A
    0418        +       RAL
    0419        +       MOV     M,A
```

```
0095    .VVVV1 -        ASL A
    0420       +        MOV     A,B
    0421       +        ORA     A
    0422       +        RAL
    0423       +        MOV     B,A

0096    .VVVV1 -        ASL B
    0424       +        MOV     A,C
    0425       +        ORA     A
    0426       +        RAL
    0427       +        MOV     C,A

0097    .VVVV2 -        ASR     B1,X
    0428       +        LXI     H,B1
    0429       +        DAD     D
    0430       +        MOV     A,M
    0431       +        RLC
    0432       +        RRC
    0433       +        RAR
    0434       +        MOV     M,A

0098    .VVVV3 -        ASR     B1
    0435       +        LXI     H,B1
    0436       +        MOV     A,M
    0437       +        RLC
    0438       +        RRC
    0439       +        RAR
    0440       +        MOV     M,A

0099    .VVVV1 -        ASR A
    0441       +        MOV     A,B
    0442       +        RLC
    0443       +        RRC
    0444       +        RAR
    0445       +        MOV     B,A

0100    .VVVV1 -        ASR B
    0446       +        MOV     A,C
    0447       +        RLC
    0448       +        RRC
    0449       +        RAR
    0450       +        MOV     C,A

0101    .RVVV2 -        LSR     B1,X
    0451       +        LXI     H,B1
    0452       +        DAD     D
    0453       +        MOV     A,M
    0454       +        ORA     A
    0455       +        RAR
    0456       +        MOV     M,A

0102    .RVVV3 -        LSR     B1
    0457       +        LXI     H,B1
    0458       +        MOV     A,M
    0459       +        ORA     A
    0460       +        RAR
    0461       +        MOV     M,A

0103    .RVVV1 -        LSR A
    0462       +        MOV     A,B
    0463       +        ORA     A
    0464       +        RAR
    0465       +        MOV     B,A

0104    .RVVV1 -        LSR B
    0466       +        MOV     A,C
    0467       +        ORA     A
    0468       +        RAR
    0469       +        MOV     C,A

0105    .VVR.3 -        STA A   B1
    0470       +        MOV     A,B
    0471       +        STA     B1

0106    .VVR.2 -        STA A   B1,X
    0472       +        LXI     H,B1
    0473       +        PUSH    PSW
    0474       +        DAD     D
    0475       +        POP     PSW
    0476       +        MOV     M,B

0107    .VVR.3 -        STA B   B1
    0477       +        MOV     A,C
    0478       +        STA     B1

0108    .VVR.2 -        STA B   B1,X
    0479       +        LXI     H,B1
    0480       +        PUSH    PSW
    0481       +        DAD     D
    0482       +        POP     PSW
    0483       +        MOV     M,C

0109    .VVVV2 -        SUB A   #B1
    0484       +        MOV     A,B
    0485       +        SUI     B1
    0486       +        MOV     B,A

0110    .VVVV2 -        SUB B   #B1
    0487       +        MOV     A,C
    0488       +        SUI     B1
    0489       +        MOV     C,A

0111    .VVVV3 -        SUB A   B1
    0490       +        LXI     H,B1
    0491       +        MOV     A,B
    0492       +        SUB     M
    0493       +        MOV     B,A

0112    .VVVV3 -        SUB B   B1
    0494       +        LXI     H,B1
    0495       +        MOV     A,C
    0496       +        SUB     M
    0497       +        MOV     C,A

0113    .VVVV2 -        SUB A   B1,X
    0498       +        LXI     H,B1
    0499       +        DAD     D
    0500       +        MOV     A,B
    0501       +        SUB     M
    0502       +        MOV     B,A

0114    .VVVV2 -        SUB B   B1,X
    0503       +        LXI     H,B1
    0504       +        DAD     D
    0505       +        MOV     A,C
    0506       +        SUB     M
    0507       +        MOV     C,A

0115    .VVVV1 -        SBA
    0508       +        MOV     A,B
    0509       +        SUB     C
    0510       +        MOV     B,A

0116    .VVVV2 -        SBC B   #B1
    0511       +        MOV     A,C
    0512       +        SBI     B1
    0513       +        MOV     C,A

0117    .VVVV2 -        SBC A   #B1
    0514       +        MOV     A,B
    0515       +        SBI     B1
    0516       +        MOV     B,A

0118    .VVVV3 -        SBC B   B1
    0517       +        LXI     H,B1
    0518       +        MOV     A,C
    0519       +        SBB     M
    0520       +        MOV     C,A
```

```
0119     .VVVV3 -        SBC A   B1
    0521        +        LXI     H,B1
    0522        +        MOV     A,B
    0523        +        SBB     M
    0524        +        MOV     B,A

0120     .VVVV2 -        SBC B   B1,X
    0525        +        LXI     H,B1
    0526        +        PUSH    PSW
    0527        +        DAD     D
    0528        +        POP     PSW
    0529        +        MOV     A,C
    0530        +        SBB     M
    0531        +        MOV     C,A

0121     .VVVV2 -        SBC A   B1,X
    0532        +        LXI     H,B1
    0533        +        PUSH    PSW
    0534        +        DAD     D
    0535        +        POP     PSW
    0536        +        MOV     A,B
    0537        +        SBB     M
    0538        +        MOV     B,A

0122     .VVR.1 -        TAB
    0539        +        MOV     C,B

0123     .VVR.1 -        TBA
    0540        +        MOV     B,C

0124     .VVRR2 -        TST     B1,X
    0541        +        LXI     H,B1
    0542        +        DAD     D
    0543        +        MOV     A,M
    0544        +        CPI     0

0125     .VVRR3 -        TST     B1
    0545        +        LDA     B1
    0546        +        CPI     0

0126     .VVRR1 -        TST A
    0547        +        MOV     A,B
    0548        +        CPI     0

0127     .VVRR1 -        TST B
    0549        +        MOV     A,C
    0550        +        CPI     0

0128     .VVV.3 -        CPX     #B1
    0551        +        LXI     H,B1
    0552        +        PUSH    B
    0553        +        PUSH    PSW
    0554        +        POP     B
    0555        +        MOV     A,C
    0556        +        ANI     0BFH        ; CLEAR Z-FLAG
    0557        +        MOV     C,A
    0558        +        MOV     A,E
    0559        +        SUB     L
    0560        +        JNZ     $+12
    0561        +        MOV     A,D
    0562        +        SBB     H
    0563        +        JNZ     $+7
    0564        +        MOV     A,C
    0565        +        ORI     40H         ; SET Z-FLAG
    0566        +        MOV     C,A
    0567        +        PUSH    B
    0568        +        POP     PSW
    0569        +        POP     B

0129     .VVV.3 -        CPX     B1
    0570        +        LXI     H,B1
    0571        +        MOV     A,M
    0572        +        INX     H
    0573        +        MOV     H,M
```

```
    0574        +       MOV     L,A
    0575        +       PUSH    B
    0576        +       PUSH    PSW
    0577        +       POP     B
    0578        +       MOV     A,C
    0579        +       ANI     0BFH            ; CLEAR Z-FLAG
    0580        +       MOV     C,A
    0581        +       MOV     A,E
    0582        +       SUB     L
    0583        +       JNZ     $+12
    0584        +       MOV     A,D
    0585        +       SBB     H
    0586        +       JNZ     $+7
    0587        +       MOV     A,C
    0588        +       ORI     40H             ; SET Z-FLAG
    0589        +       MOV     C,A
    0590        +       PUSH    B
    0591        +       POP     PSW
    0592        +       POP     B

0130    .VVV.2  -       CPX     B1,X
    0593        +       LXI     H,B1
    0594        +       PUSH    PSW
    0595        +       DAD     D
    0596        +       POP     PSW
    0597        +       MOV     A,M
    0598        +       INX     H
    0599        +       MOV     H,M
    0600        +       MOV     L,A
    0601        +       PUSH    B
    0602        +       PUSH    PSW
    0603        +       POP     B
    0604        +       MOV     A,C
    0605        +       ANI     0BFH            ; CLEAR Z-FLAG
    0606        +       MOV     C,A
    0607        +       MOV     A,E
    0608        +       SUB     L
    0609        +       JNZ     $+12
    0610        +       MOV     A,D
    0611        +       SBB     H
    0612        +       JNZ     $+7
    0613        +       MOV     A,C
    0614        +       ORI     40H             ; SET Z-FLAG
    0615        +       MOV     C,A
    0616        +       PUSH    B
    0617        +       POP     PSW
    0618        +       POP     B

0131    ..V..1  -       DEX
    0619        +       RAL
    0620        +       MOV     H,A
    0621        +       DCX     D
    0622        +       MOV     A,D
    0623        +       ORA     E
    0624        +       MOV     A,H
    0625        +       RAR

0132    .....1  -       DES
    0626        +       DCX     SP

0133    ..V..1  -       INX
    0627        +       RAL
    0628        +       MOV     H,A
    0629        +       INX     D
    0630        +       MOV     A,D
    0631        +       ORA     E
    0632        +       MOV     A,H
    0633        +       RAR

0134    .....1  -       INS
    0634        +       INX     SP

0135    .VVR.3  -       LDX     #B1
    0635        +       LXI     D,B1
```

```
0136     .VVR.3 -        LDX     B1
    0636        +        LHLD    B1
    0637        +        XCHG

0137     .VVR.2 -        LDX     B1,X
    0638        +        LXI     H,B1
    0639        +        PUSH    PSW
    0640        +        DAD     D
    0641        +        POP     PSW
    0642        +        MOV     E,M
    0643        +        INX     H
    0644        +        MOV     D,M

0138     .VVR.3 -        LDS     #B1
    0645        +        LXI     SP,B1

0139     .VVR.3 -        LDS     B1
    0646        +        LHLD    B1
    0647        +        SPHL

0140     .VVR.2 -        LDS     B1,X
    0648        +        LXI     H,B1
    0649        +        PUSH    PSW
    0650        +        DAD     D
    0651        +        POP     PSW
    0652        +        MOV     A,M
    0653        +        INX     H
    0654        +        MOV     H,M
    0655        +        MOV     L,A
    0656        +        SPHL

0141     .VVR.3 -        STX     B1
    0657        +        XCHG
    0658        +        SHLD    B1

0142     .VVR.2 -        STX     B1,X
    0659        +        LXI     H,B1
    0660        +        PUSH    PSW
    0661        +        DAD     D
    0662        +        POP     PSW
    0663        +        MOV     M,E
    0664        +        INX     H
    0665        +        MOV     M,D

0143     .VVR.3 -        STS     B1
    0666        +        LXI     H,0
    0667        +        PUSH    PSW
    0668        +        DAD     SP
    0669        +        POP     PSW
    0670        +        SHLD    B1

0144     .VVR.2 -        STS     B1,X
    0671        +        PUSH    D
    0672        +        PUSH    PSW
    0673        +        LXI     H,B1
    0674        +        DAD     D
    0675        +        XCHG
    0676        +        LXI     H,0
    0677        +        DAD     SP
    0678        +        XCHG
    0679        +        MOV     M,E
    0680        +        INX     H
    0681        +        MOV     M,D
    0682        +        POP     PSW
    0683        +        POP     D

0145     .....1 -        TXS
    0684        +        XCHG
    0685        +        SPHL
    0686        +        XCHG

0146     .....1 -        TSX
    0687        +        LXI     H,0
    0688        +        PUSH    PSW
    0689        +        DAD     SP
    0690        +        POP     PSW
    0691        +        XCHG

0147     .....2 -        BRA     B1
    0692        +        JMP     B1

0148     ....U2 -        BCC     B1
    0693        +        JNC     B1

0149     ....U2 -        BCS     B1
    0694        +        JC      B1

0150     ..U..2 -        BEQ     B1
    0695        +        JZ      B1

0151 NC  .U.U.2 -        BGE     B1
0152 NC  .UUU.2 -        BGT     B1
0153 NC  ..U.U2 -        BHI     B1
0154 NC  .UUU.2 -        BLE     B1
0155 NC  ..U.U2 -        BLS     B1
0156 NC  .U.U.2 -        BLT     B1
0157     .U...2 -        BMI     B1
    0696        +        JM      B1

0158     ..U..2 -        BNE     B1
    0697        +        JNZ     B1

0159 NC  ...U.2 -        BVC     B1
0160 NC  ...U.2 -        BVS     B1
0161     .U...2 -        BPL     B1
    0698        +        JP      B1

0162     .....2 -        BSR     B1
    0699        +        CALL    B1

0163     .....3 -        JMP     B1
    0700        +        JMP     B1

0164     .....2 -        JMP     B1,X
    0701        +        LXI     H,B1
    0702        +        PUSH    PSW
    0703        +        DAD     D
    0704        +        POP     PSW
    0705        +        PCHL

0165     .....3 -        JSR     B1
    0706        +        CALL    B1

0166     .....2 -        JSR     B1,X
    0707        +        LXI     H,$+11
    0708        +        PUSH    H
    0709        +        LXI     H,B1
    0710        +        PUSH    PSW
    0711        +        DAD     D
    0712        +        POP     PSW
    0713        +        PCHL

0167     .....1 -        NOP
    0714        +        NOP

0168 NC  VVVVV1 -        RTI
0169     .....1 -        RTS
    0715        +        RET

0170 NC  .....1 -        SWI
0171     .....1 -        WAI
    0716        +        HLT

0172     ....R1 -        CLC
    0717        +        STC
    0718        +        CMC

0173     .....1 -        CLI
    0719        +        EI
```

```
0174 NC ...R.1 -        CLV
0175    ....S1 -        SEC
    0720       +        STC

0176    .....1 -        SEI
    0721       +        DI

0177 NC ...S.1 -        SEV
0178    VVVVV1 -        TAP
    0722       +        MOV     L,B
    0723       +        MOV     H,A
    0724       +        PUSH    H
    0725       +        POP     PSW

0179    .....1 -        TPA
    0726       +        PUSH    PSW
    0727       +        POP     H
    0728       +        MOV     B,L

01800729        ;**
0181           -B1      FCB     5       * DECIMAL
    0730       +B1:     DB      5       ; DECIMAL

0182           -        FCB     $E5     * HEXA
    0731       +        DB      0E5H    ; HEXA

0183           -B2      FCB     @005    * OCTAL
    0732       +B2:     DB      0050    ; OCTAL

0184           -B3      FCB     %0101   * BINARY
    0733       +B3:     DB      0101B   ; BINARY

0185           -        FCB     'A      * CHARACTER
    0734       +        DB      'A'     ; CHARACTER

0186           -        FCC     /TEXT/
    0735       +        DB      'TEXT'

0187           -        FCC     9,TEXT
    0736       +        DB      'TEXT   '

0188           -LABEL   FDB     $1000
    0737       +LABEL:  DW      1000H

0189           -        FDB     LABEL
    0738       +        DW      LABEL

0190           -        FDB     LABEL+3
    0739       +        DW      LABEL+3

0191           -        FDB     *
    0740       +        DW      $

0192           -        FDB     *+13
    0741       +        DW      $+13

01930742        ;
0194           -        RMB     38
    0743       +        DS      38

01950744        ;**

0196           -TAG1    EQU     LABEL
    0745       +TAG1    EQU     LABEL

0197           -TAG2    EQU     LABEL-3
    0746       +TAG2    EQU     LABEL-3

0198           -TAG3    EQU     *-6
    0747       +TAG3    EQU     $-6

01990748        ;
0200           -        REG
    0749       +;** REG 6800 TO 8080 CONVERT
    0750       +;       A   --> B
```

```
    0751        +;          B    --> C
    0752        +;          X    --> D,E
    0753        +;          CCR  --> F
    0754        +;                   A,HL WORK REG
    0755        +!***---

02010756         ;
02020757                    END
```

添 付 資 料

# 2. 機 能 仕 様 書

# 目　　次

# 1. 目　　的

この機能仕様書は、8080 および 6800 の各ソースプログラムを相互に変換するに際し、種々の問題点や変換方法の基本概念等を明らかにすることを目的とする。

# 2. 8080と6800の相違

プログラムの取扱いにおいて問題となるハードウェアの相違および命令体系の相違について述べる。

(1) アドレス空間

8080 には、3個のアドレス空間が存在する。メモリ、I／O　レジスタの3個のアドレス空間である。一方、6800 にはメモリとレジスタの2個のアドレス空間が存在する。I／O はメモリ空間に共存する。

(2) レジスタ

プログラムで使用できるレジスタについて述べる。

8080 には、1個のスタックポインター（16ビット）、7個の8ビットレジスタ（A，B，C，D，E，H，L）およびフラグレジスタ(F)が存在する。アキュムレータはレジスタAのみである。また、BC，DE，HL はペアとして16ビットレジスタとして使用が可能である。

6800 には、1個のスタック・ポインター（16ビット）、2個の8ビット・アキュムレータ（A，B）、1個の16ビット・インデックスレジレタ(X)およびフラグレジスタ（CCR）が存在する。

(3) フラグ

8080 には、符号(S)、ゼロ(Z)、補助キャリー（AC）、パリティ(P)、キャリー(C)の各フラグが存在する。

6800 には、符号(N)、ゼロ(Z)、ハーフキャリー(H)、オーバーフロー(V)、キャリー(C)の各フラグが存在する。

6800 のフラグレジスタCCR内には割込許可・不許可用のフラグ(I)が存在し、セット・リセットできることの他にチェックも可能であるが、8080 においては、このようなフラグはフラグ・リジスタF内には存在せず、セット・リセットはできるがチェックすることは不可能である。

フラグの変化の仕方は、四則・論理演算においてある程度類似している以外、ほとんど一致しない。

(4) リセット時のスタート

8080 においては、0番地よりスタートする。したがって多くの場合、0番地にはROMになっている。RAMは通常ハイアドレスが割当てられる。

6800 においてはFFFE，FFFF番地の内容をアドレスとする地点よりスタートする。したがって多くの場合FFFF番地はROMになっている。0番地はダイレクトエリアとしてRAMが割当てられることが多い。

(5) スタック

レジスタとスタックの間のデータ転送は、8080 の場合は2バイト単位（AとF，BとC，DとE，HとL）であるが、6800 の場合は1バイト単位（A・B）である。

スタック　ポインターは、8080 の場合スタックの最上位バイトすなわち、スタックの最小アドレスをポイントしているが、6800 の場合は、8080 の場合より1小さい値となっている。

(6) 2バイトデータ

8080 においてはHLレジスタペア、6800 においてはXレジスタにおいて2バイト単位のデータ転送が可能である。$a$番地へストアした場合、8080 においては$a$番地へLの内容（下位側）が、$a+1$番地へはHの内容（上位側）が転送されるが、6800 においては$a$番地へXレジスタの上位側、$a+1$番地へは下位側が転送される。レジスタへのロードの場合も同様の関係が成り立つ。

また、3バイト命令の第2・第3バイトにおいても同様である。

(7) アドレッシング

8080 においてデータを取扱う際は、主にHLレジスタペアをポインターとして使用するレジスタ・インダイレクト・アドレッシングが中心となる。一方6800 においてはダイレクト・アドレッシングか、あるいはインデックス・アドレッシングが中心となる。

8080 においては、3個のレジスタ・ペアをポインターとして使用可能であるため、リエントラント・プログラムを作成できる場合が多いが、6800 においては、ポインターが1個であるため、リエントラント・プログラムを作成することはかなり困難である。

(8) 割　込

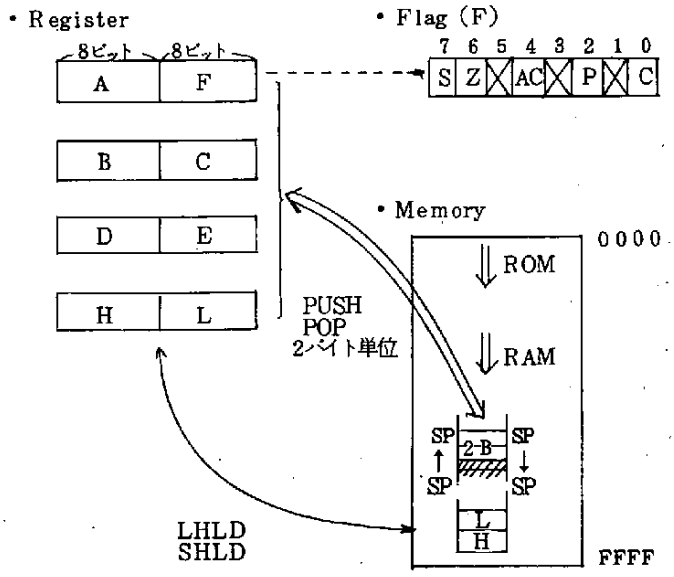
8080 の場合、割込が受付けられると、もどり番地のみがスタックに転送され、割込処理ルーチン内で使用されるレジスタはすべてプログラムでスタックへ転送しなければならない。割込処理を終了する場合には、スタックから該当するレジスタへ転送し、割込受付時の状況をプログラムにて復元しなければならない。また割込が発生した場合、自動的に割込不許可となるため、割込処理ルーチン内で割込許可命令を実行しなければならない。

6800 の場合、割込が受付けられると、すべてのレジスタの内容がスタックへ転送される。割込処理を終了する場合には、割込処理よりのリターン命令を実行することにより、割込受付時の状況へ完全に復元される。

(9) I/O 命令

すでに述べたように 6800 にはI/O アドレス空間が存在しないため、8080 のI/O アクセス命令に対応する命令は存在せず、通常のメモリ・アクセス命令によって入出力を実行する。

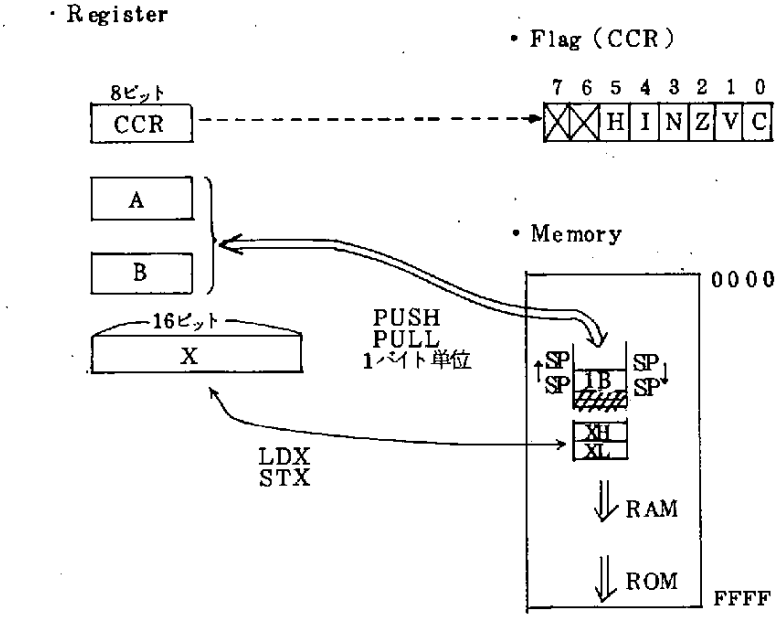
8080




8080と6800との相違

1. アドレス空間
2. レジスタの数と機能
3. フラグの種類と変化の仕方
4. リセット時のスタート
5. スタック操作(2バイト単位と1バイト単位)
6. 2バイトデータ(H, Lの反転)
7. アドレッシング
8. 割 込
9. I/O命令

6800



# 8080 プログラム例

```
        ORG     1000H
;*
        LXI     SP,WSP+128
;******---
        LXI     D,T1
        LXI     H,W1
        MVI     A,16
        CALL    AMOVE           ; TITLE SET
;*
        LDA     WCNT            ; COUNTER
        ORI     30H
        STA     W1+6
;***---
        HLT
;*******
T1:     DB      'TOTAL X GROUPS'
        DB      0DH,0AH
;*
W1:     DS      30
WCNT:   DB      8
WSP:    DS      128
;*
;*****************************************
;*      DATA AREA MOVE (REENTRANT)
;*
;*              DE=SND ADR
;*              HL=RSV ADR
;*              A=BYTE SIZE
;*      CALL    AMOVE   (A,BC SAVE)
;*              DE=SND.END+1
;*              HL=RSV.END+1
;*
AMOVE:  ORI     0
        JZ      AMOVE2
        PUSH    PSW
        PUSH    B
        MOV     B,A
;*
AMOVE1: LDAX    D
        INX     D
        MOV     M,A
        INX     H
        DCR     B
        JNZ     AMOVE1
;*
        POP     B
        POP     PSW
AMOVE2: RET
;****** ------
        END
```

# 6800 プログラム例

```
        ORG     $1000
*
        LDS     #WSP+127
****----
        LDX     #T1
        STX     BWRK1
        LDX     #W1
        STX     BWRK2
        LDAA    #16
        JSR     BMOVE           * TITLE SET
*
        LDAA    WCNT            * COUNTER
        ORAA    #$30
        STAA    W1+6
****----
        WAI
*******
T1      FCC     /TOTAL X GROUPS/
        FCB     $0D,$0A
*
W1      RMB     30
WCNT    FCB     8
WSP     RMB     128
*
******  B ROUTINE WORK AREA *******
BWRK1   RMB     2
BWRK2   RMB     2
BWRK3   RMB     2
******  ------
*****************************************
*       DATA AREA MOVE
*
*               BWRK1=SND ADR
*               BWRK2=RSV ADR
*               A=BYTE SIZE
*       JSR     BMOVE   (A,B SAVE)
*               BWRK1=SND.END+1
*               BWRK2=RSV.END+1
*
BMOVE   TSTA
        BEQ     BMOVE2
        PSHA
        PSHB
*
BMOVE1  LDX     BWRK1
        LDAB    0,X
        INX
        STX     BWRK1
        LDX     BWRK2
        STAB    0,X
        INX
        STX     BWRK2
        DECA
        BNE     BMOVE1
*
        PULB
        PULA
BMOVE2  RTS
******  ---
        END
```

# 3. 設定すべき事項

8080 と 6800 のプログラムを完全に双方向に変換することは、すでに述べたハードウェア上の違いによっても不可能である。ここでは、変換における限度等を明らかにし、変換時の各種制約等について記述する。

まず、プログラムの性質について考える。プログラムの性質としては、割込処理プログラム、Re-entrant 性などがあるが、割込の受付け方の違いや、レジスタの数の違いなどを考えると、プログラムの変換によってこれらの性質を保存できるという保証を与えることは実際上不可能である。

メモリ空間内におけるRAM／ROMの割当て方が 8080 と 6800 では異なることにより、変換においてはアドレスの値についてタッチしないこととする。すなわちＯＲＧ命令、ＥＱＵ命令などはそのまま採用するものとする。

次に各種フラグについてであるが、8080のＰ（パリティ）フラグと 6800のＶ（オーバーフロー）フラグは、互いに他に存在しないフラグである。この２つのフラグについては全く取扱わないものとする。すなわち、この２つのフラグに関する命令、たとえば条件付ジャンプ命令などは変換不可能なものとして取扱う。一方、Ｓ(N)(符号)、Ｚ(ゼロ)、Ｃ(キャリー)、AC(H)（ハーフキャリー）は両方に存在するフラグであり、機能もほとんど同じであるが、フラグの変化の仕方が大きく異っている。このため、フラグの変化の仕方を完全に保存して変換することは決して不可能ではないが、実際上そこまで行なっても、変換後のプログラム・サイズが大きくなる。多くの場合そこまで必要としない等の理由により、フラグの完全保存は考えないものとする。ただし、Ｃフラグだけは可能な限り保存するものとする。

入出力機器との入出力操作については、6800 にＩ／Ｏアドレス空間が存在せず、通常のメモリ・アクセスと同一命令で実行するようになっていることと、一般に入出力手順が標準化されていないことなどを考えて、Ｉ／Ｏ命令は変換不可能とする。

２バイト・データの高位バイトと低位バイトの関係が 8080 と 6800 とでは反対になっている。プログラムの変換においては、ターゲットとするＣＰＵの桁どりに従がう。すなわち、8080 のプログラムを 6800 用に変換した場合、得られた 6800 のプログラムにおいて上位と下位の関係は、元の 8080 のプログラムと反転したものとなっている。

以上をまとめると、次のようになる。

(1) プログラムの性質の保存は保証しない。

(2) フラグの完全変換は行なわない。ただし、キャリー・フラグのみは可能な限り完全変換を目指す。

(3) Ｉ／Ｏ命令の変換は行なわない。

(4) 数字データの桁どりは、ターゲットＣＰＵの方法を採用する。

# 4. 変換方式

## 4.1 レジスタの対応づけ

8080 と 6800 では、レジスタの数と、それぞれの機能が異なるため、変換の際にはレジスタの対応づけが問題となる。

(a) 8080 → 6800 の場合

レジスタ

A → レジスタA
B → メモリ
C → 〃
D → 〃
E → 〃
H → 〃
L → 〃
F → レジスタCCR

レジスタ B
〃 X
} ワーク用

可能な限りレジスタ同士を対応づける方がよいが、どうしてもワーク用のレジスタが必要であることと、10進補正命令はレジスタAにのみ有効であることから、レジスタA同士を対応づけることとする。レジスタHLはメモリ・ポインターとしてよく利用されるため、インデックスレジスタXと対応づけることが考えられるが、レジスタHとLは単独でも使用されること、レジスタペアBC, DEもポインタとして使用されることがあることなどから、レジスタXもワーク用とする。

ポインタを1バイト毎にセットする場合は注意を要ずる。たとえば、エリアPNTに、あるエリアのアドレスがセットされている場合、8080 のプログラムでは例えば次のようになる。

```
LXI  H, PNT
MOV  E, M
INX  H
MOV  D, M
```

この場合、当然データがアドレスの小さい方からLow, Highとなっているため、ポインタDEのレジスタEの方にLow 側がロードされるようになっている。これをそのまま6800 のプログラムに変換した時には、ポインタのLow, Highが入れかわってしまうことがある。

これは6800 から 8080 への変換の時にも起り得るので、やはり十分な注意が必要である。

(b) 6800 → 8080 の場合

レジスタ

A → レジスタB
B → 〃 C
X → 〃 DE
CCR → 〃 F

レジスタA
〃 H ｝ ワーク用
〃 L

8080のAレジスタは唯一のアキュムレータであるため、これはワーク用としなければならない。Xレジスタは通常インデックス・アドレッシングとして使用され、その際オフセットとXレジスタの内容を加算して実アドレスを決定する。この時、Xレジスタに変化はない。これよりXレジスタはDEレジスタペアに対応づけ、インデックス・アドレッシングに対しては、オフセットをHLレジスタペアにセットし、DAD D命令によって実アドレスをHLに求めることができる。

## 4.2 フラグの取扱い方

すでに述べたように、フラグをすべて保存して変換することはしない。ただし、キャリーフラグのみは可能な限り保存することとする。

## 4.3 自己相対、ラベル相対の取扱い方

自己相対（$\overset{\$}{*}\pm i$）とブランチ命令のラベル相対（Label $\pm i$）は新しいラベルを付けて対処する。

## 4.4 リテラルの取扱い方

8080においては、数字データの直後に基数を表わす英文字を1字付加し、Bによって2進数を、OまたはQによって8進数をDによって10進数を（Dはなくてもよい）、Hによって16進数を表現している。

一方6800においては、数字データの直前に基数を表わす英文字を1文字付加し（10進数の場合は何もつけない）、%によって2進数を、@によって8進数を、$によって16進数を表現している。

プログラム変換の際、ターゲットフォーマットに合わせる。

## 4.5 2バイト・データの取扱い方

8080と6800においては、すでに述べたように、2バイト・データの上位桁バイトと下位桁バイトが反転した関係になっている。2バイト・データの転送や2バイト・データの定義（DW命令、FCD命令）を優先的に考えることとし、したがって、ターゲットCPUの表現方法を採用することとする。このため、2バイト・データを1バイト単位で取扱う場合、必ずしも数字表現方法が正確に

変換されないことが起り得るので注意を要する。（特にアドレスに関しては注意を要する。）

### 4.6　命令の変換

プログラムの変換において、変換されるプログラムの各機械語命令は、ターゲットＣＰＵのいくつかの命令の集まりとみなすことにする。すなわち、マクロ命令とみなすことにする。このためには、マクロ・プロセッサが必要になる。

## 5. マクロ・プロセッサ

変換されるプログラムの各命令をマクロ命令とみなして、プログラム変換を行なうこととしたが、その理由は次に述べる通りである。

まず第一の理由は、変換用のテーブル（マクロ・ライブラリ）をフレキシブルに取扱えるようにしたいことである。変換される命令のオペランドは、各命令によって、その個数、使われ方等まちまちであり、また同一命令であっても、アドレッシング・モードによって異なる形態となるからである。また、デバッグ段階において、プログラムの変更よりはテーブルの変更の方が容易でかつ迅速になるからである。さらに、他マシン（たとえばＺ－８０）への応用の際にもテーブルの作成だけで変換が可能となるからである。

その他の主要な理由としては、既に登録されている標準マクロを自由に使用できるようにしたいことである。データのブロック転送、クリア、数バイト・データの演算・比較などは基本プログラムとして日常よく使用可能なものであるが、実際の使用の際に容易にリンケージがとれるようにしておかなければ現実には使用できないものである。

以上のような理由により、マクロ・プロセッサを必要とする。それでは、現在販売されているマクロ・プロセッサで十分かどうかということであるが、結論を述べると、現在販売されているマクロ・プロセッサでは不十分であり、もっと機能の高いマクロ・プロセッサが必要である。その理由としては、まず第一に、既に登録されている標準マクロをリンケージする機能が全くないことである。さらに、文字列の取扱い機能が低いことや、オペランドの個数をデータとして取扱えないことなどがあげられる。

# 添 付 資 料

## 3. 変換ライブラリ説明書

# 目　　次

# 1. 概　要

本説明書は、8080—6800 双方向ソース・プログラム・コンバータにおける各命令の変換を定めるファイル（ライブラリと言う）について述べる。ライブラリは2種類用意してあり、1つは8080ソースプログラムを6800ソース・プログラムへ変換する際に使用され、他のライブラリは6800ソース・プログラムを8080ソース・プログラムへ変換する際に使用される。それぞれのライブラリには、変換後の冗長命令を除去するための最適化の機能が織り込まれており、ライブラリを大きく複雑にしている。

まず最初に、この2つのライブラリに共通する事項および基本的な事項について、ライブラリ構造として説明をする。次に個々のライブラリについての内容や最適化の方法等について説明をする。

さらに、他のＣＰＵ言語、たとえば6800のソース・プログラムをＺ８０のソース・プラグラムへ変換する場合などについて解説する。

# 2. ライブラリ構造

ライブラリはステートメントにより構成される。ステートメントにはコメント・ステートメント、実行ステートメント、展開ステートメントがある。

コメント・ステートメントは＊に始まる１行であり、コメント以外の機能は持っていない。

実行ステートメントには、％に始まる行、あるいはMACRO：に始まる行、さらにMEND( MENDの前にはタブが必要）がある。この実行ステートメントは、ソース・プログラムの変換時に実行される。％に始まる行は左辺の変数に右辺の値（多くの場合は 'に囲まれた文字定数）を代入する場合、ＩＦ～GOTO～によって条件チェックし、条件を満足すれば指定したラベルの場所へジャンプする場合、NOPによって何も実行しない場合、さらにCALLによってサブルーチンをコールする場合等が用意されている。MACRO：に始まるステートメントは例１に示されるINXのように、展開されるべき命令を定義しており、MENDのステートメントまで１つのブロック（マクロ・ボディーという）となる。展開されるべき命令がオペランドを持つ場合には、例１におけるＢ１のように文字Ｂで始まる変数を使用することができる。オペランドが２個の場合にはＢ１，Ｂ２のように定義すればよく、３個以上の場合も同様である。このようにして定義した変数Ｂ１やＢ２はローカルすなわち、定義されているマクロ・ボディー内でのみ有効である。％と：の間に置かれ、文字Ｌで始まる文字列をラベルといい、GOTO の後に書いて飛び先として使用することができる。ラベルもまたローカルである。同一ラベルを複数個、同一マクロ・ボディー内で定義してはならない。

```
MACRO:  INX     B1
%       BSFLG='.....1'
%       IF B1.NE.'SP' GOTO L11
        INS
%       EXIT
%L11:   NOP
%       IF B1.EQ.BMFA GOTO L12
        LDX     M%B1
%L12:   BMFA=B1
        INX
        STX     M%B1
        MEND
```

例　1

実行ステートメントにおいても＊で始まるコメントを書くことが可能である。

MENDはマクロ・ボディーの終了を意味すると同時に、実行の終了も意味している。EXIT は実行の終了だけを意味しており、展開ステートメントによる展開が終了する。ＥＸＩＴはMEND と異なり第１文字に％が必要である。

コメント・ステートメントでも実行ステートメントでもないステートメントを展開ステートメントという。展開ステートメントは、MACRO：ステートメントとMENDステートメントの間で％以外の文字で始まるステートメントである。変換実行時に展

```
.....1  -       INX     B
        +       LDX     MB
        +       INX
        +       STX     MB
.....1  -       INX     D
        +       LDX     MD
        +       INX
        +       STX     MD
.....1  -       INX     H
        +       LDX     MH
        +       INX
        +       STX     MH
```

例　2

開ステートメントに実行が移ってくると、展開ステートメントは変換された命令として出力ファイルへ出力される。例2においては、変換される命令は－で、変換された命令は＋で示されている。展開ステートメントは、変換された命令を構成する機能を持っており、出力ファイルに出力される際には＋で示される。

例1、例2でもわかるように、オペランドに指定した変数Ｂ１は＆Ｂ１として使用することにより、Ｂ１に割当てられた文字列がそのまま展開されることになる。たとえばＩＮＸ Ｄの場合、変数Ｂ１には文字列Ｄに割当てられるからＭ＆Ｂ１の部分はＭＤとして展開されることになる。

全く展開ステートメントを通過せずにＥＸＩＴまたはＭＥＮＤ を実行することは許されていない。何も展開する必要がない場合は例３のように※の部分を用意することにより、何も展開せずに終了することができる。したがって例３の展開ステートメントは全く展開されないことになる。

```
MACRO:  CPE     B1
%       BSFLG='...U.3'
%       BSERR='NC'
%       BMFA=' '
%       BSFWREC=$0D  } ※
        NOP
        MEND
```

例　3

実行ステートメントのＣＡＬＬ によってコールされるサブルーチンは例４のようになっている。サブルーチンであることを定義する ＳＵＢ ：の後にサブルーチン名を指定し、最後は ＲＥＴとする。例４でもわかるように通常のマクロ・ボディーにおける実行ステートメントと展開ステートメントを区別する％の使い方が逆になっていることに注意しなければならない。すなわち、第１文字が％ならば展開ステートメント、％でないならば実行ステートメントとなる。

```
SUB:    MOVAB
        IF BMF4.EQ.'B' GOTO L4
%       MOV     A,B
L4:     RET
        ENDS
```

例　4

サブルーチンにおけるラベルもローカルである。

ライブラリに使用される定義済み変数としては、次のものがある。

(1)　ＢＳＦＬＧ

６バイトの文字列データをセットすべきエリアであり、例２の一記号の左側に表示される。8080 → 6800変換ライブラリにおいては、ＣＺＳＰＡ％、すなわち左側よりキャリー・フラグ、ゼロ・フラグ、符号フラグ、パリティ・フラグ、補助キャリー、命令バイト数を示している。6800→8080 変換ライブラリにおいては、ＨＮＺＶＣ％、すなわち左側より補助キャリー、符号フラグ、ゼロ・フラグ、オーバー・フロー・フラグ、キャリー・フラグ、命令バイト数を示している。フラグの表示は、・の時は不変、Ｖは変化、Ｒは０にリセット、Ｓは１にセットを意味する。またＵは条件付きジャンプ命令などで使用されるフラグを示しており、一種のウォーニングである。

(2)　ＢＳＥＲＲ

２バイトの文字列データをセットすべきエリアであり、エラーコード表示用に使用される。

(3) ＢＳＦＷＲＥＣ

ＢＳＦＷＲＥＣ＝＄０Ｄ として使用され、次の展開ステートメントの無視を指示する。

(4) ＢＭＦ０，ＢＭＦ１，ＢＭＦ２，ＢＭＦ４，ＢＫＦ１，ＢＫＦ２

6800 ソース・プログラムを 8080 ソース・プログラムへ変換する際の最適化に使用される。

マクロ・ボディーによる命令の変換には、キャリー・フラグの保存が考慮されているため、場合によっては余分な命令が展開されることになる。

ライブラリの内容はＣＰ／Ｍのエディターによって自由に変更することが可能である。また、変換されて得られたファイルも全て同様にエディターによって変更が可能である。

## 3. 8080 →6800 変換ライブラリ

### 1 レジスタ対応

まずレジスタの対応づけであるが、6800 の方がレジスタ数が少ないため、8080 のレジスタの一部は 6800 のメモリに対応づけが必要となる。また 6800 の方にワーク用のレジスタが必要であること、１０進補正命令はレジスタＡにのみ有効であることより、8080 のレジスタＡを 6800 のレジスタＡに対応づけ、6800 のレジスタＢおよびインデックス・レジスタＩＸはワーク用とする。したがって、レジスタの対応は次のようになる。

| 8080 のレジスタ | | 6800 |
|---|---|---|
| A | → | レジスタ A |
| B | → | メモリ （MB） |
| C | → | メモリ （MC） |
| D | → | メモリ （MD） |
| E | → | メモリ （ME） |
| H | → | メモリ （MH） |
| L | → | メモリ （ML） |
| F | → | レジスタ CCR |

レジスタＢ
インデックス・レジスタＩＸ
はワーク用

（Ｆ：フラグ・レジスタ
ＣＣＲ：コンディション・コード・レジスタ）

## 2. 最適化処理

定義済み変数BMFA を用いて最適化を行う。BMFA は、ワーク用のインデックス・レジスタＩＸが 8080 のどのレジスタ・ペアに等しくなっているかどうかを対応づけている。BMFA が文字Ｂであれば、ＢＣレジスタ・ペアに対応しているエリアＭＢとＭＣの内容に等しいことを示している。同様にBMFA は文字Ｄ，Ｈ，その他の値を取る。文字Ｄの時はＤＥレジスタ・ペア、文字Ｈの時はＨＬレジスタ・ペアに対応する。その他の時には、どのレジスタ・ペアにも等しくないことを示している。変換処理が開始された時と、変換される命令にラベルが定義されている場合には強制的にスペース・コードがセットされる。

その他インデックス・レジスタＩＸの対応の変化はすべてライブラリの中で管理している。たとえば、ＩＸがＭＢ，ＭＣに等しい時には、ＭＢ，ＭＣよりＩＸへの転送命令は展開されないようになっている。また、この場合、ＭＢまたはＭＣが変化した時にはＩＸとＭＢ，ＭＣが等しくないようになるため、BMFA はライブラリ内で、スペース・コードがセットされている。同様にＩＸが他の値に変化した場合にも、ライブラリ内でBMFA にスペースがセットされる。

基本的にBMFA はライブラリ内で管理されているため、最適化はいくつかの命令に渡って、あるいはいくつかの命令をへだてて実行されることがある。一方、アセンブラ命令の中でエリアを確保したりする場合には、BMFA はスペース・コードにセットされる。

注目すべき命令は変換後の命令、すなわち(A)の欄に一記号が付されている命令のみである。

(1)によってＩＸとＭＨ，ＭＬが等しくなる。(2)においてはＩＸ，ＭＨ，ＭＬに変化がないので(3)は不要となる。同様に(4)においてＩＸ，ＭＨ，ＭＬに変化がないので(5)は不要である。

(6)では、ＭＨ，ＭＬ（ＭＨ＋１で表わされている）が変化し、ＩＸとＭＨ，ＭＬとが等しいという関係がくずれてしまう。したがって(7)は必要となる。

(8)によってＩＸとＭＨ，ＭＬとが等しくなる。(9)でこの等しい関係はくずれないから、(10)は不要となる。

(11)によってＩＸとＭＢ，ＭＣとが等しくなる。そのため(12)は不要と思われるが、ラベルＬＢＬ１が定義されているため(12)を取り除くことはできない。

最適化の結果によって(3)、(5)、(10)が取り除かれていることが確認できる。

## 最適化の例

最適化を行ってない場合

```
.....3 -        LXI     H,B1
       +        LDX     #B1
       +        STX     MH ------- (1)

.....1 -        MOV     B,L
       +        LDAB    ML  }---- (2)
       +        STAB    MB  }

.....3 -        SHLD    B1
       +        LDX     MH ------- (3)
       +        STX     B1

.....1 -        PUSH    H
       +        LDAB    MH
       +        PSHB            }---- (4)
       +        LDAB    MH+1    }
       +        PSHB

.....1 -        INX     H
       +        LDX     MH ------- (5)
       +        INX
       +        STX     MH

.....1 -        POP     H
       +        PULB
       +        STAB    MH+1    }---- (6)
       +        PULB            }
       +        STAB    MH      }

.....1 -        INX     H
       +        LDX     MH-------- (7)
       +        INX
       +        STX     MH-------- (8)

VVVVV1 -        DAA
       +        DAA ---------------- (9)

.....1 -        DCX     H
       +        LDX     MH ------- (00)
       +        DEX
       +        STX     MH

.....3 -        LXI     B,B1
       +        LDX     #B1
       +        STX     MB ------- (01)

.....1 -LBL1:   STAX    B
       +LBL1    LDX     MB-------- (02)
       +        STAA    0,X
```

(A)

最適化を行った場合

```
.....3 -        LXI     H,B1
       +        LDX     #B1
       +        STX     MH-------- (1)

.....1 -        MOV     B,L
       +        LDAB    ML  }------ (2)
       +        STAB    MB  }

.....3 -        SHLD    B1
       +        STX     B1

.....1 -        PUSH    H
       +        LDAB    MH
       +        PSHB            }---- (4)
       +        LDAB    MH+1    }
       +        PSHB

.....1 -        INX     H
       +        INX
       +        STX     MH

.....1 -        POP     H
       +        PULB
       +        STAB    MH+1    }------ (6)
       +        PULB            }
       +        STAB    MH      }

.....1 -        INX     H
       +        LDX     MH-------- (7)
       +        INX
       +        STX     MH-------- (8)

VVVVV1 -        DAA
       +        DAA ---------------- (9)

.....1 -        DCX     H
       +        DEX
       +        STX     MH

.....3 -        LXI     B,B1
       +        LDX     #B1
       +        STX     MB-------- (01)

.....1 -LBL1:   STAX    B
       +LBL1    LDX     MB-------- (02)
       +        STAA    0,X
```

(3), (5), (00)は最適化によって取除かれている。

# 4. 6800→8080変換ライブラリ

## 1. レジスタ対応

6800のレジスタの数より8080のレジスタの数の方が多いため、すべてレジスタ対応が可能である。8080ではレジスタAは唯一のアキュムレータである。したがって、8080のレジスタAはワーク用とする必要がある。6800のインデックス・レジスタＩＸはインデックス・モードに使用される。すなわち、指定されたオフセット値とＩＸの内容を加算して実アドレスを求め、求まったアドレスのデータを取扱う。この場合ＩＸの値は変化しない。このようなことより、ＩＸは8080のＤＥレジスタ・ペアに対応をつけ、ＨＬレジスタ・ペアはワーク用とする。このようにすると、オフセット値をＨＬレジスタ・ペアにセットし、ＨＬレジスタ・ペアとＤＥレジスタ・ペアの加算命令により実アドレスをＨＬレジスタ・ペアに求めることができ、ＤＥレジスタ・ペアの内容は変化しない。

| 6800のレジスタ | | 8080 |
|---|---|---|
| A | → | レジスタ B |
| B | → | レジスタ C |
| IX | → | レジスタ D, E |
| CCR | → | レジスタ F |
| | | レジスタ A<br>レジスタ H<br>レジスタ L<br>はワーク用 |

## 2. 最適化処理

6800 から 8080 への変換における最適化は２種類行われている。１つは、6800 のインデックス・モードを 8080 へ変換する際の実アドレスの計算に対してであり、他の１つは 8080 のレジスタＡへの転送命令に対してである。

まず、インデックス・モードの変換における最適化は、連続する同じオフセット値を持つ命令の２番目以降における実アドレスの計算部分を取り除いている。インデックス・レジスタＩＸに対応するＤＥレジスタ・ペアおよび実アドレスが求められるワーク用のＨＬレジスタ・ペアの変化を管理することが困難であるため、連続するインデックス・モードの変換に対してのみ最適化を行っている。

定義済み変数ＢＭＦ０ には１つ前に変換した 6800 命令のアドレッシング・モードがセットされている。16 進で０３（＄０３）の時には１つ前に変換した 6800 の命令はインデックス・モードであったことがわかるようになっている。BMF1は１つ前の6800命令がインデックス・モードであった場合、オフセットの指定が数字であったことを示す。＄００の時には数字であったことを、その他の場合、オフセット値が数字以外のデータで指定されたことを示している。最適化はＢＭＦ１が＄００の時のみ、すなわちオフセット値が数字データで与えられた場合にのみ適用されるようになっている。ＢＭＦ２は、１つ前の 6800命令がインデックス・モードであった場合のオフセット値を示している。ＢＫＦ１ はこれから変換しようとしている 6800 のインデックス・モードの命令のオフセット値が数字データで与えられているかどうかを示しており、ＢＭＦ１ と同じ役割を果している。ＢＫＦ２ はこれから変換しようとしている 6800 のインデックス・モードの命令のオフセット値を示しており、ＢＭＦ２ と同じ役割を果している。

インデックス・モードの変換における最適化はライブラリ内のサブルーチンＧＥＴＡＤ１およびＧＥＴＡＤ２で実行されている。ＧＥＴＡＤ１とＧＥＴＡＤ２ の違いは、実アドレスを求める際にキャリーを保存するかどうかであり、ＧＥＴＡＤ２ではキャリー保存のためのＰＵＳＨ　ＰＳＷとＰＯＰ ＰＳＷ が余分に必要となっている。

まず、ＢＭＦ０が＄０３であるかどうか、すなわち、１つ前の変換が今回と同じインデックス・モードであったかどうかをチェックする。ＢＭＦ０が＄０３でない、すなわち、１つ前の変換がインデックス・モードでなかったならば、実アドレスの算出部分の展開へ進む。ＢＭＦ０が＄０３に等しい時は、ＢＫＦ１およびＢＭＦ１がともに＄００、すなわちオフセット値が有効かどうかをチェックし、有効でないならば実アドレスの算出部分の展開へ進む。有効であった場合は、ＢＫＦ２とＢＭＦ２が等しいかどうか、すなわちオフセット値が等しいかどうかをチェックする。等しい場合は実アドレスの算出部分をスキップし、等しくない場合は実アドレスの算出部を展開する。

なお、今変換しようとしている 6800 の命令のアドレッシング・モードは、システムサブルーチンＧＯ６８ によって与えられる。このＧＯ 68 は４個の引数（この引数はすべてローカルなラベルでなければならない）を必要とし、第１引数はオペランドを持たない命令を処理する場合のジャンプ先であり、第２引数はイクステンドあるいはダイレクト・モードの命令を処理する場合のジャンプ

先である。第3引数は即値モードの命令を処理する場合のジャンプ先であり、第4引数はインデックス・モードの命令を処理する場合のジャンプ先である。なお引数が定められていない場合、もしそのモードが指定されるとBSERRに'ER'がセットされて、そのマクロ・ボディーの実行は終了する。また、レジスタAあるいはBの指定は命令に含めることとする。すなわちLDA A #1とLDA B #2はそれぞれLDAA, LDABと異なる命令として処理される。したがって、CLR A はオペランドを持たない命令と解釈される。

このようにGO68を用いることにより、現在処理中の命令がインデックス・モードであるかどうかは容易に理解できるようになっている。

インデックス・モード変換における最適化の例

最適化を行なわない場合

```
VVVVV2  -       ADDA    S,X
        +       LXI     H,S     ----- (1)
        +       DAD     D
        +       MOV     A,M
        +       ADD     B
        +       MOV     B,A

VVVVV2  -       ADDB    S,X
        +       LXI     H,S     ----- (2)
        +       DAD     D
        +       MOV     A,M
        +       ADD     C
        +       MOV     C,A

VVVVV2  -       ADCA    S,X
        +       LXI     H,S     ----- (3)
        +       PUSH    PSW
        +       DAD     D
        +       POP     PSW
        +       MOV     A,M
        +       ADC     B
        +       MOV     B,A

VVVVV2  -       ADCB    S,X
        +       LXI     H,S     ----- (4)
        +       PUSH    PSW
        +       DAD     D
        +       POP     PSW
        +       MOV     A,M
        +       ADC     C
        +       MOV     C,A

.RSRR2  -       CLR     S,X
        +       LXI     H,S     ----- (5)
        +       DAD     D
        +       XRA     A
        +       MOV     M,A

.VVVV2  -       CMPA    S,X
        +       LXI     H,S     ----- (6)
        +       DAD     D
        +       MOV     A,B
        +       CMP     M

VVVVV2  -       ADDA    4,X
        +       LXI     H,4     ----- (7)
        +       DAD     D
        +       MOV     A,M
        +       ADD     B
        +       MOV     B,A
```

(1)によって実アドレスが算出される。(2), (3), (4), (5), (6)では(1)と全く同じアドレスが算出されるため、この部分はすべて取り除くことが可能である。しかし(7)の場合、オフセット値が4であり、(1)～(6)の8とは異なるため(7)は取り除かれることはない。

最適化を行った場合

```
VVVVV2 -        ADDA    8,X
       +        LXI     H,8     }---- (1)
       +        DAD     D       }
       +        MOV     A,M
       +        ADD     B
       +        MOV     B,A

VVVVV2 -        ADDB    8,X
       +        MOV     A,M
       +        ADD     C
       +        MOV     C,A

VVVVV2 -        ADCA    8,X
       +        MOV     A,M
       +        ADC     B
       +        MOV     B,A

VVVVV2 -        ADCB    8,X
       +        MOV     A,M
       +        ADC     C
       +        MOV     C,A

.RSRR2 -        CLR     8,X
       +        XRA     A
       +        MOV     M,A

.VVVV2 -        CMPA    8,X
       +        MOV     A,B
       +        CMP     M

VVVVV2 -        ADDA    4,X
       +        LXI     H,4     }---- (7)
       +        DAD     D       }
       +        MOV     A,M
       +        ADD     B
       +        MOV     B,A
```

最適化を行うと(2)～(6)の部分が削除される。しかし(7)の部分は確かに残っていることがわかる。

6800 から 8080 への変換における最適化には、インデックス・モードの変換における場合の他に、ワーク・レジスタである 8080 のレジスタAの管理がある。レジスタAの内容が他のどのレジスタに等しいかどうかをBMF4 によって、チェックできるようにライブラリで使用している。すべての命令の変換を調べた結果より、レジスタBあるいはレジスタCよりレジスタAへ転送する命令を取り除くかどうかをチェックすればよい。この機能はライブラリ内のサブルーチンMOVAB（例4）およびMOVACにおいて実行される。この最適化の考え方および実現方法は 8080 から 6800 への変換における最適化と同様である。

最適化の例

最適化を行っていない場合

```
.VVVV1  -       ROLB
        +       MOV     A,C
        +       RAL
        +       MOV     C,A ------ (1)

.VVVV3  -       CMPB    MEN
        +       LXI     H,MEN
        +       MOV     A,C ------ (2)
        +       CMP     M

.VVRS1  -       COMB
        +       MOV     A,C ------ (3)
        +       CMA
        +       MOV     C,A ------ (4)
        +       STC

.VVRR1  -       TSTB
        +       MOV     A,C ------ (5)
        +       CPI     0

.VVRS1  -       COMB
        +       MOV     A,C ------ (6)
        +       CMA
        +       MOV     C,A ------ (7)
        +       STC

.VVR.3  -       STAB    MEN
        +       MOV     A,E ------ (8)
        +       STA     MEN
```

(1)によってレジスタCへレジスタAの内容を転送する。したがって(2)～(3)は不要である。同様に(4)によって(5), (6)が、(7)によって(8)が不要である。

最適化を行った場合

```
.VVVV1  -       ROLB
        +       MOV     A,C
        +       RAL
        +       MOV     C,A ------ (1)

.VVVV3  -       CMPB    MEN
        +       LXI     H,MEN
        +       CMP     M

.VVRS1  -       COMB
        +       CMA
        +       MOV     C,A ------ (4)
        +       STC

.VVRR1  -       TSTB
        +       CPI     0

.VVRS1  -       COMB
        +       CMA
        +       MOV     C,A ------ (7)
        +       STC

.VVR.3  -       STAB    MEN
        +       STA     MEN
```

6800から8080への変換における最適化に使用される定義済み変数はすべて、変換処理が開始された時と、変換される命令にラベルが定義されている場合には、強制的にスペース・コードがセットされる。

# 5. 他ＣＰＵ言語への応用

## 1. 他ＣＰＵ言語間の変換

今回は8080と6800との間のソースプログラム変換を行ったが、他ＣＰＵとのソースプログラム変換も実現可能である。たとえばＺ－80と6809との間のソースプログラム変換なども当然実現可能である。そのためには、

(1) 変換ライブラリを作製する。

(2) 変換を実行するプログラムを一部変更する（実行プログラムとライブラリとのリンクを変更）。

最適化については、変換のケースに応じて行う必要があり、実行プログラムとライブラリの両方で実現してゆかなければならない。

Ｚ－80の命令を6800の命令に変換する1例を与える。この場合最適化は考慮しておらず、またＺ－80はＺilog社の表現方法とする。

1例として命令ＳＢＣのマクロ・ボディを作ってみる。この命令は、レジスタAに関する1バイト処理の場合と、H，Lレジスタ・ペアに関する2バイト処理の場合がある。この区別は第1オペランドで可能である。

まずレジスタの対応であるが、Ｚ－80のレジスタAは6800のレジスタAに対応し、Ｚ－80のその他のレジスタは6800のメモリに対応しているものとする。たとえばレジスタBはメモリMBに、レジスタCはメモリＭＣに（ただし、ＭＢ、ＭＣと連続してメモリ上に定義されなければならない）対応しているものとする。

Z－80　SBCのマクロ・ボディの例

```
MACRO : SBC   B1,B2                          ①
%       IF  B1. EQ. 'HL' GOTO  L4            ②
%       BSFLG='VVVVV1'                       ③
%       IF  B2. NE. '(HL)' GOTO  L2          ④
        LDX   MH                             ⑤
        SBCA  0, X                           ⑤
%       EXIT                                 ⑥
%L2:    IF  B2. EQ. 'A' GOTO  L3             ⑦
        SBCA  M&B2                           ⑧
%       EXiT                                 ⑨
%L3:    NOP                                  ⑩
        PSHA                                 ⑪
        TSX                                  ⑪
        SBCA  0.X                            ⑪
        PULB                                 ⑪
%       EXIT                                 ⑫
%L4:    BSFLG='VVVVV2'                       ⑬
        LDAB  ML                             ⑭
        SBCB  M&B2+1                         ⑭
        STAB  ML                             ⑭
        LDAB  MH                             ⑭
        SBCB  M&B2                           ⑭
        STAB  MH                             ⑭
        MEND                                 ⑮
```

① SBCに関するマクロ・ボディの開始を宣言している。SBCのオペランドは最大2個であり、B1，B2とする。

② 第1オペランドB1が文字列HLであるならば、H，Lレジスタ・ペアに関する2バイト処理である。この場合はL4以降で変換される。

③ これ以降L4の直前までは、レジスタAに関する1バイト処理とみなしている。1バイト処理の場合のフラグの変化の様子と命令のバイト数を指定している。

なお、B1が文字Aであったかどうかの確認をとりたければ③の部分は次のようにしてもよい。

```
%       IF  B1. EQ. 'A' GOTO  L1
%       BSERR = 'ER'        エラー表示する
```

```
%       BSFWREC = $0D
        NOP                    } 何も展開せずに終了する
%       EXIT
%L1 :   BSFLG = 'VVVVV1'
```

④　SBC A,(HL)　であるかどうかをチェック

⑤　SBC A,(HL)　を変換

⑥　変換終了

⑦　SBC A,A　であるかどうかをチェック

⑧⑨　SBC A,r(r≠A)を変換

⑩⑪⑫　SBC A,A　を変換

⑬　SBC HL,dd　の場合(2バイト処理の場合)のフラグの変化およびバイト数を指定する。

⑭　SBC HL,dd　を変換

⑮　変換終了とマクロ・ボディの終了を意味する。

なお、⑧、⑭の場合に、第2オペランドのチェックを十分には行っていないが、完全なチェックをすることは可能である。

## 2. マクロ・プロセッサ

これまで述べてきたように、異機種CPU間のプログムラ変換に使用するだけでなく、同一CPUにおけるマクロ展開にも使用が可能である。

たとえば、データのブロック転送を行うマクロの例を与える。転送すべきデータの先頭アドレス、転送先のアドレス、転送するデータのバイト数をオペランドとして与えることにする。マクロの名前はAMOVE とする。このマクロの中ではAMOVE という名前のサブルーチンがコールされている。したがって、サブルーチンAMOVE をソース・プログラム中に生成するには、元のソース・プログラム中の適当な所で、MACG SUB により、マクロ内で使用したサブルーチンの出力依頼を行なわなければならない。

マクロを書く時にオペランドの省略を許すかどうかは、マクロ・ライブラリ次第である。オペランドが省略された場合には、対応する変数にはスペース・コードがセットされる。IF～によってオペランド変数が、スペース・コードであるかどうかをチェックすることにより、オペランドが省略されているかどうかを知ることができる。

# 6. 参 考 資 料

マクロ・ライブラリの例

```
 1: ********************************************
 2: *                                          *
 3: *   8080 MACRO LIBRARY  F=I-MACRO.LIB      *
 4: *                                          *
 5: ********************************************
 6: *
 7: *
 8: MACRO:   AMOVE    BS,BR,BN        ～マクロの宣言
 9: %        IF BS.EQ.' ' GOTO L1    ～第１オペランドが指定してなければ
10:         LXI      D,&BS             Ｌ１へ
11: %L1:    IF BR.EQ.' ' GOTO L2    ～第２オペランドが指定してなければ
12:         LXI      H,&BR             Ｌ２へ
13: %L2:    IF BN.EQ.' ' GOTO L3    ～第３オペランドが指定してなければ
14:         MVI      A,&BN             Ｌ３へ
15: %L3:    NOP
16:         CALL     AMOVE           ～マクロ内でのサブルーチン・コール
17:         MEND                     ～マクロの終了
18: *
19: *
20: MACSUB: AMOVE                    ～マクロ内でコールされるサブルーチンの
21: ;*********************************************  宣言
22: ;*      DATA AREA MOVE (REENTRANT)
23: ;*
24: ;*              DE=SND ADR
25: ;*              HL=RSV ADR
26: ;*              A=BYTE SIZE
27: ;*      CALL    AMOVE   (A,BC SAVE)
28: ;*              DE=SND.END+1
29: ;*              HL=RSV.END+1
30: ;*
31: AMOVE:  ORI      0
32:         JZ       AMOVE2
33:         PUSH     PSW
34:         PUSH     B
35:         MOV      B,A
36: ;*
37: AMOVE1: LDAX     D
38:         INX      D
39:         MOV      M,A
40:         INX      H
41:         DCR      B
42:         JNZ      AMOVE1
43: ;*
44:         POP      B
45:         POP      PSW
46: AMOVE2: RET
47: ;******* ------
48:         MEND
49:         END
```

展開前のソース・プログラム

```
 1: ;*********************************************
 2: ;                                             *
 3: ;    8080 PROGRAM TEST    F=I-01M.ASM         *
 4: ;         MOVE MACRO TEST                     *
 5: ;                                             *
 6: ;*********************************************
 7: ;*
 8:          ORG       1000H
 9: ;*
10:          LXI       SP,WSP+128
11: ;***---
12:          AMOVE     T1,W1,16          ; TITLE SET
13: ;*
14:          LDA       WCNT              ; COUNTER
15:          ORI       30H
16:          STA       W1+6
17: ;***---
18:          HLT
19: ;******
20: T1:      DB        'TOTAL X GROUPS'
21:          DB        0DH,0AH
22: ;*
23: W1:      DS        30
24: WCNT:    DB        8
25: WSP:     DS        128
26: ;*
27: ;****** STANDARD SUBROUTINE *******
28:          MACG      SUB
29: ;*
30:          AMOVE     T1,W1,
31:          AMOVE     ,W1,16
32:          AMOVE     T1,,16
33:          AMOVE     ,,
34:          END
```

～マクロの指定

～マクロ内でコールされるサブルーチンの展開依頼

～オペランドを省略した場合の例

展開後のソースプログラム

```
 1: ;*****************************************
 2: ;                                         *
 3: ;    8080 PROGRAM TEST    F=I-01M.ASM     *
 4: ;         MOVE MACRO TEST                 *
 5: ;                                         *
 6: ;*****************************************
 7: ;*
 8:          ORG     1000H
 9: ;*
10:          LXI     SP,WSP+128
11: ;***---
12: ;        AMOVE   T1,W1,16        ; TITLE SET
13:          LXI     D,T1
14:          LXI     H,W1
15:          MVI     A,16
16:          CALL    AMOVE
17: ;*
18:          LDA     WCNT            ; COUNTER
19:          ORI     30H
20:          STA     W1+6
21: ;***---
22:          HLT
23: ;******
24: T1:      DB      'TOTAL X GROUPS'
25:          DB      0DH,0AH
26: ;*
27: W1:      DS      30
28: WCNT:    DB      8
29: WSP:     DS      128
30: ;*
31: ;****** STANDARD SUBROUTINE *******
32: ;        MACG    SUB
33: ;*****************************************
34: ;*       DATA AREA MOVE (REENTRANT)
35: ;*
36: ;*               DE=SND ADR
37: ;*               HL=RSV ADR
38: ;*               A=BYTE SIZE
39: ;*       CALL    AMOVE   (A,BC SAVE)
40: ;*               DE=SND.END+1
41: ;*               HL=RSV.END+1
42: ;*
43: AMOVE:   ORI     0
44:          JZ      AMOVE2
45:          PUSH    PSW
46:          PUSH    B
47:          MOV     B,A
48: ;*
49: AMOVE1:  LDAX    D
50:          INX     D
51:          MOV     M,A
52:          INX     H
53:          DCR     B
54:          JNZ     AMOVE1
55: ;*
56:          POP     B
57:          POP     PSW
58: AMOVE2:  RET
59: ;****** ------
60: ;*
```

マクロ参照(コメント化)

展開

```
61:  ;       AMOVE   T1,W1,
62:          LXI     D,T1
63:          LXI     H,W1
64:          CALL    AMOVE
65:  ;       AMOVE   ,W1,16
66:          LXI     H,W1
67:          MVI     A,16
68:          CALL    AMOVE
69:  ;       AMOVE   T1,,16
70:          LXI     D,T1
71:          MVI     A,16
72:          CALL    AMOVE
73:  ;       AMOVE   ,,
74:          CALL    AMOVE
75:          END
```

昭和56年 3 月 発行

発行所 財団法人 日本情報処理開発協会
東京都港区芝公園3－5－8
機 械 振 興 会 館 内
TEL（434）8211（代表）

印刷所 株式会社 昌 文 社
住所 東京都港区芝5－26－30
TEL （452）4931