ABL Software Environment

For Assembly Language Programming

Kim P. Vo

A technical report

in

The Electrical Engineering

Department

Presented in Partial Fulfilment of the Requirements
for the degree of Master of Engineering at
Concordia University
Montreal, Quebec, Canada

April 1984

ABSTRACT

ABL software environment for
assembly language programming

Kim P. Vo

The development and implementation of a software environment using ABL (Alternative-Based Language) for TI-980 assembly language is described. The ABL concept and utilities, emphasizing structured design technique, modularization and top-down refinement, is found suited for assembly language programmers. The application of the ABL methodology and the software environment is illustrated with the design of a utility for debugging of computer graphics software.

ACKNOWLEDGEMENT

I would like to express my greatest gratitude to my project supervisor, Dr. W.M. Jaworski for his guidance and assistance throughout this project.

I also like to extend my thanks to CAE Electronics Ltd. for allowing me to fulfill my academic requirements using the company resources.

Finally, my sincerest appreciation to my wife for her encouragement throughout the duration of the project.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

APPENDICES

1. A typical session of FUP/G utility.
2. Listings of ABL editor, interpreter.
3. Listings of FUP/G utility.

1. INTRODUCTION

With the emergence of software engineering, the importance of methodology on software quality and productivity has been widely recognized. While programming in high-level language has gradually changed and improved with the use of software techniques such as structured programming, top-down design, modularization..., and of structured languages such as PL/I, PASCAL, C, ADA ..., programming in assembly language still has not changed noticeably. Although high-level languages are used more in applications which until quite recently were written in assembly language, a very significant amount of software is still programmed in assembly language. Despite the impact of new software techniques and concepts on high-level environment, today's assembly language programming is still basically of free style and non-structured. Its quality and productivity leaves much to be desired.

It is observed that many techniques and concepts which are used in high-level language programming can also be used in assembly language programming. Contrary to the popular belief, it is possible and feasible to implement a system approach in assembly language programming.

The purpose of this project is to attempt to improve programming in TI-980 assembly language by implementing the programming concept ABL (alternative-based language).

2. ABL PRINCIPLES AND ENVIRONMENT

ABL is a programming concept and style which is remarkably different from conventional, sequential approach. In ABL technique, the program flow section is completely separated from the program components which perform actions on data objects. The separation is maintained during all stages of software activities (design, writing, testing and execution). A description of ABL concept can be found in JAW081 .

The ABL process of design and implementation of a software system consists of the following steps :

1. Divide the project into a number of tasks called abstract programs PR(i) where i :=1...N.

2. Each program PR(i) is sub-divided into a set of sub-tasks , called clusters C(j) .

   (labelled C1,C2,...,C10 in figure 1)

3. For each cluster C(j) define a set of alternatives AL(k)

   (labelled K0,K1,K2,...)

4. For each alternative AL(k)

   4.1 Define the set of predicates P(1) .

   The predicates act like "guards" for its alternative, i.e. only if the predicates of an alternative in the present cluster are satisfied then the execution of its actions selected from actions A1,A2,... will proceed. Each of the predicates may be a simple or compound predicate which is a function of other variables, but always returns a

boolean value

Y (YES) = true

or

N (NO) = false

(predicates are labelled P0,P1,...,P6 in figure 1)

4.2 Define the set of actions to be executed AC(m) .

An action may be simple or compound. A simple action is a subroutine. A compound action is another abstract program PR(n) at the next lower abstraction level n.

(The actions are labelled A1 to A29. The entries in each alternative indicate the order of actions to be executed)

4.3 Define the next cluster N(k).

In the end of action execution, the control is transferred to the cluster indicated by N(k).

If N(k)=0, the current abstract program will be terminated and control is returned to calling program.

(labelled N)

The cluster, the predicates of alternatives and the next cluster completely determine the program flow. The interpreter of abstract programs selects next alternative by matching current cluster identifier and values of predicates to cluster and predicate entries respectively.

For example :

If current cluster = C5 and predicates P0,P1 = false then

- alternative K7 is selected

- actions A8, A11, A12 are executed

- C6 is selected as next-cluster

and whole process of selecting next alternative is repeated.

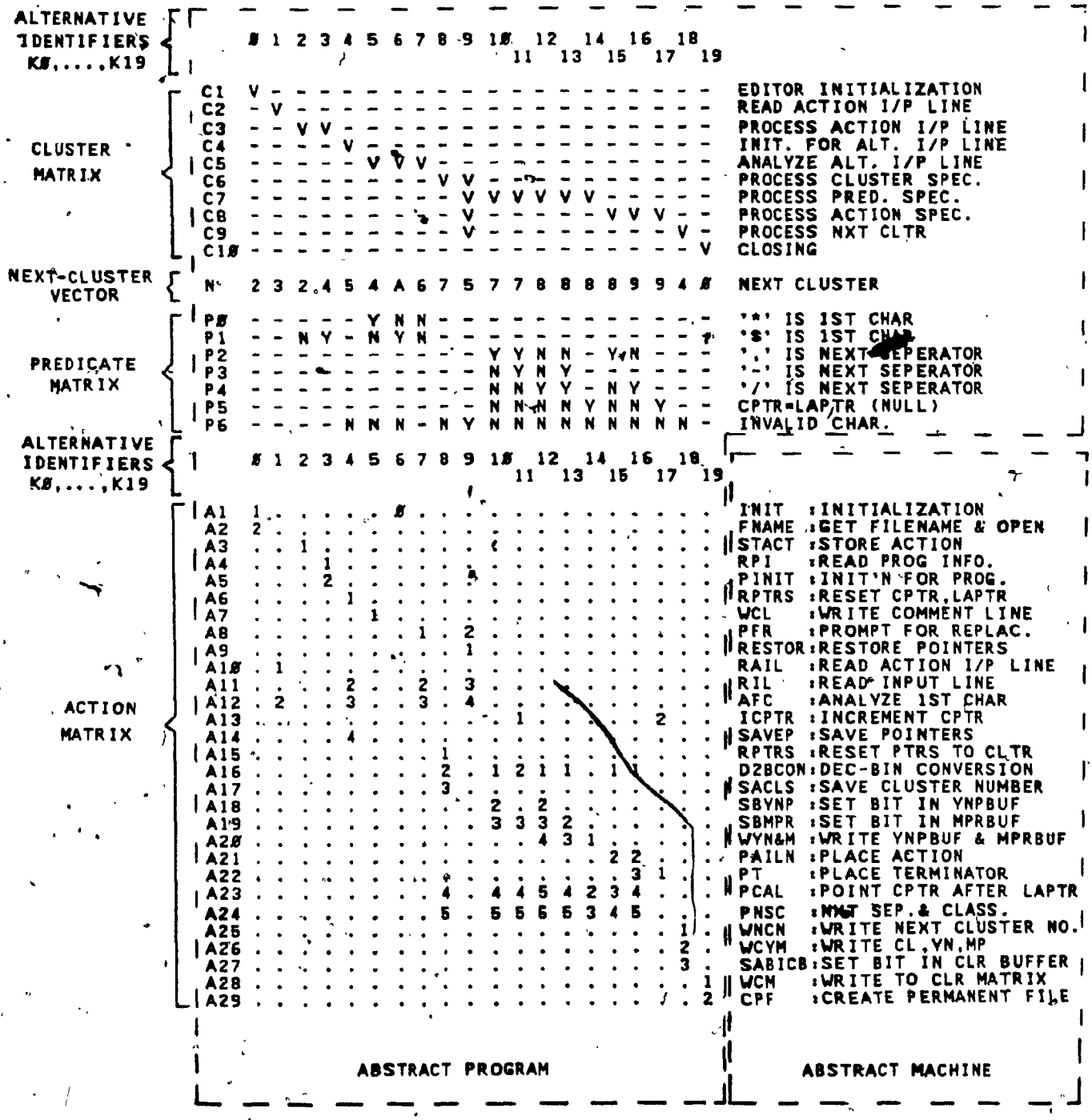


FIGURE 1. Abstract program in matrix form :
- Cluster matrix : C1-C10
- Next-cluster vector : N
- Predicate matrix : P0-P6
- Action matrix : A1-A29

3. IMPLEMENTATION OF ABL IN TI-980 ASSEMBLY LANGUAGE

3.1 HOST SYSTEM

The ABL assembly language environment was implemented on the following system configuration :

HARDWARE :

- TI-980 mini-computer with 64K words of primary memory
- 2 DIVA disk drives, each 80 mega words of random access secondary storage
- TI SILENT 700 ASR console
- ANN ARBOR CRT terminal,
- VERSATEC printer/plotter

SOFTWARE :

- CAE's operating system SIMTOS (Simulator Iterative Multi-Terminal Operating System)
- TI-980 assembler
- Various CAE-developed software utilities

3.2 DESCRIPTION OF THE IMPLEMENTED SYSTEM

Under the ABL assembly environment, a program consits of a pair : abstract program (or program) and abstract machine (or machine) . Abstract program contains all the informations in cluster matrix , next-cluster vector, predicate matrix and action matrix, and represents the control flow. Abstract machine is a set of subroutines each of which carries out a specific action.

The ABL environment consists of the following utilities :

- ABL editor
- Text editor
- ABL interpreter
- Assembler
- Linker

Figure 2 illustrates the steps during software development in ABL environment. To develop and run a software system, the ABL editor is first used to create a file containing abstract program in TI-980 assembly language format. The text editor is used to create abstract machine (subroutines) file. These files, in assembly language format, are passed through the assembler which will produce object code files. The linker will link the object code files to the ABL interpreter which at run time will execute actions of the abstract machine as specified in the abstract program.

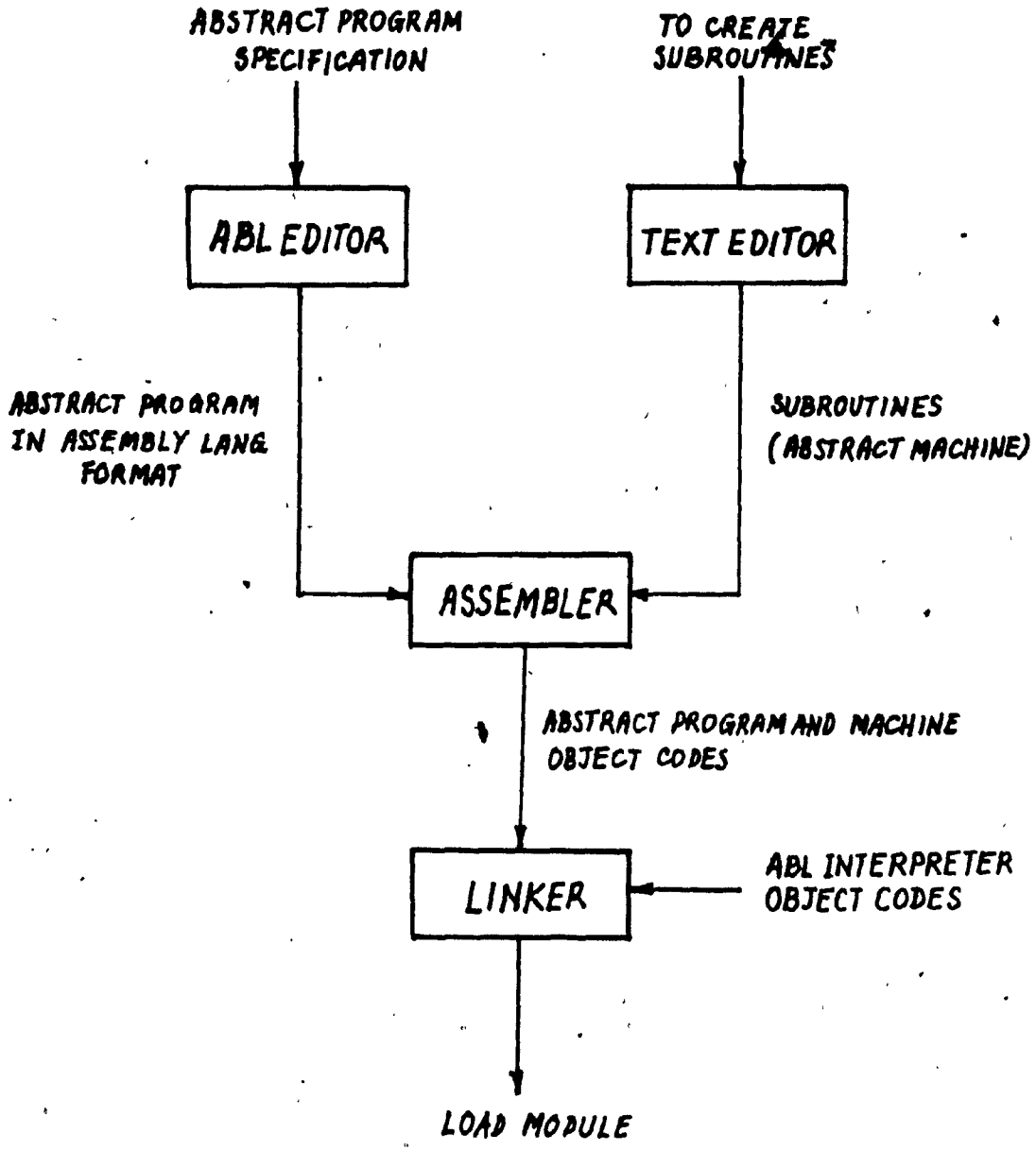


FIGURE 2. Development stages in ABL assembly environment.

3.3 REPRESENTATION OF ABSTRACT PROGRAM IN TI-980 ASSEMBLY LANGUAGE

Figure 3 shows the general structure of abstract program in assembly language, which consists of the following parts :

- Program number : for identification

- Abstract program status block. This status block represents the current status of abstract program at run time. The status block contains the current cluster, the current alternative, the action being executed within the alternative.

- Returning address and address of the calling abstract program if the present abstract program is called by another abstract program.

- Predicates of the state vector of abstract program.

- All matrices of abstract program (see figure 1) : cluster matrix, next-cluster vector, predicate matrix and action matrix.

To minimize memory requirement, the matrices are represented in binary form and there is one pointer pointing to beginning of each matrix. The size of abstract program varies depending on its complexity, therefore there is no unused memory.

A cluster is represented by 2 words, 32 bits. Each bit corresponds to one alternative. A bit in cluster is set to one if the alternative corresponding to that bit belongs to that cluster. Otherwise it is set to zero. For example, cluster 3 in figure 1 is represented as 3000 0000 (HEX) since it contains alternatives 2 and 3.

Predicate matrix is represented by two binary matrices : YES/NO matrix for true/false conditions and mask matrix for "don't

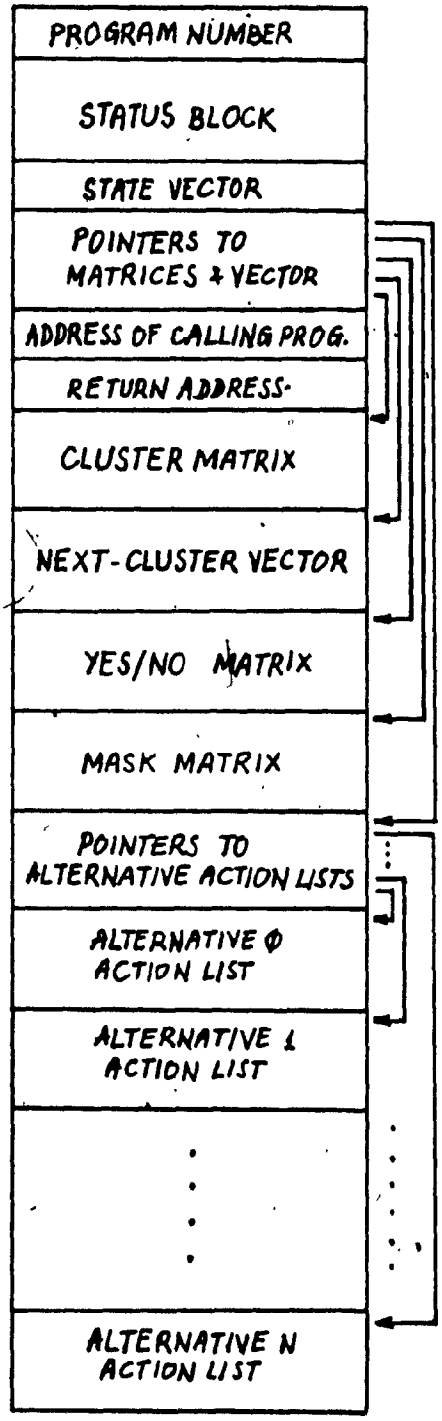


FIGURE 3. Structure of abstract program in assembly language.

care" conditions. A YES condition is represented by the corresponding bit set to one, whereas a NO condition to zero. In both cases, the corresponding bit in mask matrix is set to one. "Don't care" condition is represented by the corresponding bit in mask matrix set to zero.

Action matrix specifies actions of the alternatives. Each alternative has an action list which contains subroutine names which, when assembled and/or linked, will result in subroutine addresses. The action list is terminated by '0'. For each alternative, there is a pointer pointing to the beginning of its action list. These pointers are placed in increasing order, i.e. the pointer to the action list of alternative 0 is followed by the pointer to the action list of alternative 1 and so on. This structure allows an alternative to have any number of actions and at the same time makes efficient use of memory.

Shown in figure 4 is the abstract program from figure 1 represented in this structure :

- PROGNO contains program number.
- PCLUST, PALT, PACTOF, ACLST@ constitute the abstract program status block at run time.
- SVECTR is the predicates of the state vector at run time.
- CLSM@, NCLMA@, YNPMA@, MPRM@ are pointers to cluster matrix CLSMAT, next-cluster vector NCLMAT, predicate matrix YNPMAT, mask matrix MPRMAT respectively.
- APTRO@ is the pointer to the action list pointer area APTR00. APTR00 again contains pointers in order, each pointing to the action

list of an alternative. For example, the first word in APTR00 contains ACLS00 (refer to figure 4) which is the address of the action list of alternative 0. The alternative 0 has actions INIT and FNAME.

```
PROGNO DATA 3              PROGRAM NO.
*
PCLUST DATA Ø              PRESENT CLUSTER
PALT   DATA Ø                 "    ALT
PACT   DATA Ø                 "    ACTION
PACTOF DATA Ø                 "    ACTION OFFSET IN ALTERNATIVE
ACLST@ DATA Ø              PTR TO ACTION LIST FORALT BEING EXE'D
*
SVECTR DATA Ø              STATE VECTOR
*
CLSMA@ DATA CLSMAT         POINTER TO CLUSTER MATRIX
NCLMA@ DATA NCLMAT            "     " NEXT-CLUSTER VECTOR
YNPMA@ DATA YNPMAT            "     " PREDICATE MATRIX
MPRMA@ DATA MPRMAT            "     " PRED MASK MATRIX
APTRØØ DATA APTRØØ            "     " ACTION LIST POINTERS
*
BREG   DATA Ø              (B)=ADDRESS OF PREVIOUS ABSTR. PROGRAM
LREG   DATA Ø              (L)=RETURN ADDRESS
*
*
*     CLUSTER MATRIX  BITSØ-31
*
CLSMAT DATA >8ØØØ,>ØØØØ    C1
       DATA >4ØØØ,>ØØØØ    C2
       DATA >3ØØØ,>ØØØØ    C3
       DATA >ØBØØ,>ØØØØ    C4
       DATA >Ø7ØØ,>ØØØØ    C5
       DATA >ØØCØ,>ØØØØ    C6
       DATA >ØØ7E,>ØØØØ    C7
       DATA >ØØ41,>CØØØ    C8
       DATA >ØØ4Ø,>2ØØØ    C9
       DATA >ØØØØ,>1ØØØ    C1Ø
*
*     NEXT CLUSTER
*
NCLMAT DATA 2              ALT Ø
       DATA 3                  1
       DATA 2                  2
       DATA 4                  3
       DATA 5                  4
       DATA 4                  5
       DATA 1Ø                 6
       DATA 6                  7
       DATA 7                  8
       DATA 5                  9
       DATA 7                  1Ø
       DATA 7                  11
       DATA 8                  12
       DATA 8                  13
       DATA 8                  14
       DATA 8                  15
       DATA 9                  16
       DATA 9                  17
       DATA 4                  18
       DATA Ø                  19
*
*     Y/N PREDICATE MATRIX
*
YNPMAT DATA >ØØØØ          ALT Ø
       DATA >ØØØØ              1
       DATA >ØØØØ              2
       DATA >4ØØØ              3
       DATA >ØØØØ              4
       DATA >8ØØØ              5
```

FIGURE 4. Abstract program from figure 1 represented in TI-980 assembly language

```
        DATA  >4000                6
        DATA  >0000                7
        DATA  >0000                8
        DATA  >0200                9
        DATA  >2000                10
        DATA  >3000                11
        DATA  >0800                12
        DATA  >1800                13
        DATA  >0400                14
        DATA  >2000                15
        DATA  >0800                16
        DATA  >0400                17
        DATA  >0000                18
        DATA  >0000                19
*
*     MASK PREDICATE MATRIX
*
MPRMAT  DATA  >0000           ALT 0
        DATA  >0000                1
        DATA  >4000                2
        DATA  >4000                3
        DATA  >0200                4
        DATA  >C200                5
        DATA  >C200                6
        DATA  >C000                7
        DATA  >0200                8
        DATA  >0200                9
        DATA  >3E00                10
        DATA  >3E00                11
        DATA  >3E00                12
        DATA  >3E00                13
        DATA  >0600                14
        DATA  >2C00                15
        DATA  >2C00                16
        DATA  >0400                17
        DATA  >0200                18
        DATA  >0000                19
*
*     ACTION LIST POINTERS
*
APTR00  DATA  ACLST0          PTR TO ACTION LIST OF ALT 1
        DATA  ACLST1                                    2
        DATA  ACLST2
        DATA  ACLST3
        DATA  ACLST4
        DATA  ACLST5
        DATA  ACLST6
        DATA  ACLST7
        DATA  ACLST8
        DATA  ACLST9
        DATA  ACLT10
        DATA  ACLT11
        DATA  ACLT12
        DATA  ACLT13
        DATA  ACLT14
        DATA  ACLT15
        DATA  ACLT16
        DATA  ACLT17
        DATA  ACLT18
        DATA  ACLT19
*
ACLST0  BSS   0       ALT 0 : INITIALIZATION
        DATA  INIT              ABL EDITOR INIT'N
        DATA  FNAME             GET FILE NAME & DO A/O
        DATA  0
*
```

FIGURE 4. (continued)

```
ACLST1 BSS  Ø      ALT 1 : READ MACHINE I/P LINE
       DATA RAIL          READ ACTION I/P LINE
       DATA AFC           ANALYZE FIRST CHARACTER
       DATA Ø
*
ACLST2 BSS  Ø      ALT 2 : ACTION I/P LINE (SUB NAME)
       DATA STACT         STORE ACTION INTO TEMP. FILE
       DATA Ø
*
ACLST3 BSS  Ø      ALT 3 : MACHINE TERMINATION
       DATA RPI           READ INFO. ON PROGRAM SIZE
       DATA PINIT         INIT'N FOR PROGRAM CREATION
       DATA Ø
*
ACLST4 BSS  Ø      ALT 4 : READ & ANALYZE I/P LINE FOR
       DATA CLRSV         CLEAR STATE VERTOR
       DATA RPTRS         RESET CPTR, LAPTR
       DATA RIL           READ ALT I/P LINE
       DATA AFC           ANALYZE 1ST CHAR
       DATA SAVEP         SAVE POINTERS
       DATA Ø
*
ACLST5 BSS  Ø      ALT 5 : WRITE COMMENT LINE
       DATA WCL           WRITE COMMENT LINE
       DATA Ø
*
ACLST6 BSS  Ø      ALT 6 : PROGRAM I/P TERMINATION
       DATA Ø
*
ACLST7 BSS  Ø      ALT 7 : ALT SPEC I/P LINE
       DATA PFR           PROMPT FOR REPLACEMENT
       DATA RIL           READ I/P LINE
       DATA AFC           ANALYZE 1ST CHARACTER
       DATA Ø
*
ACLST8 BSS  Ø      ALT 8 : PROCESS CLUSTER SPEC
       DATA RPTRS         RESET POINTERS TO CLUSTER POS.
       DATA D2BCON        DEC TO BIN CONVERSION
       DATA SACLS         SAVE CLUSTER
       DATA PCL           POINT CPTR AFTER LAPTR
       DATA PNSC          POINTER TO NEXT SEP & CLASSIFY
       DATA Ø
*
ACLST9 BSS  Ø      ALT 9 : INVALID I/P CHARACTER
       DATA RESTOR        RESTORE POINTERS
       DATA PFR           PROMPT FOR REPLCAEMENT
       DATA RIL           READ I/P LINE
       DATA AFC           ANALYZE 1ST CHARACTER
       DATA Ø
*
ACLT1Ø BSS  Ø      ALT 1Ø: POSITIVE PREDICATE
       DATA D2BCON        DEC TO BIN CONVERSION
       DATA SBYNP         SET BIT IN YNP BUFFER
       DATA SBMPR         SET BIT IN MPR BUFFER
       DATA PCAL          POINT CPTR AFTER LAPTR
       DATA PNSC          POINT TO NEXT SEP. & CLASSIFY
       DATA Ø
*
ACLT11 BSS  Ø      ALT 11: NEGATIVE PREDICATE
       DATA ICPTR         INCREMENT CPTR
       DATA D2BCON        DEC TO BIN CONVERSION
       DATA SBMPR         SET BIT IN MASK PREDICATE MATRIX
       DATA PCAL          POINT CPTR AFTER LAPTR
       DATA PNSC          POINT TO NEXT SEPERATOR & CLASSIFY
       DATA Ø
*
```

FIGURE 4. (continued)

```
ACLT12 BSS  Ø      ALT 12: POSITIVE PREDICATE,LAST PREDICATE
       DATA D2BCON          DEC TO BIN CONVERSION
       DATA SBYNP           SET BIT IN YNP BUFFER
       DATA SBMPR           SET BIT IN MASK PREDICATE MATRIX
       DATA PCAL            POINT CPTR AFTER LAPTR
       DATA WYN&M           WRITE YNPBUF & MPRBUF
       DATA PNSC            POINT TO NEXT SEP. & CLASSIFY
       DATA Ø
*
ACLT13 BSS  Ø      ALT 13: NEGATIVE PREDICATE,LAST PREDICATE
       DATA D2BCON          DEC TO BIN CONVERSION
       DATA SBMPR           SET BIT IN MASK PREDICATE MATRIX
       DATA PCAL            POINT CPTR AFTER LAPTR
       DATA PNSC            POINT TO NEXT SEPERATOR & CLASSIFY
       DATA Ø
*
ACLT14 BSS  Ø      ALT 14: NO PREDICATE SPECIFICATION
       DATA PCAL            POINT CPTR AFTER LAPTR
       DATA PNSC            POINT TO NEXT SEP. & CLASSIFY
       DATA Ø
*
ACLT15 BSS  Ø      ALT 15: ACTION NUMBER
       DATA D2BCON          DEC TO BIN CONVERSION
       DATA PAILN           PLACE ACTION IN ACTION LIST
       DATA PCAL            POINT CPTR AFTER LAPTR
       DATA PNSC            POINT TO NEXT SEPERATOR & CLASSIFY
       DATA Ø
*
ACLT16 BSS  Ø      ALT 16: LAST ACTION NUMBER
       DATA D2BCON          DEC TO BIN CONVERSION
       DATA PAILN           PLACE ACTION IN ACTION LIST
       DATA PT              PLACE TERMINATOR IN ACTION LIST
       DATA PCAL            POINT CPTR AFTER LAPTR
       DATA PNSC            POINT TO NEXT SEPERATOR & CLASSIFY
       DATA Ø
*
ACLT17 BSS  Ø      ALT 17: NO ACTION SPECIFICATION
       DATA PT              PLACE TERMINATOR
       DATA ICPTR           INCREMENT CPTR
       DATA Ø
*
ACLT18 BSS  Ø      ALT 18: PROCESS NEXT CLUSTER
       DATA WNCN            WRITE NEXT CLUSTER NUMBER
       DATA WYM             WRITE TO YN, MASK MATRIX
       DATA SABICB          SET ALT BIT IN CLS. BUFFER,
       DATA Ø
*
ACLT19 BSS  Ø      ALT 19: CLOSING
       DATA WCM             WRITE TO CLUSTER MATRIX
       DATA CPF             CREATE PERMANENT FILE
       DATA Ø
```

FIGURE 4. (continued)

3.4 ABL EDITOR

3.4.1 Description

The process of transforming abstract program from its vertical form (Figure 1) into TI-980 assembly language format (Figure 4) is often very time-consuming and error-prone. Therefore, a software utility which handles this process automatically is naturally needed : the ABL editor accepts the specification of abstract program in a convenient and efficient way from the user and generates a corresponding source file, structured as discussed in section 3.3 (Figure 4), in TI-980 assembly language format.

The ABL editor creates programs by requesting the user to input information through the following steps :

- File name :

For the abstract program to be created. A file name, under SIMTOS, has the form AMNRRT, where :

A : alphabet A,...,Z

M : alphanumeric A,...,Z,0,...,9

N : ASCII

RR: revision level 00,...,99

T : file type

The editor accepts three characters for AMN and opens a source file (type S) with highest revision.

- Abstract machine :

For the abstract machine on which the abstract program will

run on. The abstract machine is a set of subroutines. A subroutine in assembly language is symbolically represented by a label which is actually an address. A subroutine name is entered as a label of up to six characters and optionally comment part. Each subroutine, when entered, is numbered by the ABL editor. The subroutines of abstract machine are referenced in abstract program specification by its number. An abstract machine can have any number of subroutines.

- Abstract program :

Through this step, the user specifies the following information :

- Program number
- Number of clusters
- Number of alternatives
- Alternative specification

The user inputs alternative specification by inputting the specification for each alternative which consists of a comment part for the alternative and the actual specification itself :

- Cluster to which the alternative belongs
- Its relative order within the cluster
- Predicates (negative or positive) for the alternative
- Subroutines (actions) to be executed
- Next-cluster to transfer to when execution of actions is completed

Alternative specification is input sequentially : the first specification is for the first alternative, the second specification is for the second alternative, etc.

3.4.2 Description of ABL editor command language in Backus-Naur form

This section describes the command language for the ABL editor in Backus-Naur form.

Notes :

1. The following three symbols belong to Backus-Naur formalism, and not of ABL editor.

   ::= denotes 'defined as'

   [*] denotes possible repetition of the enclosed symbol(s) zero or one more time.

   ! denotes "OR" condition

2. The following six symbols are of ABL editor command language :

   $ denotes termination of machine or program section

   * denotes first character of the comment line

   . denotes seperator of cluster number and relative alternative number within cluster

   / denotes delimiter between predicate, action and next cluster sections

   - denotes negative predicate condition

   , denotes continuation of predicate or action section

⟨abstract program spec.⟩ ::= ⟨file name spec.⟩
⟨machine spec.⟩
⟨program spec.⟩

⟨file name spec.⟩ ::= ⟨alphabet⟩ ⟨alphanumeric⟩ ⟨ascii⟩

⟨alphabet⟩ ::= A,...,Z

⟨alphanumeric⟩ ::= A,...,Z,0,...,9

⟨ascii⟩ ::= any ASCII character

⟨machine spec.⟩ ::= [⟨subroutine⟩]
⟨terminator⟩

⟨subroutine⟩ ::= ⟨subroutine name⟩ [⟨comment⟩]

⟨subroutine name⟩ ::= valid label in TI-980 assembly, up to
6 characters

⟨comment⟩ ::= ⟨empty⟩ ! ⟨character string⟩

⟨empty⟩ ::=

⟨character string⟩ ::= any combination of ASCII, including blank
terminated by carriage return

⟨terminator⟩ ::= $

⟨program spec.⟩ ::= ⟨program number⟩
⟨number of clusters⟩
⟨number of alternatives⟩
⟨alternatives⟩
⟨terminator⟩

⟨program number⟩ ::= 0,...,99

⟨number of clusters⟩ ::= 1,...,99

⟨number of alternatives⟩ ::= 1,...,32

⟨alternatives⟩ ::= [⟨alternative⟩]

⟨alternative⟩ ::= [⟨alternative comment⟩]
⟨alternative spec.⟩

⟨alternative comment⟩ ::= *⟨comment⟩

⟨alternative spec.⟩ ::= ⟨cluster⟩ /⟨pred⟩/⟨action⟩/⟨next cluster⟩

⟨cluster⟩ ::= ⟨cluster number⟩.⟨relative alternative number⟩

⟨cluster number⟩ ::= any number smaller or equal number of clusters

⟨relative alt. number⟩ ::= relative number of alternative within
cluster, sequentially numbered

⟨pred⟩ ::= ⟨empty⟩ ![⟨predicate description⟩]

⟨pred description⟩ ::= ⟨sign⟩⟨pred #⟩ [,⟨sign⟩⟨pred #⟩]

⟨sign⟩ ::= ⟨empty⟩ ! -

⟨pred #⟩ ::= 0,...,15

⟨action⟩ ::= ⟨empty⟩ ![⟨action description⟩]

⟨action description⟩ ::= ⟨action number⟩ [,⟨action number⟩]

⟨action number⟩ ::= any number assigned to a subroutine by ABL
editor during machine specification

⟨next cluster⟩ ::= any number smaller or equal to the number of
clusters

3.4.3 Example

Figure 5 shows a session of ABL editor to create an abstract program. The user inputs are entered after the prompt '>' from the editor.

Figure 6 is the file in TI-980 assembly language format created by the ABL editor.

```
EH? 00:00:15 ABLEDI

ABL EDITOR


FILE NAME )V6#

MACHINE SPECIFICATION
  ENTER SUBROUTINE NAMES
  1 )INIT        1.INITIALIZATION
  2 )RDCMD       2.READ COMMAND & SET PREDICATES
  3 )A3          3.CHECK COMMAND SYNTAX
  4 )GRTI1       4.READ GR7 INST. TO TI BUFFER 1
  5 )A5          5.DISASSEMBLE GR7 INST. TO CONSOLE
  6 )GRTI2       6.READ GR7 INST. TO TI BUFFER 2
  7 )A6          7.DISASSEMBLE GR7 INST. TO L.P.
  8 )MODD        8.MODIFY BUFFER
  9 )WBTGR7      9.WRITE BUFFER TO GR7 MEMORY
 10 )RLPCSP     10.RELEASE LP & CLOSE SPOOL FILE
 11 )RQLOSP     11.REQUEST LP & OPEN SPOOL FILE
 12 )RQCON      12.REQUEST CONSOLE FOR OUTPUT
 13 )RLCON      13.RELEASE CONSOLE
 14 )$

PROGRAM SPECIFICATION
  PROGRAM NUMBER )1
  NUMBER OF CLUSTERS )3
  NUMBER OF ALTERNATIVES )6
  ALTERNATIVE SPECIFICATION :
  )*            - ALT 0 : INITIALIZATION
  )1.1 //1/2
  )*            - ALT 1 : READ COMMAND & CHECK SYNTAX
  )2.1 //2,3/3
  )*            - ALT 2 : 'L'
  )3.1 /-0,-1,-2,3,-4/12,4,5,13/2
  )*            - ALT 3 : 'H'
  )3.2 /-0,-1,2,-3,-4/4,6,11,7,10/2
  )*            - ALT 4 : 'M'
  )3.3 /-0,-1,2,3,-4/8,5/2
  )*            - ALT 5 : 'W'
  )3.4 /-0,1,-2,-3,-4/9/2
  )$

ABL PROGRAM CREATED IN FILE V6#32S
....S'LONG  00:11:46!
```

FIGURE 5. A session of ABL editor.

```
 1 PROGNO DATA 1
 2 PCLUST DATA Ø
 3 PALT   DATA Ø
 4 PACT   DATA Ø
 5 PACTOF DATA Ø
 6 ACLST@ DATA Ø
 7 SVECTR DATA Ø
 8 CLSMA@ DATA CLSMAT
 9 NCLMA@ DATA NCLMAT
1Ø YNPMA@ DATA YNPMAT
11 MPRMA@ DATA MPRMAT
12 APTRØ@ DATA APTRØ
13 BREG   DATA Ø
14 LREG   DATA Ø
15 *
16 CLSMAT BSS Ø
17        DATA  >8ØØØ
18        DATA  >ØØØØ
19        DATA  >4ØØØ
2Ø        DATA  >ØØØØ
21        DATA  >3CØØ
22        DATA  >ØØØØ
23 NCLMAT BSS Ø
24        DATA 2
25        DATA 3
26        DATA 2
27        DATA 2
28        DATA 2
29        DATA 2
3Ø YNPMAT BSS Ø
31        DATA  >ØØØØ
32        DATA  >ØØØØ
33        DATA  >1ØØØ
34        DATA  >2ØØØ
35        DATA  >3ØØØ
36        DATA  >4ØØØ
37 MPRMAT BSS Ø
38        DATA  >ØØØØ
39        DATA  >ØØØØ
4Ø        DATA  >FBØØ
41        DATA  >FBØØ
42        DATA  >FBØØ
43        DATA  >FBØØ
44 APTRØ  BSS Ø
45        DATA ACLSØØ
46        DATA ACLSØ1
47        DATA ACLSØ2
48        DATA ACLSØ3
49        DATA ACLSØ4
5Ø        DATA ACLSØ5
51 ACLSØØ BSS Ø
52 *            - ALT Ø : INITIALIZATION
53        DATA INIT       1.INITIALIZATION
54        DATA Ø
55 *
56 ACLSØ1 BSS Ø
57 *            - ALT 1 : READ COMMAND & CHECK SYNTAX
58        DATA RDCMD      2.READ COMMAND & SET PREDICATES
59        DATA A3         3.CHECK COMMAND SYNTAX
6Ø        DATA Ø
61 *
```

FIGURE 6. - An abstract program created by ABL editor.

```
 62 ACLS#2 BSS #
 63 *              - ALT 2 : 'L'
 64         DATA RQCON         12.REQUEST CONSOLE FOR OUTPUT
 65         DATA GRTI1         4.READ GR7 INST. TO TI BUFFER 1
 66         DATA A5            5.DISASSEMBLE GR7 INST. TO CONSOLE
 67         DATA RLCON         13.RELEASE CONSOLE
 68         DATA #
 69 *
 7# ACLS#3 BSS #
 71 *              - ALT 3 : 'H'
 72         DATA GRTI1         4.READ GR7 INST. TO TI BUFFER 1
 73         DATA GRTI2         6.READ GR7 INST. TO TI BUFFER 2
 74         DATA RQLOSP        11.REQUEST LP & OPEN SPOOL FILE
 75         DATA A6            7.DISASSEMBLE GR7 INST. TO L.P.
 76         DATA RLPCSP        1#.RELEASE LP & CLOSE SPOOL FILE
 77         DATA #
 78 *
 79 ACLS#4 BSS #
 8# *              - ALT 4 : 'M'
 81         DATA MODD          8.MODIFY BUFFER
 82         DATA A5            5.DISASSEMBLE GR7 INST. TO CONSOLE
 83         DATA #
 84 *
 85 ACLS#5 BSS #
 86 *              - ALT 5 : 'W'
 87         DATA WBTGR7        9.WRITE BUFFER TO GR7 MEMORY
 88         DATA #
 89 *
 9#         END
V6#32S    #2/#3/84    13:#3:14
```

FIGURE 6. (continued)

3.5 ABL INTERPRETER

At the heart of the ABL environment is the ABL interpreter. This interpreter runs abstract programs by comparing the predicates of the alternatives in the current cluster. It executes the actions of the alternative whose predicates match the state vector, and then goes to the next cluster to repeat the same process.

The predicates of the state vector are set or cleared in the action subroutines. The program flow is dependent on these action subroutines through the state vector predicates that they set or clear. The interpreter, acting like a state-driven driver, accepts the predicates and executes appropriate actions.

Hierarchical structure

Actions in abstract program can be a simple or a compound action, an abstract program. Therefore, an abstract program can call another abstract programs. The interpreter supports any number of abstract programs in this hierarchical structure. This is done by making use of the base relative addressing mode in TI-980 : the base register always contains the starting address of the abstract program that the interpreter is presently executing. When an action within a program calls another program, the starting address of the called program is passed to the interpreter. The interpreter responds to the call to execute a new program by saving the address of the current program, currently contained in the base register, into a pre-determined location in the called program, placing the address of the called program into the base register,

and then starts executing the called program. When execution is completed, the interpreter restores the previously saved address of the calling abstract program into the base register and resumes executing the calling program. This scheme allows the interpreter to execute any number of abstract programs in any level. Figure 7 illustrates this inter-program call feature.

Program tracing

The interpreter has the capability to trace the program flow of abstract programs. A program can be traced by its program number, alternative numbers within the program and the state vector at the end (or before) the alternative execution. This information completely determines the sequence of alternatives executed within each program as well as the sequence of program-to-program calls.

When tracing option is selected, the following information is output to the console :

- Program number to announce the beginning of the program execution.

- Alternative numbers and its resulting state vectors, in execution order.

- The completion of program execution by its number.

Detection of run-time unauthorized conditions

The interpreter also detects unauthorized conditions at run time. These conditions, arising when the interpreter cannot find a match between the state vector of the program and the predicates of any of the alternatives in the current cluster, will cause the

interpreter to stop executing. The interpreter will output the number of the last alternative executed and the final state vector. This information is useful for debugging.

Calling procedure

An abstract program is executed by calling the ABL interpreter with the M register containing its starting address.

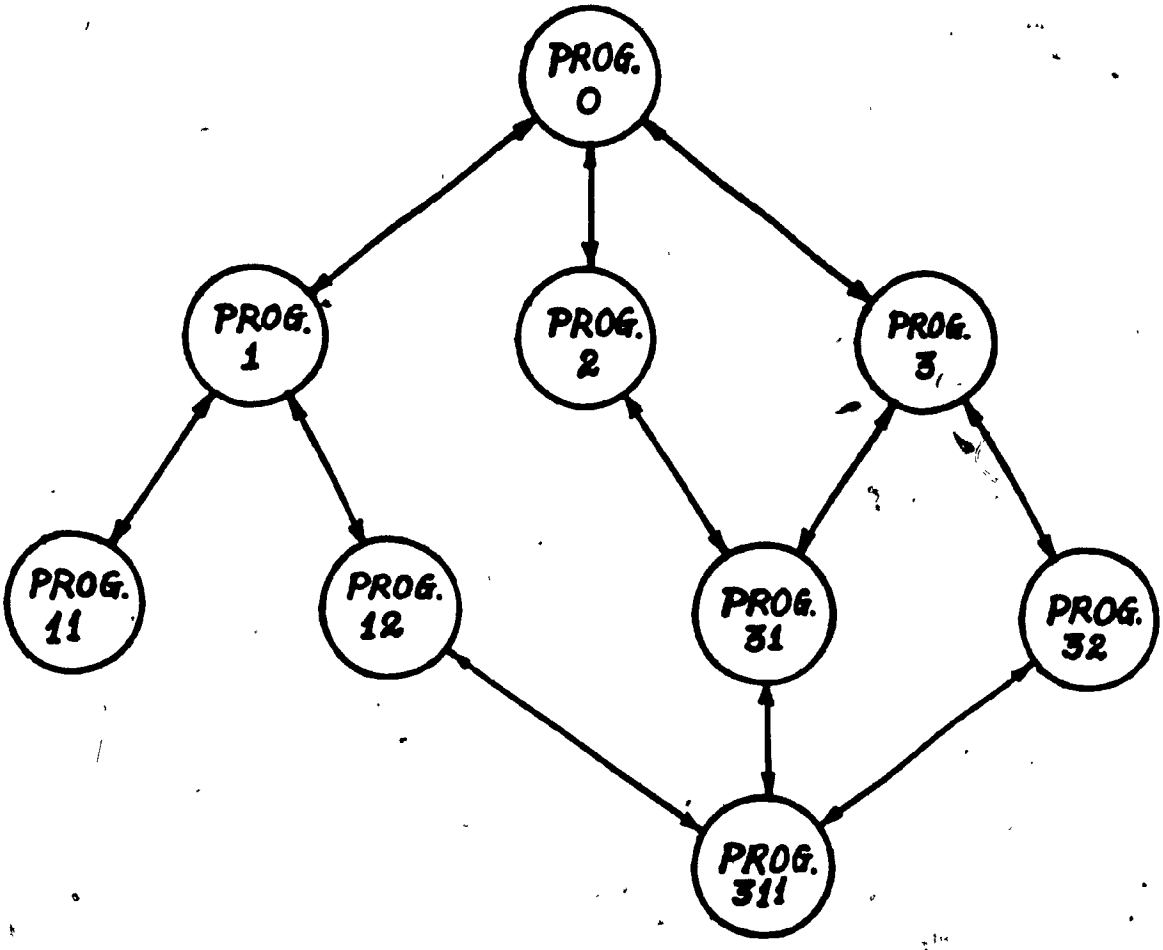


FIGURE 7. Hierarchical structure of abstract programs.

4. APPLICATION : A UTILITY FOR DEBUGGING OF REAL-TIME INTERACTIVE COMPUTER GRAPHICS

4.1 THE EXISTING ENVIRONMENT

HARDWARE :

A typical flight simulator system consists of the simulator itself and an instructor station. The instructor station consists of the facilities which provide the flight instructor means to control the simulator and to keep track of the pilot trainee's performance. This is done through various types of input/output devices such as keyboards, panel of special function buttons, graphic display units (DGU), hard copy units, etc. Of these devices, the DGU is the most important with its capability to provide information in different forms :

- alpha-numeric pages
- real-time interactive maps
- real-time interactive plottings...

In the early generation of flight simulators, the DGU had its display file stored in host-resident memory. Update of display file, which is generated by the host computer, is just a matter of updating host memory by host program. The debugging of the display file is done by examining and/or modifying the memory portion affected. Utilities for this type of debugging are simple and available.

With the evolution of the micro-processors, most of today's graphic display systems have its own processor(s) and its own

memory. In the current generation of flight simulators, one of the most widely used graphic systems is GRAPHIC-7 and GRAPHIC-8 series (Calcom, Sanders Associates Inc.).

GRAPHIC-7/8 is intelligent graphic system which has :

- Two seperate 16-bit processors : a graphic processor and a graphic controller. The graphic processor, which emulates PDP-11 instructions, is used to handle the communication between the host and the GRAPHIC-7/8 unit, the input from position entry devices such as track ball, joystick, photopen, etc. The graphic controller is used to generate display images to the monitor from display file.

- Its own memory of 32K 16-bit words, expandable to 128K words. The graphic processor and the graphic controller share the same memory; each uses a different area.

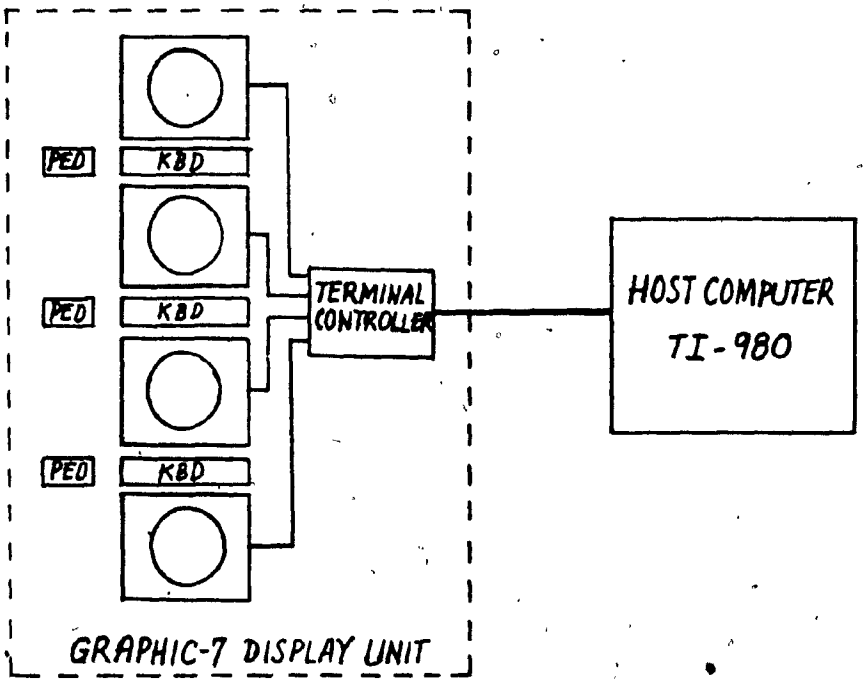
- GRAPHIC-7 is a vector display whereas GRAPHIC-8 is a pixel-scanned display.

- GRAPHIC-7 graphic instruction set is a subset of GRAPHIC-8 (upward compatibility).

Figure 8 shows a typical configuration between GRAPHIC-7/8 and its TI-980 host computer.

SOFTWARE :

For applications such as flight simulator, GRAPHIC-7/8 is used to display alpha-numeric pages, real-time interactive maps and plottings. Display files, containing GRAPHIC-7/8 instructions, are generated by the programs in the host computer and then sent to



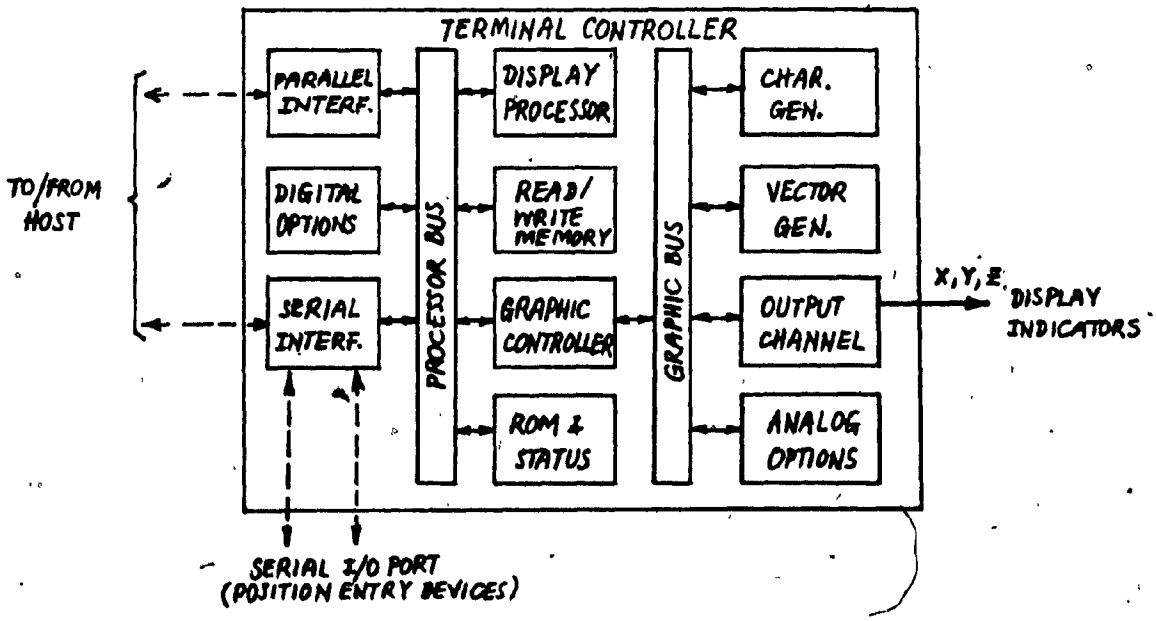


FIGURE 8. GRAPHIC-7/8 display unit and its TI-980 host computer.

GRAPHIC-7/8 memory which is used by the graphic controller to draw images on the monitor. Updating display files is done in the same way. The size of a typical display file is normally about 32K words.

Debugging is done through a special keyboard supplied by Sanders Inc.. This keyboard lets the user examine and modify any location in GRAPHIC-7/8 memory, one word at a time. The addresses and contents are specified or displayed in octal numbers. Image generation can be restarted from any location in display file.

Under this debugging facility, the user has to perform the following steps in order to examine and/or modify refesh display file :

- typing in the address (octal) of the location to be examined
- when the content is displayed, decode it from machine code to mnemonic code
- typing in the new content, if desired
- repeating the same process for every location to be examined and/or modified.

4.2 THE NEW DEBUGGING FACILITY

The above debugging facilities are clearly very time-consuming. In addition, TI-980 programs run on hexadecimal numbers whereas GRAPHIC-7/8 runs on octal numbers. The programmer, when debugging his/her software, must constantly perform conversion between the two number systems. This places another burden on an already inadequate debugging method.

A new software debugging facility is naturally desired. The Foreground Utility Program for Graphics (FUP/G) is to provide the programmers with the facility to debug display file on-line :

- FUP/G lets the user examine any portion of display file :
  * the size of the portion to be examined varies from 1 to 76 words.
  * two formats are supported when specifying the portion to be examined : in the first format, the user specifies the starting address and the number of bytes. In the second format, the user specifies the starting and ending address.
- The content of the portion of the display file is displayed in GRAPHIC-7/8 mnemonic form as well as in machine code in in octal and hexadecimal representation. The addresses of the locations are displayed in decimal, octal and hexadecimal numbers.
- The user can modify any location(s) in the examined portion of display file.
- Debugging activities are carried out on-line : the image of modified display file is immediately generated on the monitor.

Command language :

At present, the utility FUP/G supports the following commands:

L : List the content of the specified portion of display file to console. This command reads the desired area from GRAPHIC-7/8 memory to a buffer in TI-980. The content is displayed in

machine code (octal and hexadecimal) and in mnemonic form.

Format :

L ⟨start address⟩ TO ⟨end address⟩

or

L ⟨start address⟩ ⟨number of bytes⟩

H : Give a Hard copy of display file to line printer. This command prints the content of the specified portion in machine and mnemonic form to line printer.

Format :

H ⟨start address⟩ TO ⟨end address⟩

or

H ⟨start address⟩ ⟨number of bytes⟩

M : Modify the content of display file previously brought to TI-980 buffer by 'L' command. More than one location can be modified by a single 'M' command.

Format :

M ⟨start address⟩ ⟨new content⟩[,⟨new content⟩]

W : Write the modified content of display file from TI-980 buffer to GRAPHIC-7/8 memory, i.e. update the display file to generate a new image on the monitor.

Format :

W

CR : Carriage return, to repeat the previous command

Format :

(carriage return key)

Q : Terminate utility.

Format :

Q

Future expansion and requirement on program structure

The above commands of the utility FUP/G satisfy the immediate debugging need. To provide more debugging power, more commands will gradually be developed and implemented in future. Therefore, it is required that the utility FUP/G be written in such a structure that the future expansion in the command language can be implemented easily.

4.3 DESCRIPTION OF IMPLEMENTATION OF THE UTILITY USING THE ABL ENVIRONMENT

The utility FUP/G is designed, structured and created under the ABL environment. The ABL structure, with its highly modular, top-down structured characteristics, gives the utility the flexibility needed to satisfy the future expansion requirements.

The ABL editor was used to create abstract programs. The text editor was then used to create abstract machines, i.e. subroutines. These modules were linked to the ABL interpreter to create the load module which was named 'FUP/G' in the utility library.

The utility FUP/G presently consists of three programs :

- Main program (module V1#)
- Syntax check (module V3#), to perform syntax check on all commands.

- Disassembler (module V4#), to disassemble GRAPHIC-7/8 graphic instructions into mnemonic form. This module is used by 'L' and 'H' commands. Module V9# contains a number of action subroutines for the disassembler program.

In addition, modules VD# and MM# contain all common subroutines such as conversions, formatting shared by all action subroutines.

Appendix 1 shows a typical session of the utility FUP/G and the result.

5. CONCLUSION

During the design and implementation of the utility FUP/G, it was observed that the ABL approach gave a remarkable improvement on productivity over the conventional approach during the design, coding and testing stages. On the other hand, program space and execution time are two main advantages of assembly language programming. Therefore, let us examine the effects of the implemented ABL environment on these two factors :

- SPACE EFFICIENCY : instructions to control program flow such as 'test and skip if' each takes one or two words (16 bits/word). In ABL, each predicate takes only two bits. In a number of typical real-time assembly programs surveyed, it was found that the flow control instructions accounted for about ten percent of program size. The reduction of memory size by replacing flow control instructions by ABL binary predicates is significant.

- TIME EFFICIENCY : although there is an overhead during run time when the interpreter evaluates the predicates to select an alternative to execute, this overhead is negligible due to the fact that only the predicates of the alternatives in the current cluster are evaluated and only one instruction is used to compare the state vector to the predicates of each alternative. Time-wise, execution in ABL is as good as in conventional assembly programming.

The implemented ABL environment is a minimum configuration. Other features of ABL are found very useful which, when implemented,

should significantly enhance its power. Considered most important are :

- The ability of the interpreter to change program flow during run time by temporarily stopping execution when certain conditions in the state vector are met, changing the predicates of the state vector and/or of alternative(s) and next-cluster vector, and then resuming execution. This would be a very useful, dynamic debugging tool which effectively lets the user try different abstract programs on the same machine.

- The ability of the ABL editor to generate the matrix form (as in Figure 1) of the program created by the user.

The ABL concept and techniques are found very suited for assembly language programming. Although this project was implemented on TI-980 mini-computer, the techniques and design are basically transferrable to other assembly languages.

The implemented ABL environment offers a highly systematic approach and a great amount of flexibility in design and programming and at the same time it enhances and retains space and time efficiency respectively, which constitute the two most important advantages of assembly language programming. Its suitability and therefore potential use as tool for building software in assembly language is convincing and clear.

BIBLIOGRAPHY

ACMS81 ACM SIGSOFT - 'NBS programming environment workshop report' Software engineering notes, Vol 6 No. 4, August 1981.

ANAS79 M.S. Anastas & R.F. Vaughau - 'Direct architectural implementation of a requirement-oriented computing structure', Microprogramming workshop, AMC Sigmicro newsletter, 1979.

FREE75 P. Freeman - 'Software system principles - A survey', Science Research Associates Inc., 1975.

JAWO81 W.M. Jaworski & H. Hinterger - 'Controlled program design by use of the programming concept ABL', Applied Informatics, July 1981.

KRON82 M.N. Kronick - 'An ABL software environment for a minicomputer', thesis for the degree of master of computer science, Concordia University, March 1982.

LINA82 J. Linares - 'Design of a comprehensive micro-programming system', thesis for the degree of master of computer science, Concordia University, July 1982.

MITC79 J.G. Mitchell - 'The design and construction of flexible and efficient interactive programming system', Garland Publishing, New York 1979.

MYER80 G.J. Myers - 'The art of software testing' - John Wiley & Sons, Toronto 1980.

NEWM79 W.M. Newman & R.F. Sproull - 'Principles of interactive

computer graphics', McGraw-Hill 1979.

SAND78 Sanders Ass. Inc., 'Graphic-7, computer graphic display system, technical description', Sanders Ass. Inc., New Hampshire, 1978.

SAND81 Sanders Ass. Inc., 'Graphic-8, computer graphic display system, series 8X00, programmer's reference manual', Sanders Ass. Inc., New Hampshire, 1981.

SCHN79 N.F. Schneidewind & H.M. Hoffman - 'An experiment in software data collection and analysis', IEEE transaction on software engineering, May 1979.

SHNE80 B. Shneiderman - 'Software psychology'- Human factors in computer and information systems', Winthrop Publishers Inc., 1980.

SHOO83 M.L. Shooman - 'Software engineering - Design, Reliability, Management', McGraw-Hill 1983.

APPENDIX 1

A1.1 A typical session of utility FUP/G on a display file which contains GRAPHIC-7 instruction set, stored from location 3000 to 3072 (hex) in GRAPHIC-7 memory.

A1.2 Hard copy print-outs on line printer as result of 'H' commands in FUP/G session.

```
EH? 17:16:49 FUP/G

FUP/G V8


*FUP/G: L 3000 74
```

DEC ADDR	HEX ADDR	OCT ADDR	HEX VALUE	OCT VALUE		
12288	3000	030000	0000	000000	HALT	
12290	3002	030002	0200	001000	JUMP	1000
12292	3004	030004	1000	010000		
12294	3006	030006	0200	001000	*JUMP	1000
12296	3008	030010	9000	110000		
12298	300A	030012	0240	001100	JRMP	100
12300	300C	030014	0100	000400		
12302	300E	030016	0280	001200	JMPZ	2000
12304	3010	030020	2000	020000		
12306	3012	030022	0280	001200	*JMPZ	2000
12308	3014	030024	A000	120000		
12310	3016	030026	02C0	001300	JPRZ	100
12312	3018	030030	0100	000400		
12314	301A	030032	0400	002000	JMPM	1500
12316	301C	030034	1500	012400		
12318	301E	030036	0440	002100	CALL	9000
12320	3020	030040	9000	110000		
12322	3022	030042	0480	002200	CALR	1000
12324	3024	030044	1000	010000		
12326	3026	030046	04C0	002300	RTRN	
12328	3028	030050	0600	003000	IZPR	
12330	302A	030052	0800	004000	*LINK	0100
12332	302C	030054	8100	100400		
12334	302E	030056	0840	004100	SAVD	0
12336	3030	030060	0880	004200	RESD	0
12338	3032	030062	08C0	004300	ADDI	0,-100
12340	3034	030064	FF00	177400		
12342	3036	030066	0A00	005000	JMPR	0
12344	3038	030070	0C00	006000	LDRI	0,0,00FF
12346	303A	030072	00FF	000377		
12348	303C	030074	0C40	006100	LDDI	0,0500
12350	303E	030076	0500	002400		
12352	3040	030100	0C80	006200	LDSP	4000
12354	3042	030102	4000	040000		
12356	3044	030104	0E00	007000	WATE	
12358	3046	030106	1000	010000	LDDZ	0
12360	3048	030110	1800	014000	LDDP	0
12362	304A	030112	2000	020000	LDXA	0
12364	304C	030114	2800	024000	LDXR	0
12366	304E	030116	3000	030000	DRXA	0
12368	3050	030120	3800	034000	DRXR	0
12370	3052	030122	4000	040000	DRYA	0
12372	3054	030124	4800	044000	DRYR	0
12374	3056	030126	5000	050000	MVXA	0

```
    12376  3058  030130     5800  054000    MVXR  0
    12378  305A  030132     6000  060000    MVYA  0
    12380  305C  030134     6800  064000    MVYR  0
    12382  305E  030136     7000  070000    LDKX  0
    12384  3060  030140     7800  074000    DRKY  0
    12386  3062  030142     8000  100000    DRSR  0,0
    12388  3064  030144     8040  100100    MVSR  0,0
    12390  3066  030146     C000  140000    PPLR  0,0
    12392  3068  030150     C040  140100    LDTI  0
    12394  306A  030152     9FC1  117701    CHAR  0
    12396  306C  030154     C3C2  141702    TEXT  CB
    12398  306E  030156     0E40  007100    INIT
    12400  3070  030160     0DC0  006700    CLRM
    12402  3072  030162     0E80  007200    MODE  0

*FUP/G: H 3000 TO 3072

*FUP/G: M 3008 1500

*FUP/G: M 302E 842 881 08C3 100

*FUP/G: W

*FUP/G: H 3000 74

*FUP/G: L 3000 TO 3072
    12288  3000  030000     0000  000000    HALT
    12290  3002  030002     0200  001000    JUMP  1000
    12292  3004  030004     1000  010000
    12294  3006  030006     0200  001000    JUMP  1000
    12296  3008  030010     9000  110000
    12298  300A  030012     0240  001100    JRMP  100
    12300  300C  030014     0100  000400
    12302  300E  030016     0280  001200    JMPZ  2000
    12304  3010  030020     2000  020000
    12306  3012  030022     0280  001200   *JMPZ  2000
    12308  3014  030024     A000  120000
    12310  3016  030026     02C0  001300    JPRZ  100
    12312  3018  030030     0100  000400
    12314  301A  030032     0400  002000    JMPM  1500
    12316  301C  030034     1500  012400
    12318  301E  030036     0440  002100    CALL  9000
    12320  3020  030040     9000  110000
    12322  3022  030042     0480  002200    CALR  1000
    12324  3024  030044     1000  010000
    12326  3026  030046     04C0  002300    RTRN
    12328  3028  030050     0600  003000    IZPR
    12330  302A  030052     0800  004000   *LINK  0100
    12332  302C  030054     8100  100400
    12334  302E  030056     0840  004100    SAVD  2
    12336  3030  030060     0880  004200    RESD  1
    12338  3032  030062     08C0  004300    ADDI  3,100
    12340  3034  030064     FF00  177400
```

```
12342  3036  030066    0A00  005000    JMPR  .0
12344  3038  030070    0C00  006000    LDRI  0,0,00FF
12346  303A  030072    00FF  000377
12348  303C  030074    0C40  006100    LDDI  0,0500
12350  303E  030076    0500  002400
12352  3040  030100    0C80  006200    LDSP  4000
12354  3042  030102    4000  040000
12356  3044  030104    0E00  007000    WATE
12358  3046  030106    1000  010000    LDDZ  0
12360  3048  030110    1800  014000    LDDP  0
12362  304A  030112    2000  020000    LDXA  0
12364  304C  030114    2800  024000    LDXR  0
12366  304E  030116    3000  030000    DRXA  0
12368  3050  030120    3800  034000    DRXR  0
12370  3052  030122    4000  040000    DRYA  0
12372  3054  030124    4800  044000    DRYR  0
12374  3056  030126    5000  050000    MVXA  0
12376  3058  030130    5800  054000    MVXR  0
12378  305A  030132    6000  060000    MVYA  0
12380  305C  030134    6800  064000    MVYR  0
12382  305E  030136    7000  070000    LDKX  0
12384  3060  030140    7800  074000    DRKY  0
12386  3062  030142    8000  100000    DRSR  0,0
12388  3064  030144    8040  100100    MVSR  0,0
12390  3066  030146    C000  140000    PPLR  0,0
12392  3068  030150    C040  140100    LDTI  0
12394  306A  030152    9FC1  117701    CHAR  0
12396  306C  030154    C3C2  141702    TEXT  CB
12398  306E  030156    0E40  007100    INIT
12400  3070  030160    0DC0  006700    CLRM
12402  3072  030162    0E80  007200    MODE  0

*FUP/G: Q

FUP/G TERMINATES
....S'LONG  17:27:21!
```

```
FUP/G     09/03/84     17:28:1
   DEC    HEX     OCT     HEX     OCT
   ADDR   ADDR    ADDR    VALUE   VALUE
   ----   ----    ----    -----   -----
   12288  3000    030000    0000   000000    HALT
   12290  3002    030002    0200   001000    JUMP   1000
   12292  3004    030004    1000   010000
   12294  3006    030006    0200   001000   *JUMP   1000
   12296  3008    030010    9000   110000
   12298  300A    030012    0240   001100    JRMP   100
   12300  300C    030014    0100   000400
   12302  300E    030016    0280   001200    JMPZ   2000
   12304  3010    030020    2000   020000
   12306  3012    030022    0280   001200   *JMPZ   2000
   12308  3014    030024    A000   120000
   12310  3016    030026    02C0   001300    JPRZ   100
   12312  3018    030030    0100   000400
   12314  301A    030032    0400   002000    JMPM   1500
   12316  301C    030034    1500   012400
   12318  301E    030036    0440   002100    CALL   9000
   12320  3020    030040    9000   110000
   12322  3022    030042    0480   002200    CALR   1000,
   12324  3024    030044    1000   010000
   12326  3026    030046    04C0   002300    RTRN
   12328  3028    030050    0600   003000    IZPR
   12330  302A    030052    0800   004000   *LINK   0100
   12332  302C    030054    0100   000400
   12334  302E    030056    0840   004100    SAVD   0
   12336  3030    030060    0880   004200    RESD   0
   12338  3032    030062    08C0   004300    ADDI   0,-100
   12340  3034    030064    FF00   177400
   12342  3036    030066    0A00   005000    JMPR   0
   12344  3038    030070    0C00   006000    LDRI   0,0,00FF
   12346  303A    030072    00FF   000377
   12348  303C    030074    0C40   006100    LDDI   0,0500
   12350  303E    030076    0500   002400
   12352  3040    030100    0C80   006200    LDSP   4000
   12354  3042    030102    4000   040000
   12356  3044    030104    0E00   007000    WATE
   12358  3046    030106    1000   010000    LDDZ   0
   12360  3048    030110    1800   014000    LDDP   0

   DEC    HEX     OCT     HEX     OCT
   ADDR   ADDR    ADDR    VALUE   VALUE
   ----   ----    ----    -----   -----
   12362  304A    030112    2000   020000    LDXA   0
   12364  304C    030114    2800   024000    LDXR   0
   12366  304E    030116    3000   030000    DRXA   0
   12368  3050    030120    3800   034000    DRXR   0
   12370  3052    030122    4000   040000    DRYA   0
   12372  3054    030124    4800   044000    DRYR   0
   12374  3056    030126    5000   050000    MVXA   0
   12376  3058    030130    5800   054000    MVXR   0
   12378  305A    030132    6000   060000    MVYA   0
   12380  305C    030134    6800   064000    MVYR   0
   12382  305E    030136    7000   070000    LDKX   0
   12384  3060    030140    7800   074000    DRKY   0
   12386  3062    030142    8000   100000    DRSR   0,0
   12388  3064    030144    8040   100100    MVSR   0,0
   12390  3066    030146    C000   140000    PPLR   0,0
   12392  3068    030150    C040   140100    LDTI   0
   12394  306A    030152    9FC1   117701    CHAR   A
   12396  306C    030154    C2C3   141303    TEXT   BC
   12398  306E    030156    0E40   007100    INIT
   12400  3070    030160    0DC0   006700    CLRM
   12402  3072    030162    0E80   007200    MODE   0
FUP/G     09/03/84     17:28:22
```

```
FUP/G     09/03/84     17:21:03
   DEC    HEX     OCT      HEX     OCT
   ADDR   ADDR    ADDR     VALUE   VALUE
   ----   ----    ----     -----   -----
   12288  3000    030000     0000   000000   HALT
   12290  3002    030002     0200   001000   JUMP   1000
   12292  3004    030004     1000   010000
   12294  3006    030006     0200   001000   JUMP   1500
   12296  3008    030010     1500   012400
   12298  300A    030012     0240   001100   JRMP   100
   12300  300C    030014     0100   000400
   12302  300E    030016     0280   001200   JMPZ   2000
   12304  3010    030020     2000   020000
   12306  3012    030022     0280   001200   *JMPZ  2000
   12308  3014    030024     A000   120000
   12310  3016    030026     02C0   001300   JPRZ   100
   12312  3018    030030     0100   000400
   12314  301A    030032     0400   002000   JMPM   1500
   12316  301C    030034     1500   012400
   12318  301E    030036     0440   002100   CALL   9000
   12320  3020    030040     9000   110000
   12322  3022    030042     0480   002200   CALR   1000
   12324  3024    030044     1000   010000
   12326  3026    030046     04C0   002300   RTRN
   12328  3028    030050     0600   003000   IZPR
   12330  302A    030052     0800   004000   *LINK  0100
   12332  302C    030054     8100   100400
   12334  302E    030056     0842   004102   SAVD   2
   12336  3030    030060     0881   004201   RESD   1
   12338  3032    030062     08C3   004303   ADDI   3,100
   12340  3034    030064     0100   000400
   12342  3036    030066     0A00   005000   JMPR   0
   12344  3038    030070     0C00   006000   LDRI   0,0,00FF
   12346  303A    030072     00FF   000377
   12348  303C    030074     0C40   006100   LDDI   0,0500
   12350  303E    030076     0500   002400
   12352  3040    030100     0C80   006200   LDSP   4000
   12354  3042    030102     4000   040000
   12356  3044    030104     0E00   007000   WATE
   12358  3046    030106     1000   010000   LDDZ   0
   12360  3048    030110     1800   014000   LDDP   0

   DEC    HEX     OCT      HEX     OCT
   ADDR   ADDR    ADDR     VALUE   VALUE
   ----   ----    ----     -----   -----
   12362  304A    030112     2000   020000   LDXA   0
   12364  304C    030114     2800   024000   LDXR   0
   12366  304E    030116     3000   030000   DRXA   0
   12368  3050    030120     3800   034000   DRXR   0
   12370  3052    030122     4000   040000   DRYA   0
   12372  3054    030124     4800   044000   DRYR   0
   12374  3056    030126     5000   050000   MVXA   0
   12376  3058    030130     5800   054000   MVXR   0
   12378  305A    030132     6000   060000   MVYA   0
   12380  305C    030134     6800   064000   MVYR   0
   12382  305E    030136     7000   070000   LDKX   0
   12384  3060    030140     7800   074000   DRKY   0
   12386  3062    030142     8000   100000   DRSR   0,0
   12388  3064    030144     8040   100100   MVSR   0,0
   12390  3066    030146     C000   140000   PPLR   0,0
   12392  3068    030150     C040   140100   LDTI   0
   12394  306A    030152     9FC1   117701   CHAR   A
   12396  306C    030154     C3C2   141702   TEXT   CB
   12398  306E    030156     0E40   007100   INIT
   12400  3070    030160     0DC0   006700   CLRM
   12402  3072    030162     0E00   007200   MODE   0
FUP/G     09/03/84     17:21:39
```

APPENDIX 2

A2.1 Listings of ABL editor :

- Module V7# (ABL editor)
- Module V8# (common subroutines)

A2.2 Listing of ABL interpretor :

- Module T4#

```
V7#88S     19/#3/84     16:#7:33

  98#A OFF LINE ASSEMBLER                                          SHEET    1
                      1  *       ABL EDITOR
                      2  *
                      3  *       TI 98#A
                      4  *
                      5  *       K. VO
                      6  *
                      7  *       REVISION #
                      8  *
                      9          IDT  V7#
                     1#  *
                     11  *
                     12  *            THE ABL EDITOR ACCEPTS SPECIFICATION FOR ABSTRACT
                     13  *            PROGRAM (SEE SHEET 3) AND CREATES A CORRESPONDING
                     14  *            SOURCE FILE IN TI-98# ASSEMBLY LANGUAGE WHICH CAN BE
                     15  *            RUN BY THE ABL INTERPRETOR (MODULE T4#).
                     16  *
                     17  *            LUN ASSIGNMENT :
                     18  *
                     19  *            LUN # : CONSOLE
                     2#  *                1 : SCRATCH FILE TO STORE ACTIONSOF ABL MACHINES
                     21  *                3 : SCRATCH FILE TO TEMPORARILY STORE PROGRAM
                     22  *                4 : SCRATCH FILE TO WHICH LUN 3 WILL BE COPED
                     23  *                      AT THE END OF SUCCESFUL ABL PROGRAM CREATION
                     24  *
                     25  *
                     26          REF   RFC,WTC,GETCHA,DEC2BI,ABLENT,BI2HAS,HDER#,ACLS##
                     27          REF   M12,ACLBSS,BI2DAS,M13,HLABL1
                     28          DEF   IPLNX,IPLNS,IPLN,OPBUF
                     29  *
              ####   3#  A       EQU   #
              ###1   31  E       EQU   1
              ###2   32  X       EQU   2
              ###3   33  M       EQU   3
              ###4   34  S       EQU   4
              ###5   35  L       EQU   5
              ###6   36  B       EQU   6
              ###7   37  P       EQU   7
                     38  *
              ###1   39  BR      EQU   1
              ###3   4#  XB      EQU   3
              CAE#   41  SVC     OPD   >CAE#,3
              #A#D   42  LFCR    EQU   >#A#D
              A##D   43  BBCR    EQU   >A##D
              4###   44  EMPTY   EQU   >4###
                     45  *
              ###1   46  LU1     EQU   1
              ###3   47  LU3     EQU   3
              ###4   48  LU4     EQU   4
```

```
                        49          PEJ
                        50  *
                        51  *
   0000  1800           52          @LDM =PROGNO        M REG. CONTAINS ADDRESS OF PROGRAM
P  0001  0005
   0002  7400           53          @BRL ABLENT         CALL ABL INTERP. TO EXECUTE ABL EDITOR PROGRAM
         0004
E  0003  0000
   0004  CAE5           54          SVC  5              TERMINATE ABL EDITOR UTILITY
```

```
98#A OFF LINE ASSEMBLER                                    SHEET    3
                    55          PEJ
                    56  *
                    57  *       SPECIFICATION FOR ABL EDITOR :
                    58  *
                    59  *       ALB EDITOR
                    6#  *
                    61  *       ENTER FILE NAME : MMI
                    62  *       ENTER MACHINE :
                    63  *       1 >INIT      INITIALIZATION
                    64  *       2 >D2BCON    DEC-BIN CONVERSION
                    65  *       3 >.
                    66  *         .
                    67  *         .
                    68  *       N >$
                    69  *       PROGRAM NUMBER >
                    7#  *       ENTER NUMBER OF CLUSTERS : 2
                    71  *       ENTER NUMBER OF ALTERNATIVES : 1#
                    72  *       ENTER PROGRAM :
                    73  *       >*           INITIALIZATION & CLASSIFICATION
                    74  *       >1.1 //1/2
                    75  *       >*           L,H - FORMAT 1
                    76  *       >2.1 /#,3,-4/2,3,2,4/#
                    77  *         .
                    78  *         .
                    79  *         .
                    8#  *         .
                    81  *       >$
                    82  *
```

```
98#A OFF LINE ASSEMBLER                                                SHEET    4
                 83                PEJ
                 84  *     #  1  2  3  4  5  6  7  8  9  1#  12  14  16  18
                 85  *                                    11  13  15  17  19
                 86  *
                 87  *C1   V  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - EDITOR INITIALIZATION
                 88  *C2   -  V  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - READ ACTION I/P LINE
                 89  *C3   -  -  V  V  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - PROCESS    "
                 9#  *C4   -  -  -  -  V  -  -  -  -  -  -  -  -  -  -  -  -  -  - INIT. FOR ALT. IP LINE
                 91  *C5   -  -  -  -  -  V  V  V  -  -  -  -  -  -  -  -  -  -  - ANALYZE ALT. INPUT LINE
                 92  *C6   -  -  -  -  -  -  -  -  V  V  -  -  -  -  -  -  -  -  - PROCESS CLUSTER SPEC.
                 93  *C7   -  -  -  -  -  -  -  -  -  V  V  V  V  V  V  -  -  -  - PROCESS PRED. SPEC.
                 94  *C8   -  -  -  -  -  -  -  -  -  V  -  -  -  -  V  V  V  -  - PROCESS ACTION SPEC.
                 95  *C9   -  -  -  -  -  -  -  -  -  V  -  -  -  -  -  -  -  V  - PROCESS NXT CLTR
                 96  *C1#  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  V CLOSING
                 97  *
                 98  *N    2  3  2  4  5  4  A  6  7  4  7  7  8  8  8  8  9  9  4  # NEXT CLUSTER
                 99  *
                1##  *P#   -  -  -  -  -  Y  N  N  -  -  -  -  -  -  -  -  -  -  - '*' IS 1ST CHAR
                1#1  *P1   -  -  N  Y  -  N  Y  N  -  -  -  -  -  -  -  -  -  -  - '$'  "   "   "
                1#2  *P2   -  -  -  -  -  -  -  -  -  -  Y  Y  N  N  -  Y  N  -  - ','  "   "   "
                1#3  *P3   -  -  -  -  -  -  -  -  -  -  N  Y  N  Y  -  -  -  -  - '-'  "   "   "
                1#4  *P4   -  -  -  -  -  -  -  -  -  -  N  N  Y  Y  -  N  Y  -  - '/'  "   "   "
                1#5  *P5   -  -  -  -  -  -  -  -  -  -  N  N  N  N  Y  N  N  Y  - CPTR=LAPTR (NULL)
                1#6  *P6   -  -  -  -  N  N  N  -  N  Y  N  N  N  N  N  N  N  N  - INVALID CHAR.
                1#7  *
                1#8  *     #  1  2  3  4  5  6  7  8  9  1#  12  14  16  18  2#
                1#9  *                                    11  13  15  17  19
                11#  *     1  .  .  .  .  .  #  .  .  .  .  .  .  .  .  .  .  .  . INIT  :EDI INIT.
                111  *     2  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  . FNAME :GET FILNAME, OPEN
                112  *     .  .  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  . STACT :STORE ACTION
                113  *     .  .  .  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  . RPI   :READ PROGINF.
                114  *     .  .  .  2  .  .  .  .  .  .  .  .  .  .  .  .  .  .  . PINIT : INIT'N FOR PROG.
                115  *     .  .  .  .  1  .  .  .  .  .  .  .  .  .  .  .  .  .  . RPTRS :RESET CPTR,LAPTR
                116  *     .  .  .  .  .  1  .  .  .  .  .  .  .  .  .  .  .  .  . WCL   :WRITE COMMENT LINE
                117  *     .  .  .  .  .  .  1  .  2  .  .  .  .  .  .  .  .  .  . PFR   :PROMPT FOR REPLAC.
                118  *     .  .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .  . RESTOR:RESTORE POINTERS
                119  *     .  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  . RAIL  :READ ACTION I/P LINE
                12#  *     .  .  .  .  2  .  .  2  .  .  .  .  .  .  .  .  .  .  . RIL   :READ INPUT LINE
                121  *     .  2  .  .  3  .  .  3  .  .  .  .  .  .  .  .  .  .  . AFC   :ANALYZE 1ST CHAR
                122  *     .  .  .  .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  . ICPTR :INCREMENT CPTR
                123  *     .  .  .  .  4  .  .  .  .  .  .  .  .  .  .  .  .  .  . SAVEP :SAVE POINTERS
                124  *     .  .  .  .  .  .  .  1  .  .  .  .  .  .  .  .  .  .  . RPTRS :RESET PTRS TO CLTR
                125  *     .  .  .  .  .  .  .  2  .  1  2  1  1  .  1  1  .  .  . D2BCON:DEC-BIN CONV.
                126  *     .  .  .  .  .  .  .  3  .  .  .  .  .  .  .  .  .  .  . SACLS :SAVE CLUSTER #
                127  *     .  .  .  .  .  .  .  .  .  2  .  2  .  .  .  .  .  .  . SBYNP :SET BIT IN YNPBUF
                128  *     .  .  .  .  .  .  .  .  .  3  3  3  2  .  .  .  .  .  . SBMPR :SET BIT IN MPRBUF
                129  *     .  .  .  .  .  .  .  .  .  .  .  4  3  .  .  .  .  .  . WYN&M :WRITE YNPBUF & MPRBU
                13#  *     .  .  .  .  .  .  .  .  .  .  .  .  .  .  2  2  .  .  . PAILN :PLACE ACTION
                131  *     .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  3  .  .  . PT    :PLACE TERMINATOR
                132  *     .  .  .  .  .  .  .  .  4  .  4  4  5  4  1  3  4  .  . PCAL  :POINT CPTR AFTER LA
                133  *     .  .  .  .  .  .  .  .  5  .  5  5  6  5  2  4  5  .  . PNSC  :NXT SEP.& CLASS.
                134  *     .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1  . WNCN  :WRITE NEXT CLTR#
                135  *     .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  2  . WCYM  :WRITE CL,YN,MP
                136  *     .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  3  . SABICB:SET ALT BIT IN CLS B
                137  *     .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1 WCM   :WRITE TO CLUSTER MAT
                138  *     .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  2 CPF   :CREATE PERMANENT FIL
                139  *
```

```
98@A OFF LINE ASSEMBLER                                          SHEET    5
                14@  *    @ 1 2 3 4 5 6 7 8 9 1@  12  14  16  18
                141  *                        11  13  15  17  19
```

```
                142          PEJ
                143  *
                144  *
                145  *       ALT #  :  ABL EDITOR INITIALIZATION
                146  *       ALT 1  :  READ MACHINE I/P LINE
                147  *       ALT 2  :  ACTION I/P LINE (SUBROUTINE NAME)
                148  *       ALT 3  :  MACHINE TERMINATION - ABL PROGRAM INIT'N
                149  *       ALT 4  :  READ & ANALYZE I/P LINE FOR ALTER. SPEC.
                150  *       ALT 5  :  COMMENT I/P LINE
                151  *       ALT 6  :  PROGRAM I/P TERMINATION
                152  *       ALT 7  :  ALTER. SPEC. I/P LINE
                153  *       ALT 8  :  PROCESS CLUSTER SPECIFICATION
                154  *       ALT 9  :  INVALID I/P CHARACTER
                155  *       ALT 10 :  POSITIVE PREDICATE
                156  *       ALT 11 :  NEGATIVE PREDICATE
                157  *       ALT 12 :  POSITIVE PREDICATE, LAST PREDICATE
                158  *       ALT 13 :  NEGATIVE
                159  *       ALT 14 :  NO SPECIFICATION (CPTR=LAPTR)
                160  *       ALT 15 :  ACTION NUMBER
                161  *       ALT 16 :  LAST ACTION
                162  *       ALT 17 :  NO SPECIFICATION (CPTR=LAPTR)
                163  *       ALT 18 :  PROCESS NEXT CLUSTER
                164  *       ALT 19 :  CLOSING
```

```
                    165          PEJ
                    166  *
                    167  *
                    168  *            ABL PROGRAM SECTION
                    169  *            ===================
                    17#  *
                    171  *
  ####5  ####3      172  PROGNO DATA 3                    PROGRAM NO.
                    173  *
  ####6  ####       174  PCLUST DATA #                    PRESENT CLUSTER
  ####7  ####       175  PALT   DATA #                       *    ALT
  ####8  ####       176  PACT   DATA #                       *    ACTION
  ####9  ####       177  PACTOF DATA #                       *    ACTION OFFSET IN ALTERNATIVE
  ####A  ####       178  ACLST@ DATA #                    PTR TO ACTION LIST FORALT BEING EXE'D
                    179  *
  ####B  ####       18#  SVECTR DATA #                    STATE VECTOR
                    181  *
P ####C  ##13       182  CLSMA@ DATA CLSMAT        POINTER TO CLUSTER MATRIX
P ####D  ##27       183  NCLMA@ DATA NCLMAT           *     *  NEXT-CLUSTER VECTOR
P ####E  ##3B       184  YNPMA@ DATA YNPMAT           *     *  PREDICATE MATRIX
P ####F  ##4F       185  MPRMA@ DATA MPRMAT           *     *  PRED MASK MATRIX
P ##1#   ##63       186  APTR#@ DATA APTR#            *     *  ACTION LIST POINTERS
                    187  *
  ##11   ####       188  BREG   DATA #                    (B)=ADDRESS OF PREVIOUS ABSTR. PROGRAM
  ##12   ####       189  LREG   DATA #                    (L)=RETURN ADDRESS
                    19#  *
                    191  *            CLUSTER MATRIX  BITS#-31
                    192  *
  ##13   8###       193  CLSMAT DATA >8###,#       C1
  ##14   ####
  ##15   4###       194          DATA >4###,#       C2
  ##16   ####
  ##17   3###       195          DATA >3###,#       C3
  ##18   ####
  ##19   #8##       196          DATA >#8##,#       C4
  ##1A   ####
  ##1B   #7##       197          DATA >#7##,#       C5
  ##1C   ####
  ##1D   ##C#       198          DATA >##C#,#       C6
  ##1E   ####
  ##1F   ##7E       199          DATA >##7E,#       C7
  ##2#   ####
  ##21   ##41       2##          DATA >##41,>C###   C8
  ##22   C###
  #023   ##4#       2#1          DATA >##4#,>2###   C9
  #024   2###
  #025   ####       2#2          DATA >####,>1###   C1#
  #026   1###
                    2#3  *
                    2#4  *            NEXT CLUSTER
                    2#5  *
  #027   ###2       2#6  NCLMAT DATA 2             ALT #
  #028   ###3       2#7          DATA 3                 1
  #029   ###2       2#8          DATA 2                 2
  #02A   ###4       2#9          DATA 4                 3
  #02B   ###5       21#          DATA 5                 4
  #02C   ###4       211          DATA 4                 5
```

```
0002D  0006A  212          DATA 10                6
0002E  0006C  213          DATA 6                 7
0002F  0006E  214          DATA 7                 8
00030  00070  215          DATA 4                 9
00031  00072  216          DATA 7                 10
00032  00074  217          DATA 7                 11
00033  00076  218          DATA 8                 12
00034  00078  219          DATA 8                 13
00035  0007A  220          DATA 8                 14
00036  0007C  221          DATA 8                 15
00037  0007E  222          DATA 9                 16
00038  00080  223          DATA 9                 17
00039  00082  224          DATA 4                 18
0003A  00084  225          DATA 0                 19
               226  *
               227  *            Y/N PREDICATE MATRIX
               228  *
0003B  00086  229  YNPMAT  DATA >0000        ALT 0
0003C  00088  230          DATA >0000             1
0003D  0008A  231          DATA >0000             2
0003E  4000   232          DATA >4000             3
0003F  0000   233          DATA >0000             4
00040  8000   234          DATA >8000             5
00041  4000   235          DATA >4000             6
00042  0000   236          DATA >0000             7
00043  0000   237          DATA >0000             8
00044  0200   238          DATA >0200             9
00045  2000   239          DATA >2000             10
00046  3000   240          DATA >3000             11
00047  0800   241          DATA >0800             12
00048  1800   242          DATA >1800             13
00049  0400   243          DATA >0400             14
0004A  2000   244          DATA >2000             15
0004B  0800   245          DATA >0800             16
0004C  0400   246          DATA >0400             17
0004D  0000   247          DATA >0000             18
0004E  0000   248          DATA >0000             19
               249  *
               250  *            MASK FOR PREDICATE MATRIX
               251  *
0004F  0000   252  MPRMAT  DATA >0000        ALT 0
00050  0000   253          DATA >0000             1
00051  4000   254          DATA >4000             2
00052  4000   255          DATA >4000             3
00053  0200   256          DATA >0200             4
00054  C200   257          DATA >C200             5
00055  C200   258          DATA >C200             6
00056  C000   259          DATA >C000             7
00057  0200   260          DATA >0200             8
00058  0200   261          DATA >0200             9
00059  3E00   262          DATA >3E00             10
0005A  3E00   263          DATA >3E00             11
0005B  3E00   264          DATA >3E00             12
0005C  3E00   265          DATA >3E00             13
0005D  0600   266          DATA >0600             14
0005E  2C00   267          DATA >2C00             15
0005F  2C00   268          DATA >2C00             16
```

```
   0066  0400   269          DATA >0400            17
   0061  0200   270          DATA >0200            18
   0062  0000   271          DATA >0000            19
                272  *
                273  *            ACTION MATRIX
                274  *
P  0063  0077   275  APTR0   DATA ACLST0
P  0064  007A   276          DATA ACLST1
P  0065  007D   277          DATA ACLST2
P  0066  007F   278          DATA ACLST3
P  0067  0082   279          DATA ACLST4
P  0068  0088   280          DATA ACLST5
P  0069  008A   281          DATA ACLST6
P  006A  0088   282          DATA ACLST7
P  006B  008C   283          DATA ACLST8
P  006C  0092   284          DATA ACLST9
P  006D  0095   285          DATA ACLT10
P  006E  009B   286          DATA ACLT11
P  006F  00A1   287          DATA ACLT12
P  0070  00A7   288          DATA ACLT13
P  0071  00AD   289          DATA ACLT14
P  0072  00B0   290          DATA ACLT15
P  0073  00B5   291          DATA ACLT16
P  0074  00BB   292          DATA ACLT17
P  0075  00BE   293          DATA ACLT18
P  0076  00C2   294          DATA ACLT19
                295  *
   0077         296  ACLST0  BSS  0          INITIALIZATION
P  0077  0133   297          DATA INIT             ABL EDITOR INIT'N
P  0078  0157   298          DATA FNAME            GET FILE NAME & DO A/O
   0079  0000   299          DATA 0
                300  *
   007A         301  ACLST1  BSS  0          READ MACHINE I/P LINE
P  007A  018D   302          DATA RAIL             READ ACTION I/P LINE
P  007B  027C   303          DATA AFC              ANALYZE FIRST CHARACTER
   007C  0000   304          DATA 0
                305  *
   007D         306  ACLST2  BSS  0          ACTION I/P LINE (SUB NAME)
P  007D  01A8   307          DATA STACT            STORE ACTION INTO TEMP. FILE
   007E  0000   308          DATA 0
                309  *
   007F         310  ACLST3  BSS  0          MACHINE TERMINATION
P  007F  01B6   311          DATA RPI              READ INFO. ON PROGRAM SIZE
P  0080  02B6   312          DATA PINIT            INIT'N FOR PROGRAM CREATION
   0081  0000   313          DATA 0
                314  *
   0082         315  ACLST4  BSS  0          READ & ANALYZE I/P LINE FOR ALT SPEC
P  0082  01F4   316          DATA CLRSV            CLEAR STATE VERTOR
P  0083  01ED   317          DATA RPTRS            RESET CPTR, LAPTR
P  0084  01F7   318          DATA RIL              READ ALT I/P LINE
P  0085  027C   319          DATA AFC              ANALYZE 1ST CHAR
P  0086  040A   320          DATA SAVEP            SAVE POINTERS
   0087  0000   321          DATA 0
                322  *
   0088         323  ACLST5  BSS  0          WRITE COMMENT LINE
P  0088  02DD   324          DATA WCL              WRITE COMMENT LINE
   0089  0000   325          DATA 0
```

```
                    326  *
  00A               327  ACLST6 BSS  0          PROGRAM I/P TERMINATION
  00A  0000         328         DATA 0
                    329  *
  00B               330  ACLST7 BSS  0          ALT SPEC I/P LINE
  00B  0000         331  *      DATA 0
                    332  *
  00C               333  ACLST8 BSS  0          PROCESS CLUSTER SPEC.
P 00C  01ED         334         DATA RPTRS            RESET POINTERS TO CLUSTER. POS.
P 00D  02EB         335         DATA D2BCON           CONVERT TO BINARY
P 00E  0312         336         DATA SACLS            SAVE CLUSTER
P 00F  02CE         337         DATA PCL              POINT CPTR AFTER LAPTR
P 010  029F         338         DATA PNSC             POINTER TO NEXT SEP & CLASSIFY
  011  0000         339         DATA 0
                    340  *
  012               341  ACLST9 BSS  0          INVALID I/P CHARACTER
P 012  0410         342         DATA RESTOR           RESTORE POINTERS
P 013  02D5         343         DATA PFR              PROMPT FOR REPLACEMENT
  014  0000         344         DATA 0
                    345  *
  015               346  ACLT10 BSS  0          POSITIVE PREDICATE
P 015  02EB         347         DATA D2BCON           DEC-BIN CONVERSION
P 016  02F9         348         DATA SBYNP            SET BIT IN YNP BUFFER
P 017  0316         349         DATA SBMPR            "  "  "  MPR  "
P 018  02CA         350         DATA PCAL             POINT CPTR AFTER LAPTR
P 019  029F         351         DATA PNSC             POINT TO NEXT SEP. & CLASSIFY
  01A  0000         352         DATA 0
                    353  *
  01B               354  ACLT11 BSS  0          NEGATIVE PREDICATE
P 01B  02D3         355         DATA ICPTR            INCREMENT CPTR
P 01C  02EB         356         DATA D2BCON           DEC-BIN CONVERSION
P 01D  0316         357         DATA SBMPR            SET BIT IN MASK MATRIX
P 01E  02CA         358         DATA PCAL             POINT CPTR AFTER LAPTR
P 01F  029F         359         DATA PNSC             POINT TO NEXT SEP. & CLASSIFY
  020  0000         360         DATA 0
                    361  *
  021               362  ACLT12 BSS  0          POSITIVE PREDICATE,LAST PREDICATE
P 021  02EB         363         DATA D2BCON           DEC-BIN CONVERSION
P 022  02F9         364         DATA SBYNP            SET BIT IN YNP BUFFER
P 023  0316         365         DATA SBMPR            "  "  "  MPR  "
P 024  02CA         366         DATA PCAL             POINT CPTR AFTER LAPTR
P 025  029F         367         DATA PNSC             POINT TO NEXT SEP. & CLASSIFY
  026  0000         368         DATA 0
                    369  *
  027               370  ACLT13 BSS  0          NEGATIVE PREDICATE,LAST PREDICATE
P 027  02D3         371         DATA ICPTR            INCREMENT CPTR
P 028  02EB         372         DATA D2BCON           DEC-BIN CONVERSION
P 029  0316         373         DATA SBMPR            SET BIT IN MASK MATRIX
P 02A  02CA         374         DATA PCAL             POINT CPTR AFTER LAPTR
P 02B  029F         375         DATA PNSC             POINT TO NEXT SEP. & CLASSIFY
  02C  0000         376         DATA 0
                    377  *
  02D               378  ACLT14 BSS  0          NO PREDICATE SPECIFICATION
P 02D  02CA         379         DATA PCAL             POINT CPTR AFTER LAPTR
P 02E  029F         380         DATA PNSC             POINT TO NEXT SEP. & CLASSIFY
  02F  0000         381         DATA 0
                    382  *
```

```
   0B0B          383  ACLT15 BSS  0
P  0B0C  02EB    384         DATA D2BCON
P  0B0D  031F    385         DATA PAILN
P  0B0E  02CA    386         DATA PCAL
P  0B0F  029F    387         DATA PNSC
   0B10  0000    388         DATA 0
                 389  *
   0B11          390  ACLT16 BSS  0
P  0B12  02EB    391         DATA D2BCON
P  0B13  031F    392         DATA PAILN
P  0B14  03D8    393         DATA PT
P  0B15  02CA    394         DATA PCAL
P  0B16  029F    395         DATA PNSC
   0B17  0000    396         DATA 0
                 397  *
   0B18          398  ACLT17 BSS  0
P  0B19  03D8    399         DATA PT
P  0B1A  02CA    400         DATA PCAL
   0B1B  0000    401         DATA 0
                 402  *
   0B1C          403  ACLT18 BSS  0
P  0B1D  041C    404         DATA WNCN
P  0B1E  046A    405         DATA WYM
P  0B1F  0447    406         DATA SABICB
   0B20  0000    407         DATA 0
                 408  *
   0B21          409  ACLT19 BSS  0
P  0B22  03B0    410         DATA WCM
P  0B23  0495    411         DATA CPF
   0B24  0000    412         DATA 0
```

ACTION NUMBER
- DEC-BIN CONVERSION
- WRITE SUB NAME TO ACTION LIST
- POINT CPTR AFTER LAPTR
- POINT TO NEXT SEP. & CLASSIFY

LAST ACTION NUMBER
- DEC-BIN CONVERSION
- WRITE SUB NAME TO ACTION LIST
- PLACE TERMINATOR
- POINT CPTR AFTER LAPTR
- POINT TO NEXT SEP. & CLASSIFY

NO ACTION SPECIFICATION
- PLACE TERMINATOR
- PLACE CPTR AFTER LAPTR

PROCESS NEXT CLUSTER
- WRITE NEXT CLUSTER NUMBER
- WRITE YN, MASK MATRIX
- SET ALT BIT IN CLS. BUFFER

CLOSING
- WRITE TO CLUSTER MATRIX
- CREATE PERMANENT FILE KV

```
                    413          PEJ
                    414  *
                    415  *            DATA SECTION
                    416  *            ------------
                    417  *
          ###F      418  HSIZE   EQU  15          SIZE OF HEADER'
          ##24      419  SRLNTH  EQU  36          'S' TYPE RECORD LENGTH
          ##23      42#  LLNTH   EQU  7#/2        LENGTH OF SOURCE LINE,EXCLUDING CR
                    42L  *
          ####      422  RFC@    DATA RFC
E ##C5    ####
          ###1      423  WTC@    DATA WTC
E ##C6    ####
          ###2      424  GETCH@  DATA GETCHA
E ##C7    ####
          ###3      425  DE2BI@  DATA DEC2BI
E ##C8    ####
          ###5      426  BI2HA@  DATA BI2HAS
E ##C9    ####
          ###A      427  BI2DA@  DATA BI2DAS
E ##CA    ####
  ##CB    ####      428  NOCLS   DATA #         # OF CLUSTERS
  ##CC    ####      429  NOALTS  DATA #         # OF ALTERNATIVES
  ##CD    ####      43#  NOACTS  DATA #         # OF ACTIONS
  ##CE    ####      431  NOCHRS  DATA #         # OF CHARACTERS IN I/P BUFFER
  ##CF    ####      432  ALT#    DATA #         THE ALTERNATIVE CURRENTLY PROCESSED
  ##D#    ####      433  CLUS#   DATA #         THE CLUSTER WHICH THE CURRENT ALT BELONG
                    434  *
  ##D1    ###F      435  COFFS   DATA HSIZE     CLSMAT OFFSET
  ##D2              436  NOFFS   BSS  1         NCLMAT OFFSET, FROM SOURCE LINE NO 1
  ##D3              437  YOFFS   BSS  1         YNPMAT  "
  ##D4              438  MOFFS   BSS  1         MPRMAT  "
  ##D5              439  AOFFS   BSS  1         APTR#   "
  ##D6              44#  ACOFFS  BSS  1         ACLST1  "
  ##D7              441  OFFS    BSS  #
                    442  *
                    443  *       BINARY VALUES OF ALT I/P SPECIFICATIONS
                    444  *
  ##D7              445  NCLBUF  BSS  1         BUFFER     NCLMAT
  ##D8    ####      446  YNPBUF  DATA #         BUFFER FOR YNPMAT
  ##D9    ####      447  MPRBUF  DATA #             "      "  MPRMAT ENTRY
P ##DA    ##DB      448  ACTNS@  DATA ACTNS
  ##DB              449  ACTNS   BSS  16        ACTIONS FOR ALTERNATIVE
  ##EB              45#  ACTNSE  BSS  #
                    451  *
  ##EB    ####      452  CLSBUF  DATA #,#        CLS 1
  ##EC    ####
  ##ED    ####      453          DATA #,#            2
  ##EE    ####
  ##EF    ####      454          DATA #,#            3
  ##F#    ####
  ##F1    ####      455          DATA #,#            4
  ##F2    ####
  ##F3    ####      456          DATA #,#            5
  ##F4    ####
  ##F5    ####      457          DATA #,#            6
  ##F6    ####
```

```
00F7 0000    458          DATA 0,0                7
00F8 0000                                         
00F9 0000    459          DATA 0,0                8
00FA 0000                                         
00FB 0000    460          DATA 0,0                9
00FC 0000                                         
00FD 0000    461          DATA 0,0                10
00FE 0000                                         
00FF 0000    462          DATA 0,0                11
0100 0000                                         
0101         463 CPTR     BSS  1          CURRENT POINTER, BYTE OFFSET
0102         464 LAPTR    BSS  1          LOOK-AHEAD POINTER, BYTE OFFSET
0103         465 OCPTR    BSS  1          OLD CPTR
0104         466 OLAPTR   BSS  1          OLD LAPTR
0105         467 BV       BSS  1          CONVERTED BINARY VALUE
             468 *
P 0106 0107  469 IPLNX    DATA S+1
0107 4050    470          DATA EMPTY+80
0108 0000    471          DATA 0,0
0109 0000                                         
010A         472 IPLNS    BSS  0
010A         473 IPLN     BSS  40         INPUT BUFFER
0132         474 IPLNE    BSS  0
0132 BA8D    475          DATA LFCR
             476 *
```

```
                    477          PEJ
                    478  *
                    479  *            ACTION SECTION
                    480  *            ==============
                    481  *
P       0005        482          BRS  PROGNO
                    483  *
                    484  *       -- ABL EDITOR INITIALIZATION
                    485  *
  0133              486  INIT    BSS  0
  0133  C550        487          RMO  L,A
  0134  800F        488          STA  INITRE
                    489  *
  0135  1F00        490          LDM  =0              SKIP A LINE
  0136  748F        491          BRL  *WTC0
  0137  1F02        492          LDM  =2              'ABL EDITOR'
  0138  748D        493          BRL  *WTC0
  0139  1F00        494          LDM  =0            ; SKIP 2 LINES
  013A  7488        495          BRL  *WTC0
  013B  1F00        496          LDM  =0
  013C  7489        497          BRL  *WTC0
                    498  *
                    499  *            OPEN FILES
                    500  *
  013D  1800        501          @LDM =PRBLU1         OPEN TEMP. FILE FOR INPUT ACTIONS
P 013E  0145
  013F  CAE8        502          SVC  8
  0140  1800        503  L1##    @LDM =PRBLU3          .     .     .  -FOR ABL PROGRAM
P 0141  014E
  0142  CAE8        504          SVC  8
  0143  7C00        505  L1#1    BRU  *INITRE         RETURN
                    506  *
  0144              507  INITRE  BSS  1               RETURN ADDR
                    508  *            PRB TO A/O SCRATCH FILE TO TEMPORARYLY
                    509  *            STORE INPUT ACTIONS
  0145  0001        510  PRBLU1  DATA 1               LUN
  0146  0001        511          DATA 1               DISK FILE OP-CODE
  0147  C4D3        512          DATA 'DS'            DISK MNEM.
  0148  D320        513          DATA >D320           O/P FILE TYPE: CATEG (2) & TYPE (N/A)
  0149  C1C2        514          DATA 'ABL/ED'        FILE NAME
  014A  CCAF
  014B  C5C4
  014C  01F4        515          DATA 500             EXTENT IN SECTORS
  014D  0080        516          DATA 128             RECORD LENGTH
                    517  *            PRB TO OPEN SCRATCH FILE TO TEMPORARILY STORE
                    518  *            PROGRAM TO BE CREADTED
  014E  0003        519  PRBLU3  DATA LU3             LUN
  014F  0001        520          DATA 1               DISK FILE OP-CODE
  0150  C4D3        521          DATA 'DS'            DISK MNEM.
  0151  D320        522          DATA >D320           CAT (2), TYPE (N/A)
  0152  A0A0        523          DATA '   00S'       FILE NAME
  0153  A000
  0154  0000
  0155  0000        524          DATA 3000            EXTENT IN SECTOR
  0156  0080        525          DATA 128             RECORD LENGTH IN WORD
```

```
                    526          PEJ .
                    527  *
                    528  *       -- GET FILE NAME IN FORM 'MMI' AND OPEN/ASSIGN
                    529  *          A SCRATCH8 FILE TO LUN 4. THIS SCRATCH FILE WILL
                    530  *          BE COPIED INTO PERMANENT PARTITION AREA UNFER
                    531  *          FILE NAME 'MMIRRS' WHERE RR IS HIGHEST REVISION
                    532  *
  #157              533  FNAME   BSS  #
  #157  C55#        534          RMO  L,A
  #158  8#33        535          STA  FNARET        SAVE RETURN
                    536  *
  #159  1F#3        537  FN2     LDM  =3
  #15A  75C1        538          BRL  *WTC@         'ENTER FILE NAME : '
  #15B  75C#        539          BRL  *RFC@         READ RESPONSE
  #15C  CCA#        540          SNO  M             ERROR ?
  #15D  781A        541          BRU  FN5           YES
  #15E  C53#        542          RMO  M,A           NO
  #15F  6F#3        543          CPA  #3            AT LEAST 3 CHARS ?
  #16#  CD8#        544          SGE
  #161  7816        545          BRU  FN5           NO - INVALID
  #162  B4##        546          @DLD IPLN          YES
P #163  #1#A
  #164  A4##        547          @DST AOPRB+4       PLACE IN A/O PRB
P #165  #17F
  #166  18##        548          @LDM =AOPRB
P #167  #178
  #168  CAE8        549          SVC  8             A/O 'MMI' TO HIGHEST REV.
  #169  ##11        550          LDA  AOPRB
  #16A  38##        551          @AND =>F###
  #16B  F###
  #16C  CC##        552          SZE  A             SUCCESFUL A/O ?
  #16D  78#A        553          BRU  FN5           NO
  #16E  18##        554          @LDM =PRBFA        YES- GET FILE ATTRIBUTES OF LUN 4
P #16F  #184
  #17#  CAE#        555          SVC  #
  #171  1F##        556          LDM  =#            SKIP ONE LINE
  #172  75C1        557          BRL  *WTC@
  #173  1F#4        558          LDM  =4            ' MACHINE SPEC.'
  #174  75C1        559          BRL  *WTC@
  #175  1F#E        560          LDM  =14           'ENTER SUB NAMES'
  #176  75C1        561          BRL  *WTC@
  #177  7C14        562          BRU  *FNARET       RETURN
                    563  *
  #178  1F#8        564  FN5     LDM  =8
  #179  75C1        565          BRL  *WTC@         'ERROR ON FILE OPENING'
  #17A  78DE        566          BRU  FN2           GET ANOTHER FILE NAME
                    567  *
                    568  *
  #17B  ###4        569  AOPRB   DATA 4             LUN
  #17C  ###1        570          DATA 1             DISK FILE OPCODE
  #17D  C4D3        571          DATA 'DS'          DISK MNE
  #17E  D33#        572          DATA >D33#         TYPE (S), CAT (3), TYPE (#)
  #17F              573          BSS  2             FILE NAME
  #181  B#D3        574          DATA '#S'
  #182  #FA#        575          DATA 4###          EXTENT IN SECTORS
  #183  ###8        576          DATA 128           S TYPE REC LENGTH
                    577  *
```

```
                    578  *       PRB TO GET FILE ATTRIBUTES OF LUN 4
 #184  0004         579  PRBFA   DATA 4              LUN
 #185  000A         580          DATA >A             OPCODE
 #186               581          BSS  1              CURRENT REC
 #187               582          BSS  1              RECORD LENGTH
 #188               583          BSS  1              FILE EXTENT
 #189               584          BSS  3              FILE NAME
                    585  *
 #18C               586  FHARET BSS  1
```

```
                    587          PEJ
                    588  *
                    589  *
                    590  *     -- READ ACTION INPUT LINE INTO INPUT BUFFER
                    591  *
  0180              592  RAIL    BSS  0
  0180  C560        593          RMO  L,A
  0181  8016        594          STA  RAIRET       SAVE RETURN
                    595  *
  0182  01C8        596  RAIL2   LDA  NOACTS       ACT #
  0183  C300        597          RIN  A,A          1ST ACTION IS 1
  0184  75C5        598          BRL  *BI2DA@      CONVERT BINARY VALUE TO DEC.
  0185  8C00        599          @STE M13+2        PLACE 2 LS DIGITS IN MESSAGE
        0000
E 0186  0002
  0187  1F0D        600          LDM  -13
  0188  75C1        601          BRL  *WTC@        'NN > '
  0189  B00F        602          DLD  BBBB         INPUT BUFFER IS FIRST BLANKED OUT
  018A  17D8        603          LDX  --40         ..TO CLEAR ALL LEFT OVER INFOS
  018B  A600        604  RAIL5   @DST IPLNE,X
P 018C  0132
  018D  C322        605          RIN  X,X
  018E  40FC        606          BIX  RAIL5
  018F  75C0        607          BRL  *RFC@        READ I/P LINE INTO BUFFER
  0190  CCA3        608          SNO  M            READ ERROR ?
  0191  7803        609          BRU  RAIL7        YES - READ NEW I/P
  0192  C530        610          RMO  M,A          NO
  0193  81C9        611          STA  NOCHRS       SAVE # OF CHARS IN I/P LINE
  0194  7CC3        612          BRU  *RAIRET      RETURN
                    613  *
  0195  1F08        614  RAIL7   LDM  -11
  0196  75C1        615          BRL  *WTC@        'IMPROPER INPUT LINE'
  0197  7B5A        616          BRU  RAIL2
                    617  *
  0198              618  RAIRET  BSS  1            RETURN ADDRESS
  0199  A0A0        619  BBBB    DATA '  '         4 BLANKS
  019A  A0A0
```

```
                    620#          PEJ
                    621  *
                    622  *        -- STORE ACTION SUBROUTINE IN INPUT BUFFER
                    623  *           TO TEMPORARY FILE (LUN 1) AND BUMP NO. OF ACTIONS
                    624  *           EACH ACTION IS STORED IN A RECORD STARTING FROM RECORD 1.
                    625  *
  #1A8              626  STACT   BSS  #
  #1A8  51C8        627           IMO  NOACTS        BUMP # OF ACTIONS
  #1A9  #1C8        628           LDA  NOACTS        ACTION #
  #1AA  8###        629           STA  PRBSA+4       ..AS RECORD #
  #1AB  18##        63#           @LDM =PRBSA
P #1AC  #1AF
  #1AD  CAE#        631           SVC  #             STORE ACTION NAME INTO TEMP FILE
  #1AE  C657        632           RMO  L,P
                    633  *
                    634  *             PRB TO SAVE ACTION NAME INTO TEMP. FILE
  #1AF  ###1        635  PRBSA   DATA LU1           LUN
  #1B#  ###2        636           DATA 2             WRITE
  #1B1  ####        637           DATA #
P #1B2  #1#A        638           DATA IPLN          BUFFER
  #1B3  ####        639           DATA #             RECORD #
  #1B4  ####        64#           DATA #             WORD #
  #1B5  ##23        641           DATA LLNTH         WORD COUNT
```

```
                    642          PEJ
                    643  *
                    644  *       -- READ VALID PROGRAM NUMBER, NUMBER OF CLUSTERS,
                    645  *          AND NUMBER OF ALTERNATIVES
                    646  *
  #1B6              647  RPI     BSS  #
  #1B6  C55#        648          RMO  L,A
  #1B7  8#34        649          STA  RPIRET
                    65#  *
  #1B8  1F##        651          LDM  =#          SKIP A LINE
  #1B9  75C1        652          BRL  *WTC@
  #1BA  1F#F        653  RPI3    LDM  =15         'PROGRAM SPEC.'
  #1BB  75C1        654          BRL  *WTC@
  #1BC  1F1#        655          LDM  =16         'PROGRAM NUMBER'
  #1BD  75C1        656          BRL  *WTC@
  #1BE  75C#        657          BRL  *RFC@       READ INPUT
  #1BF  C53#        658          RMO  M,A         # OF CHARACTERS READ
  #1C#  6F#2        659          CPA  =2
  #1C1  #4##        66#          @LDA IPLNS       GET PROGRAM NUMBER
P #1C2  #1#A
  #1C3  CDA#        661          SNE
  #1C4  78#4        662          BRU  RPI4        2 DIGITS
  #1C5  38##        663          @AND =>FF##      1 DIGIT
  #1C6  FF##
  #1E7  2###        664          @ADD =>##A#      BLANK FILLER
  #1C8  ##A#
  #1C9  84##        665  RPI4    @STA HLABL1+6
        ###C
E #1CA  ####6
                    666  *
  #1CB  1F#6        667  RPI5    LDM  =6          'NUMBER OF CLUSTERS :'
  #1CC  75C1        668          BRL  *WTC@
  #1CD  75C#        669          BRL  *RFC@       READ RESPONSE INTO I/P BUFFER
  #1CE  C53#        67#          RMO  M,A         #OF CHARS
  #1CF  6F#2        671          CPA  =2
  #1D#  CD8#        672          SGE
  #1D1  78F9        673          BRU  RPI5        CAN'T HAVE MORE THAN 99 CLSR'S
  #1D2  C532        674          RMO  M,X         # OF ASCIIS
  #1D3  ####        675          @LDA =IPLN
P #1D4  #1#A
  #1D5  C8C1        676          LLA  1           BYTE POINTER TO BEGINNING OF I/P BUFFER
  #1D6  75C3        677          BRL  *DE2BI@     CONVERT INTO BINARY
  #1D7  CCA2        678          SNO  X           SUCCESFUL CONVERSION ?
  #1D8  78F2        679          BRU  RPI5        NO - TRY TO GET VALID # OF CLUSTERS
  #1D9  81C6        68#          STA  NOCLS       YES- SAVE
                    681  *
  #1DA  1F#7        682  RPI1#   LDM  =7          'NUMBER OF ALTERNATIVES :'
  #1DB  75C1        683          BRL  *WTC@
  #1DC  75C#        684          BRL  *RFC@       READ RESPONSE
  #1DD  C53#        685          RMO  M,A
  #1DE  6F#2        686          CPA  =2
  #1DF  CD8#        687          SGE
  #1E#  78F9        688          BRU  RPI1#       CAN'T HAVE MORE THAN 99 ACTIONS
  #1E1  C532        689          RMO  M,X
  #1E2  ####        69#          @LDA =IPLN
P #1E3  #1#A
  #1E4  C8C1        691          LLA  1           BYTE POINTR TO BEGINNING OF I/P BUFFER
```

```
980A OFF LINE ASSEMBLER                                          SHEET   2#
  #1E5  76C3   692          BRL   *DE2BI@
  #1E6  CCA2   693          SNO   X          SUCCESFUL CONVERSION ?
  #1E7  78F2   694          BRU   RPI1#      NO
  #1E8  81C7   695          STA   NOALTS     YES - THEN SAVE IT
  #1E9  1F#8   696          LDM   =8         'ENTER PROGRAM'
  #1EA  76CJ   697          BRL   *WTC@
  #1EB  7C##   698          BRU   *RPIRET    RETURN
               699  *
  #1EC         7##  RPIRET BSS   1          RETURN ADDR
```

```
                    7#1          PEJ
                    7#2  *
                    7#3  *       -- 1. RESET CURRENT POINTER AND LOOK-AHEAD POINTER
                    7#4  *          TO CLUSTER POSITION
                    7#5  *          2. RESET PREDICATE & MASK BUFFER
                    7#6  *
 #1ED               7#7  RPTRS   BSS  #
 #1ED   #7#1        7#8          LDA  =1
 #1EE   #1FD        7#9          STA  LAPTR
 #1EF   #7##        71#          LDA  =#
 #1F#   #1FC        711          STA  CPTR
 #1F1   #1D3        712          STA  YNPBUF
 #1F2   #1D4        713          STA  MPRBUF
 #1F3   C557        714          RMO  L,P
                    715  *
                    716  *
                    717  *       -- CLEAR STATE VECTOR
                    718  *
 #1F4   #7##        719  CLRSV   LDA  =#
 #1F5   #1#6        72#          STA  SVECTR
 #1F6   C557        721          RMO  L,P
```

```
                    722       PEJ
                    723  *
                    724  *
                    725  *    -- READ ALT. INPUT LINE INTO INPUT BUFFER
                    726  *
  01F7              727  RIL   BSS  0
  01F7  C550        728        RMO  L,A
  01F8  000C        729        STA  RILRET      SAVE RETURN
                    730  *
  01F9  00AC        731        DLD  BBBB        INPUT BUFFER IS FIRST BLANKED OUT
  01FA  17D8        732        LDX  =-40        ..TO CLEAR ALL LEFT OVER INFOS
  01FB  A000        733  RIL5  @DST IPLNE,X
P 01FC  0132
  01FD  C322        734        RIN  X,X
  01FE  40FC        735        BIX  RIL5
  01FF  1F0A        736        LDM  =10
  0200  75C1        737        BRL  *WTC0       ' '
  0201  75C0        738        BRL  *RFC0       READ I/P LINE INTO BUFFER
  0202  C530        739        RMO  M,A         SAVE # OF CHARS IN I/P-LINE
  0203  01C9        740        STA  NOCHRS
  0204  7C00        741        BRU  *RILRET     RETURN
                    742  *
  0205              743  RILRET BSS  1          RETURN ADDRESS
```

```
                    744          PEJ
                    745  *
                    746  *       -- INITIALIZATION :
                    747  *            1. CALCULATES OFFSETS
                    748  *            2. CREATE HEADER
                    749  *            3. CREATE LABELS (NCLMAT,YNPMAT...),
                    750  *               POINTERS (NCL0,YNP0...)
                    751  *
    0206            752  PINIT  BSS  0
                    753  *
                    754  *            CALCULATE OFFSETS
                    755  *
    0206  01C6      756          LDA  NOCLS          # OF CLUSTERS
    0207  C0C1      757          LLA  1              2 LINES/CLUSTER
    0208  270F      758          ADD  =HSIZE         PLUS HEADER SIZE
    0209  2701      759          ADD  =1             FOR 'CLSMAT BSS 0'
    020A  01CD      760          STA  NOFFS          OFFSET TO 'NCLMAT'
    020B  21C7      761          ADD  NOALTS
    020C  2701      762          ADD  =1             FOR 'BSS 0'
    020D  01CE      763          STA  YOFFS            "       'YNPMAT'
    020E  21C7      764          ADD  NOALTS
    020F  2701      765          ADD  =1             FOR 'BSS 0'
    0210  01CF      766          STA  MOFFS            "       MPRMAT'
    0211  21C7      767          ADD  NOALTS
    0212  2701      768          ADD  =1             FOR 'BSS 0'
    0213  01D0      769          STA  AOFFS            "       APTR0
    0214  21C7      770          ADD  NOALTS
    0215  2701      771          ADD  =1             FOR 'BSS 0'
    0216  01D1      772          STA  ACOFFS         OFFSET TO 'ACLS0'
                    773  *
                    774  *       CREATE HEADER
                    775  *
    0217  17F1      776          LDX  =-15           15 LINES IN HEADER
    0218  1800      777  PIN3    @LDM =PRBHDR
P   0219  023C
    021A  CAE0      778          SVC  0              WRITE TO TEMP FILE
    021B  0023      779          LDA  PRBHDR+3
    021C  270A      780          ADD  =10            10 WORDS/LINE
    021D  8021      781          STA  PRBHDR+3       NEXT LABEL
    021E  5021      782          IMO  PRBHDR+4       NEXT RECORD
    021F  40F8      783          BIX  PIN3
                    784  *
                    785  *            CREATE LABELS
                    786  *
    0220  17FA      787          LDX  =-6
    0221  03D2      788  PIN05   LDA  OFFS,X         LINE OFFSET
    0222  8024      789          STA  PRB3+4         ..AS RECORD OFFSET
    0223  C300      790          RIN  A,A            POINT TO LINE AFTER '... BSS 0'
    0224  03D2      791          STA  OFFS,X
    0225  1800      792          @LDM =PRB3
P   0226  0243
    0227  CAE0      793          SVC  0
    0228  001D      794          LDA  PRB3+3         POINT TO NEXT LABEL
    0229  2707      795          ADD  =7
    022A  8018      796          STA  PRB3+3
    022B  40F5      797          BIX  PIN05
                    798  *
```

```
                    799  *       CREATE 'DATA ACLS##,ACLS#1..' AFTER APTR#
                    8##  *
  #22C  #1C7        8#1          LDA  NOALTS
  #22D  8#23        8#2          STA  LINDX
  #22E  #1D#        8#3          LDA  AOFFS
  #22F  8#1E        8#4          STA  PRB4+4
  #23#  18##        8#5          #LDM =PRB4
P #231  #24A
  #232  CAE#        8#6  PIN#7   SVC  #
  #233  5#1A        8#7          IMO  PRB4+4       BUMP TO NEXT RECORD
  #234  #418        8#8          LDA  PRB4+3
  #235  27#7        8#9          ADD  =7
  #236  8#16        81#          STA  PRB4+3
  #237  4819        811          DMT  LINDX
  #238  78F9        812          BRU  PIN#7
                    813  *
  #239  #7##        814          LDA  =#           RESET ALT # TO #
  #23A  81CA        815          STA  ALT#
  #23B  C557        816          RMO  L,P
                    817  *
                    818  *       PRB TO WRITE HEADER LINES
  #23C  ###3        819  PRBHDR  DATA LU3
  #23D  ##2         82#          DATA 2
  #23E  ####        821          DATA #
        ##6         822          DATA HDER#
E #23F  ####
  #24#  ####        823          DATA #            STARTING FROM 1ST LINE (REC #)
  #241  ####        824          DATA #
  #242  ##A         825          DATA 1#
                    826  *
                    827  *            PRB TO CREATE LABELS IN SCRATCH FILE (LUN 3)
  #243  ###3        828  PRB3    DATA LU3          LUN
  #244  ##2         829          DATA 2            WRITE OP-CODE
  #245  ####        83#          DATA #
P #246  #252        831          DATA LABEL        BUFFER
  #247  ####        832          DATA #            RECORD#
  #248  ####        833          DATA #            WORD#
  #249  ##7         834          DATA 7            WORD COUNT
                    835  *
  #24A  ###3        836  PRB#     DATA LU3
  #24B  ##2         837          DATA 2
  #24C  ####        838          DATA #
        ##7         839          DATA ACLS#0
E #24D  ####
  #24E  ####        84#          DATA #            REC#
  #24F  ####        841          DATA #            WD#
  #25#  ##7         842          DATA 7            WD COUNT
                    843  *
  #251              844  LINDX   BSS  1            INDEX
                    845  *
  #252  C3CC        846  LABEL   DATA 'CLSMAT BSS #',BBCR
  #253  D3CD
  #254  C1D4
  #255  A#C2
  #256  D3D3
  #257  A#8#
  #258  A#0D
```

```
990A OFF LINE ASSEMBLER                                          SHEET   25
 0259  CEC3   847          DATA 'NCLMAT BSS 0',BBCR
 025A  CCCD
 025B  C1D4
 025C  A0C2
 025D  D3D3
 025E  A0B0
 025F  A00D
 0260  D9CE   848          DATA 'YNPMAT BSS 0',BBCR
 0261  D0CD
 0262  C1D4
 0263  A0C2
 0264  D3D3
 0265  A0B0
 0266  A00D
 0267  CDD0   849          DATA 'MPRMAT BSS 0',BBCR
 0268  D2CD
 0269  C1D4
 026A  A0C2
 026B  D3D3
 026C  A0B0
 026D  A00D
 026E  C1D0   850          DATA 'APTR0  BSS 0',BBCR
 026F  D4D2
 0270  B0A0
 0271  A0C2
 0272  D3D3
 0273  A0B0
 0274  A00D
 0275  C1C3   851          DATA 'ACLS00 BSS 0',BBCR
 0276  CCD3
 0277  B0B0
 0278  A0C2
 0279  D3D3
 027A  A0B0
 027B  A00D
```

```
                    852          PEJ
                    853  *
                    854  *       -- ANALYZE 1ST CHARACTER IN INPUT LINE
                    855  *
 0027C              856  AFC     BSS  0
 0027C  C550        857          RMO  L,A
 0027D  B014        858          STA  AFCRET        SAVE RETURN
                    859  *
 0027E  1800        860          @LDN =IPLN
P0027F  018A
 00280  0700        861          LDA  =0
 00281  75C2        862          BRL  *GETCH@       GET CHARACTER AT CPTR
 00282  C501        863          RMO  A,E           ..INTO (E)
 00283  17FB        864          LDX  =-5           5 TYPES
 00284  0214        865  AFC05   LDA  CHAR,X
 00285  C290        866          REO  E,A
 00286  3800        867          @AND =>007F
 00287  007F
 00288  CC80        868          SNZ  A             MATCHED ?
 00289  7802        869          BRU  AFC10         YES
 0028A  40F9        870          BIX  AFC05         NO - TRY NEXT
 0028B  7805        871          BRU  AFC15
                    872  *
 0028C  0212        873  AFC10   LDA  PREDB,X
 0028D  8001        874          STA  SETP
 0028E  0106        875          LDA  SVECTR
 0028F              876  SETP    BSS  1             SET APPROP. PRED. FOR CHAR TYPE
 00290  0106        877          STA  SVECTR
 00291  7C00        878  AFC15   BRU  *AFCRET
                    879  *
 00292              880  AFCRET  BSS  1             RETURN ADDR.
                    881  *
 00293  000D        882          DATA >000D         CR
 00294  A0AF        883          DATA ' /'
 00295  FFFF        884          DATA >FFFF         NOT USED
 00296  A0AC        885          DATA ' ,'
 00297  A0A4        886          DATA ' $'
 00298  A0AA        887          DATA ' *'
 00299              888  CHAR    BSS  0
                    889  *
 00299  DB55        890          SABO 5
 0029A  DB54        891          SABO 4
 0029B  DB53        892          SABO 3
 0029C  DB52        893          SABO 2
 0029D  DB51        894          SABO 1
 0029E  DB50        895          SABO 0
 0029F              896  PREDB   BSS  0
```

```
                    897          * REJ
                    898  *
                    899  *       <- POINT LAPTR TO NEXT SEPARATOR & CLASSIFY IT
                    900  *
  029F              901  PNSC    BSS  0
  029F  C550        902          RMO  L,A
  02A0  0028        903          STA  PNSCRE
                    904  *
  02A1  0700        905          LDA  =0             FIRST CLEAR STATE VECTOR
  02A2  0106        906          STA  SVECTR
                    907  *
  02A3  1800        908          @LDM =IPLN
P 02A4  010A
  02A5  51F0        909  PNSC02  IMO  LAPTR
  02A6  01FD        910          LDA  LAPTR
  02A7  75C2        911          BRL  *GETCH@
  02A8  C501        912          RMO  A,E            GET CHAR IN (E)
  02A9  17FA        913          LDX  =-6            6 TYPES INCLUDING CR
  02AA  02EE        914  PNSC05  LDA  CHAR,X
  02AB  C200        915          REO  E,A
  02AC  3800        916          @AND =>007F
  02AD  007F
  02AE  CC00        917          SNZ  A              THIS TYPE ?
  02AF  7802        918          BRU  PNSC10         YES
  02B0  40F9        919          BIX  PNSC05         NO - TRY NEXT TYPE
  02B1  78F3        920          BRU  PNSC02
                    921  *
  02B2  01FD        922  PNSC10  LDA  LAPTR
  02B3  69FC        923          CPA  CPTR
  02B4  CDA0        924          SNE                 LDAPTR AT SAME POSITION AS CPTR ?
  02B5  7810        925          BRU  PNSC15         YES
  02B6  02E8        926          LDA  PREDB,X        NO
  02B7  8001        927          STA  SETPR
  02B8  0106        928          LDA  SVECTR
  02B9              929  SETPR   BSS  1              SET PREDICATE FOR THE SEPARATOR
  02BA  0106        930          STA  SVECTR
  02BB  1800        931          @LDM =IPLN          (M)= START OF IP BUFFER
P 02BC  010A
  02BD  01FC        932          LDA  CPTR           CURRENT POINTER
  02BE  75C2        933          BRL  *GETCH@
  02BF  6800        934          @CPA =>00AD         '-' AT CPTR ?
  02C0  00AD
  02C1  CD20        935          SEQ
  02C2  7802        936          BRU  PNS12          NO
  02C3  D873        937          SMBO 3,SVECTR       YES - SET PREDICATE FOR -
P 02C4  0008
                    938  *       IMO  CPTR           MOVE CPTR PAST '-'
  02C5  7C03        939  PNS12   BRU  *PNSCRE       RETURN
                    940  *
  02C6  D875        941  PNSC15  SMBO 5,SVECTR       SET 'LAPTR=CPTR' PREDICATE
P 02C7  0008
  02C8  7C00        942          BRU  *PNSCRE        RETURN
                    943  *
  02C9              944  PNSCRE  BSS  1
```

```
                    945          PEJ
                    946  *
                    947  *
                    948  *       -- POINT CURRENT POINTER ONE WORD AFTER
                    949  *       LOOK-AHEAD POINTER
                    95Ø  *
 Ø2CA  Ø1FD         951  PCAL    LDA  LAPTR
 Ø2CB  C3ØØ         952          RIN  A,A
 Ø2CC  Ø1FC         953          STA  CPTR
 Ø2CD  C557         954          RMO  L,P
                    955  *
                    956  *
                    957  *       -- POINT CPTR & LAPTR TO START PROCESS PRED. SPEC.
                    958  *
 Ø2CE  Ø7Ø4         959  PCL     LDA  =4
 Ø2CF  Ø1FD         96Ø          STA  LAPTR
 Ø2DØ  Ø7Ø5         961          LDA  =5
 Ø2D1  Ø1FC         962          STA  CPTR
 Ø2D2  C557         963          RMO  L,P
                    964  *
                    965  *
                    966  *       -- INCREMENT CPTR
                    967  *
 Ø2D3  51FC         968  ICPTR   IMO  CPTR
 Ø2D4  C557         969          RMO  L,P
                    97Ø  *
                    971  *       -- PROMPT FOR REPLACEMENT
                    972  *
 Ø2D5               973  PFR     BSS  Ø
 Ø2D5  C55Ø         974          RMO  L,A          SAVE RETURN
 Ø2D6  ØØØ5         975          STA  PFRRET
                    976  *
 Ø2D7  1FØ6         977          LDM  =6           'INVALID CHARACTER...'
 Ø2D8  75C1         978          BRL  *VTC@
 Ø2D9  1FØ5         979          LDM  =5           '>'
 Ø2DA  75C1         98Ø          BRL  *VTC@
 Ø2DB  7CØØ         981          BRU  *PFRRET
                    982  *
 Ø2DC               983  PFRRET  BSS  1            RETURN ADDR.
```

```
                    984          PEJ
                    985  *
                    986  *
                    987  *    -- WRITE A COMMENT LINE TO ACTION MATRIX
                    988  *
  02DD              989  WCL    BSS  0
  02DD  01D1        990          LDA  ACOFFS       POINTER TO CURRENT ENTRY IN ACTION MAT.
  02DE  0009        991          STA  PRBWCL+4     ..AS RECORD #
  02DF  1800        992          @LDM --PRBWCL
P 02E0  02E4
  02E1  CAE0        993          SVC  0
  02E2  51D1        994          IMO  ACOFFS
  02E3  C557        995          RMO  L,P
                    996  *
                    997  *
  02E4  0003        998  PRBWCL DATA LU3          LUN
  02E5  0002        999          DATA 2            WRITE
  02E6  0000       1000          DATA 0
P 02E7  018A       1001          DATA IPLN         BUFFER ADDR
  02E8  0000       1002          DATA 0            RECORD #
  02E9  0000       1003          DATA 0            WORD #
  02EA  0023       1004          DATA LLNTH        WORD COUNT
```

```
                    1805          PEJ
                    1806  *
                    1807  *       -- CONVERT DEC ASCII STRING INTO BINARY VALUE
                    1808  *          START AND END OF STRING ARE DETERMINED BY
                    1809  *          (CPTR) AND (LAPTR)-1
                    1810  *
                    1811  *       RETURN : BINARY VALUE IN BV
                    1812  *
  02EB              1813  D2BCON BSS  0
  02EB  C550        1814         RMO  L,A          SAVE RETURN
  02EC  8000        1815         STA  CONRET
                    1816  *
  02ED  01FB        1817         LDA  LAPTR        GET LOOK-AHEAD PTR
  02EE  29FC        1818         SUB  CPTR
  02EF  C502        1819         RMO  A,X          (X)= # OF ASCIIS
  02F0  0000        1820         @LDA =IPLN        START ADDR OF I/P LINE BUFFER
P 02F1  01DA
  02F2  C0C1        1821         LLA  1            CONVERT TO BYTE POINTER
  02F3  21FC        1822         ADD  CPTR         (A)= BYTE POINTER TO 1ST ASCII
  02F4  75C3        1823         BRL  *DE2BI@      DEC-TO-BIN CONVERSION
  02F5  8400        1824         @STA BV           PLACE RESULT IN BV
P 02F6  01D5
  02F7  7C00        1825         BRU  *CONRET      RETURN
                    1826  *
  02F8              1827  CONRET BSS  1            RETURN ADDR
```

```
                    1#28          PEJ
                    1#29  *
                    1#3#  *       - SET PREDICATE BIT IN YNPBUF
                    1#31  *         BIT NO. IS DETERMINED BY BV
                    1#32  *
                    1#33  *           INPUT  : BV
                    1#34  *           OUTPUT : YNPBUF
                    1#35  *
  #2F9              1#36  SBYNP   BSS  #
  #2F9  LA##        1#37          @LDX BV           BIT NO. TO SET
P #2FA  #1#5
  #2FB  #6##        1#38          @LDA SABO#,X
P #2FC  #3#2
  #2FD  8##1        1#39          STA  SETBIT
  #2FE  #1D3        1#4#          LDA  YNPBUF       (A)=YN BUFFER TO TO SET
  #2FF              1#41  SETBIT  BSS  1            'SABO #' TO SET BIT IN (A)
  #3##  81D3        1#42          STA  YNPBUF
  #3#1  C557        1#43          RHO  L,P          RETURN
                    1#44  *
  #3#2  DB5#        1#45  SABO#   SABO #
  #3#3  DB51        1#46          SABO 1
  #3#4  DB52        1#47          SABO 2
  #3#5  DB53        1#48          SABO 3
  #3#6  DB54        1#49          SABO 4
  #3#7  DB55        1#5#          SABO 5
  #3#8  DB56        1#51          SABO 6
  #3#9  DB57        1#52          SABO 7
  #3#A  DB58        1#53          SABO 8
  #3#B  DB59        1#54          SABO 9
  #3#C  DB5A        1#55          SABO 1#
  #3#D  DB5B        1#56          SABO 11
  #3#E  DB5C        1#57          SABO 12
  #3#F  DB5D        1#58          SABO 13
  #31#  DB5E        1#59          SABO 14
  #311  DB5F        1#6#          SABO 15
                    1#61  *
                    1#62  *
                    1#63  *       -- SAVE CLUSTER NUMBER
                    1#64  *
  #312  #4##        1#65  SACLS   @LDA BV
P #313  #1#5
  #314  81CB        1#66          STA  CLUS#
  #315  C557        1#67          RHO  L,P
```

```
9900A OFF LINE ASSEMBLER                                        SHEET   32
                    1#68          PEJ
                    1#69    *
                    1#7#    *     - SET BIT IN MPRBUF
                    1#71    *       BIT NO. IS DETERMINED BY BV
                    1#72    *
                    1#73    *         INPUT  : BV
                    1#74    *         OUTPUT : MPRBUF
                    1#75    *
  #316              1#76  SBMPR   BSS  #
  #316  14##        1#77          @LDX BV             BIT TO BE SET
P #317  #1#5
  #318  #6##        1#78          @LDA SAB0#,X        INST. TO SET BIT IN (A)
P #319  #3#2
  #31A  8##1        1#79          STA  SBIT
  #31B  #1D4        1#8#          LDA  MPRBUF         (A)= MPR BUFFER
  #31C              1#81  SBIT    BSS  1              INST. TO SET BIT
  #31D  81D4        1#82          STA  MPRBUF
  #31E  C557        1#83          RMO  L,P            RETURN
                    1#84    *
                    1#85    *
                    1#86    *     -- WRITE SUBROUTINE NAME INTO ACTION LIST
                    1#87    *
  #31F              1#88  PAILN   BSS  #
  #31F  #4##        1#89          @LDA BV             ACTION NUMBET
P #32#  #1#5
  #321  69C8        1#9#          CPA  NOACTS
  #322  CD8#        1#91          SGE                 VALID ACTION NUMBER ?
  #323  C557        1#92          RMO  L,P            NO
                    1#93    *
  #324  8##E        1#94          STA  PRBRAC+4       YES - USE IT AS RECORD NUMBER
  #325  18##        1#95          @LDM =PRBRAC
P #326  #32F
  #327  CAE#        1#96          SVC  #
                    1#97    *
  #328  #1D1        1#98          LDA  ACOFFS         OFFSET TO CURRENT LINE IN ACTION LIST
  #329  8#1#        1#99          STA  PRBWAC+4       ..AS RECORD NUMBER
  #32A  18##        11##          @LDM =PRBWAC
P #32B  #336
  #32C  CAE#        11#1          SVC  #
                    11#2    *
  #32D  51D1        11#3          IMO  ACOFFS         BUMP TO NEXT LINE
  #32E  C557        11#4          RMO  L,P
                    11#5    *
                    11#6    *
  #32F  ###1        11#7  PRBRAC  DATA LU1
  #33#  ####        11#8          DATA #              READ
  #331  ####        11#9          DATA #
P #332  #34#        111#          DATA ABUF           BUFFER ADDR
  #333  ####        1111          DATA #              RECORD
  #334  ####        1112          DATA #              WORD #
  #335  ##23        1113          DATA LLNTH          WD COUNT
  #336  ###3        1114  PRBWAC  DATA LU3
  #337  ###2        1115          DATA 2              WRITE
  #338  ####        1116          DATA #
P #339  #33D        1117          DATA WABUF
  #33A  ####        1118          DATA #
  #33B  ####        1119          DATA #
```

```
 #33C  0000  1120          DATA 128
             1121  *
 #33D  A0C4  1122  VABUF   DATA ' DATA '
 #33E  C1D4
 #33F  C1A0
 #340        1123  ABUF    BSS  128-3        BUFFER
```

```
                    1124          PEJ
                    1125  *
                    1126  *
                    1127  *       -- CONVERT BINARY CLUSTER BUFFER INTO CLUSTER MATRIX
                    1128  *
  #38D              1129  WCM     BSS  #
  #38D  C55#        113#          RMO  L,A          SAVE RETURN
  #38E  8#16        1131          STA  WCMRET
                    1132  *
  #38F  #1C6        1133          LDA  NOCLS        # OF CLUSTERS
  #3C#  C#8#        1134          RAD  A,A          2 WORDS / CLUSTER
  #3C1  8#14        1135          STA  NOWD
                    1136  *
  #3C2  #7##        1137          LDA  =#           RESET INDEX
  #3C3  8#13        1138          STA  INDX
  #3C4  1#12        1139  WCM5    LDX  INDX
  #3C5  #3E6        114#          LDA  CLSBUF,X     BINARY VALUE
  #3C6  #F##        1141          LDE  =#           NO ZERO SUPPRESSION
  #3C7  75C4        1142          BRL  *BI2HA@      CONVERT BIN TO HEX
  #3C8  A4##        1143          @DST OPBUF+4      'HHHH'
P #3C9  #55A
  #3CA  #1CC        1144          LDA  COFFS        OFFSET CURRENT ENTRY IN CLSMAT
  #3CB  84##        1145          @STA PRBWYN+4     ..AS RECORD #
P #3CC  #48E
  #3CD  18##        1146          @LDM =PRBWYN
P #3CE  #48A
  #3CF  CAE#        1147          SVC  #
  #3D#  51CC        1148          IMO  COFFS        BUMP CLUSTER ENTRY OFFSET
  #3D1  5##5        1149          IMO  INDX         BUMP INDEX
  #3D2  48#3        115#          DMT  NOWD         FINISHED ?
  #3D3  78F#        1151          BRU  WCM5         NO
                    1152  *
  #3D4  7C##        1153          BRU  *WCMRET      RETURN
                    1154  *
                    1155  *
  #3D5              1156  WCMRET BSS  1            RETURN
  #3D6              1157  NOWD   BSS  1            # OF WORDS IN CLUSTER MATRIX
  #3D7              1158  INDX   BSS  1            INDEX
```

```
                    1159          PEJ
                    116#  *
                    1161  *
                    1162  *       -- PLACE TERMINATOR 'DATA #' INTO ACTION LIST
                    1163  *
   #3D8   #1D1      1164  PT      LDA  ACOFFS       CURRENT LINE
   #3D9   8#18      1165          STA  PRBT+4
   #3DA   18##      1166          @LDM =PRBT
P  #3DB   #3EE
   #3DC   CAE#      1167          SVC  #
                    1168  *
   #3DD   51D1      1169          IMO  ACOFFS       BUMP TO NEXT LINE
   #3DE   #1D1      117#          LDA  ACOFFS       USE AS REC # IN PRB
   #3DF   8#1E      1171          STA  PRBAST+4
   #3E#   18##      1172          @LDM =PRBAST
P  #3E1   #3FA
   #3E2   CAE#      1173          SVC  #            PLACE '*' AFTER 'DATA #'
                    1174  *
   #3E3   51D1      1175          IMO  ACOFFS       BUMP TO NEXT LINE
   #3E4   ##21      1176          LDA  PRBS+3
   #3E5   27#7      1177          ADD  =7           POINT TO NEXT NEXT LABEL
   #3E6   8#1F      1178          STA  PRBS+3
   #3E7   #1D1      1179          LDA  ACOFFS       CURRENT LINE
   #3E8   8#1E      118#          STA  PRBS+4
   #3E9   18##      1181          @LDM =PRBS
P  #3EA   #4#3
   #3EB   CAE#      1182          SVC  #            WRITE 'ACLS..'
   #3EC   51D1      1183          IMO  ACOFFS
   #3ED   C557      1184          RMO  L,P          RETURN
                    1185  *
   #3EE   ###3      1186  PRBT    DATA LU3
   #3EF   ###2      1187          DATA 2            WRITE
   #3F#   ####      1188          DATA #
   #3F1   #3F5      1189          DATA TERMTR       BUFFER
   #3F2   ####      119#          DATA #            RECORD#
   #3F3   ####      1191          DATA #            WORD#
   #3F4   ###5      1192          DATA 5            WD COUNT
   #3F5   A#C4      1193  TERMTR  DATA ' DATA # ',BBCR
   #3F6   C1D4
   #3F7   C1A#
   #3F8   B#A#
   #3F9   A##D
                    1194  *
   #3FA   ###3      1195  PRBAST DATA LU3
   #3FB   ###2      1196          DATA 2            WRITE_
   #3FC   ####      1197          DATA #
P  #3FD   #4#1      1198          DATA ASTER        BUFFER ADDR
   #3FE   ####      1199          DATA #            REC
   #3FF   ####      12##          DATA #            WORD #
   #4##   ###2      12#1          DATA 2            WORD COUNT
   #4#1   AAA#      12#2  ASTER   DATA '* ',BBCR
   #4#2   A##D
   #4#3   ###3      12#3  PRBS    DATA LU3
   #4#4   ###2      12#4          DATA 2
   #4#5   ####      12#5          DATA #
          ###9      12#6          DATA ACLBSS
E  #4#6   ####
```

```
#487  0000  1287        DATA 0
#488  0000  1288        DATA 0
#489  0007  1289        DATA 7
```

```
                    1210          PEJ
                    1211  *
                    1212  *       -- SAVE POINTERS
                    1213  *
                    1214  *
  0400              1215  SAVEP   BSS  0
  0400  17FA        1216          LDX  =NOPTRS      -VE NO. OF POINTERS TO BE SAVED
  0401  0600        1217  SAVEP0  @LDA OFFS,X       GET VALUE OF POINTER
P 0402  00D7
  0403  820E        1218          STA  OPTRS,X      ..SAVE IT
  0404  40FC        1219          BIX  SAVEP0
  0405  C557        1220          RMO  L,P
                    1221  *
                    1222  *
                    1223  *       --   RESTORE POINTERS
                    1224  *
  0406              1225  RESTOR  BSS  0
  0406  17FA        1226          LDX  =NOPTRS      -VE NO. OF POINTERS TO SAVE
  0407  020A        1227  RES00   LDA  OPTRS,X      GET OLD VALUE OF POINTER
  0408  0600        1228          @STA OFFS,X       RESTORE
P 0409  00D7
  040A  40FC        1229          BIX  RES00        SAVE RESTORE NEXT POINTER
  040B  C557        1230          RMO  L,P
                    1231  *
  040C              1232  OPTRSB  BSS  1           FOR COFFS
  040D              1233          BSS  1               NOFFS
  040E              1234          BSS  1               YOFFS
  040F              1235          BSS  1               MOFFS
  0410              1236          BSS  1               AOFFS
  0411              1237          BSS  1               ACOFFS
  0412              1238  OPTRS   BSS  0           OLD VALUES OF POINTERS
        FFFA        1239  NOPTRS  EQU  OPTRSB-OPTRS -VE NO. OF POINTERS
                    1240  *
                    1241  *
                    1242  *       -- WRITE TO NEXT CLUSTER MATRIX
                    1243  *
  0412              1244  WNCN    BSS  0
  0412  C55A        1245          RMO  L,A
  0413  8028        1246          STA  WNCRET       SAVE RETURN
                    1247  *
  0414  1800        1248          @LDM =IPLN        (M)= START OF I/P BUFFER
P 0415  0100
  0416  01FC        1249          LDA  CPTR         (A)= BYTE OFFSET
  0417  75C2        1250          BRL  *GETCH0      GET 1ST CHAR
  0418  C8C0        1251          LLA  8            SHIFT TO LH BYTE
  0419  8021        1252          STA  SAVE
  041A  01FD        1253          LDA  LAPTR
  041B  29FC        1254          SUB  CPTR         # OF CHARS
  041C  6F01        1255          CPA  =1
  041D  CDA0        1256          SNE
  041E  7805        1257          BRU  WNC5         1 CHAR
  041F  01FC        1258          LDA  CPTR
  0420  C300        1259          RIN  A,A          2ND CHAR
  0421  75C2        1260          BRL  *GETCH0
  0422  2018        1261          ADD  SAVE         PLACE 2 CHARS TOGETHER
  0423  7803        1262          BRU  WNC7
                    1263  *
```

```
 #42E  0016  1264  WNC6   LDA  SAVE
 #42F  2000  1265        @ADD ->00A0        PLACE BLANK
 #430  00A0
 #431  0011  1266  WNC7   STA  OPBUF2+3
 #432  01CD  1267        LDA  NOFFS         CURRENT NCL ENTRY
 #433  0009  1268        STA  PRBWNC+4      ..AS RECORD#
 #434  1800  1269        @LDM -PRBWNC
P #435  0439
 #436  CAE0  1270        SVC  0
 #437  51CD  1271        IMO  NOFFS         POINT TO NEXT ENTRY IN NCLMAT
 #438  7C0D  1272        BRU  *WNCRET       RETURN
             1273  *
 #439  0003  1274  PRBWNC DATA LU3
 #43A  0002  1275        DATA 2             WRITE
 #43B  0000  1276        DATA 0
P #43C  0440  1277        DATA OPBUF2
 #43D  0000  1278        DATA 0             REC#
 #43E  0000  1279        DATA 0             WORD#
 #43F  0005  1280        DATA 5             WORD COUNT
             1281  *
 #440  A0C4  1282  OPBUF2 DATA ' DATA NN',BBCR
 #441  C1D4
 #442  C1A0
 #443  CECE
 #444  A00D
 #445  0001  1283  SAVE   DATA 1             TEMP SAVE
 #446        1284  WNCRET BSS  1             RETURN
```

```
                    1285          PEJ
                    1286  *
                    1287  *
                    1288  *       -- SET THE ALTERNATIVE BIT IN THE CLUSTER BUFFER
                    1289  *
0447                1290  SABIC@  BSS   @
0447  C55@          1291          RMO   L,A
0448  8@2@          1292          STA   SABRET
0449  @1CB          1293          LDA   CLUS@
044A  C7@@          1294          RDE   A,A
044B  C8@1          1295          LLA   1
044C  8@44          1296          STA   CINDX       SAVE CLS DOUBLE INDEX
                    1297  *
044D  C5@2          1298          RMO   A,X
044E  83E6          1299          DLD   CLSBUF,X    GET BIN CLS BUFFER
044F  A@42          13@@          DST   WD1         & SAVE
0450  @1CA          13@1          LDA   ALT@        CURRENT ALTERNATIVE @
0451  6F1@          13@2          CPA   =16
0452  CDC@          13@3          SLE
0453  78@9          13@4          BRU   WC5         ALTS @-15 : WORD 1
0454  2F1@          13@5          SUB   =16         ALTS 16-31 : WORD 2
0455  C5@2          13@6          RMO   A,X
0456  @6@@          13@7          @LDA  SABO@,X
P 0457  @3@2
0458  8@@1          13@8          STA   LOC3
0459  @@39          13@9          LDA   WD2         WORD 1 OF CLS
045A                1310  LOC3    BSS   1           'SABO @'
045B  8@37          1311          STA   WD2
                    1312  **      @BRL  RLOC
045C  78@7          1313          BRU   WC7
                    1314  *
045D  C5@2          1315  WC5     RMO   A,X
045E  @6@@          1316          @LDA  SABO@,X
P 045F  @3@2
0460  8@@@          1317          STA   LOC4
0461  @@3@          1318          LDA   WD1         SET BIT IN WORD 1 OF CLUSTER
0462                1319  LOC4    BSS   1           'SABO @'
0463  8@2E          132@          STA   WD1
                    1321  **      @BRL  RLOC
0464  1@2C          1322  WC7     LDX   CINDX
0465  B@2C          1323          DLD   WD1
0466  A3E6          1324          DST   CLSBUF,X
0467  51CA          1325          IMO   ALT@        NEXT ALT TO BE PROCESSED
0468  7C@@          1326          BRU   *SABRET
                    1327  *
0469                1328  SABRET  BSS   1           RETURN
```

```
                    1329          PEJ
                    1330  *
                    1331  *
                    1332  *       -- WRITE TO Y/N, MASK MATRIX:
                    1333  *
  0046A             1334  WYM     BSS  0
  0046A  C550       1335          RMO  L,A
  0046B  8028       1336          STA  WCYRET
                    1337  *
                    1338  *
                    1339  *       WRITE INTO YNMMAT
                    1340  *
  0046C  01D3       1341          LDA  YNPBUF         Y/N IN BINARY FORM
  0046D  0F00       1342          LDE  =0             NO ZERO SUPPRESION
  0046E  75C4       1343          BRL  *BI2HA@        CONVERT TO HEX
  0046F  A400       1344          @DST OPBUF+4        '....'
P 00470  0055A
  00471  01CE       1345          LDA  YOFFS          CURRENT YNM ENTRY
  00472  8400       1346          @STA PRBWYN+4       ..AS RECORD #
P 00473  048E
  00474  1800       1347          @LDM =PRBWYN
P 00475  048A
  00476  CAE0       1348          SVC  0
  00477  D8E0       1349          SRF
P 00478  04D0       1350          DATA REGFIL
                    1351  *       @BRL WSTR
  00479  D8A0       1352          LRF
P 0047A  04D0       1353          DATA REGFIL
  0047B  5400       1354          @IMO YOFFS          BUMP YNM ENTRY
P 0047C  00D3
                    1355  *
                    1356  *       WRITE INTO MPR MATRIX
                    1357  *
  0047D  01D4       1358          LDA  MPRBUF         MASK IN BINARY FORM
  0047E  0F00       1359          LDE  =0             NO ZERO SUPPRESSION
  0047F  75C4       1360          BRL  *BI2HA@        CONVERT TO HEX
  00480  A400       1361          @DST OPBUF+4        '....'
P 00481  0055A
  00482  01CF       1362          LDA  MOFFS          MASK ENTRY
  00483  8400       1363          @STA PRBWYN+4       ..AS RECORD #
P 00484  048E
  00485  1800       1364          @LDM =PRBWYN
P 00486  048A
  00487  CAE0       1365          SVC  0
  00488  51CF       1366          IMO  MOFFS
                    1367  *
  00489  7C0A       1368          BRU  *WCYRET        RETURN
                    1369  *
                    1370  *
  0048A  0003       1371  PRBWYN  DATA LU3
  0048B  0002       1372          DATA 2              WRITE
  0048C  0000       1373          DATA 0
P 0048D  0556       1374          DATA OPBUF
  0048E  0000       1375          DATA 0              RECORD #
  0048F  0000       1376          DATA 0              WORD #
  00490  0007       1377          DATA 7              WORD COUNT
                    1378  *
```

```
990A OFF LINE ASSEMBLER                                          SHEET   41
  #491  ####  1379  CINDX   DATA #          CLS DOUBLE INDEX
  #492        138#  WD1     BSS  1          CLS WORD 1
  #493        1381  WD2     BSS  1                   2
  #494        1382  WCVRET  BSS  1          RETURN
              1383  *
              1384  *
```

```
                    1385          PEJ
                    1386  *
                    1387  *
                    1388  *       -- CREATE PERMANENT SOURCE FILE BY COPYING
                    1389  *          LUN 3 TO LUN 4
  0495              1390  CPF     BSS  0
  0495  C550       1391          RMO  L,A          SAVE RETURN
  0496  0038       1392          STA  CPFRET
                    1393  *
  0497  01D1       1394          LDA  ACOFFS       LAST RECORD USED IN LUN 3
  0498  C700       1395          RDE  A,A          DO NOT COUNT 'ACLS..' FOR NEXT ALT
  0499  C102       1396          RCO  A,X          ..USED AS LOOP INDEX
                    1397  *
  049A  1800       1398  CPF3    @LDM =PRBL3       READ LUN 3 RECORD
P 049B  04DC
  049C  CAE0       1399          SVC  0
  049D  5042       1400          IMO  PRBL3+4      BUMP TO NEXT RECORD
  049E  704E       1401          BRL  CC           COUNT # OF WORD
  049F  8049       1402          STA  WDCNT        PLACE IN PRB WORD COUNT
  04A0  C520       1403          RMO  X,A
  04A1  6FFF       1404          CPA  =-1          LAST REC
  04A2  CD20       1405          SEQ
  04A3  700D       1406          BRU  WR           NO
  04A4  0000       1407          @LDA =OPBUF       YES
P 04A5  0556
  04A6  2042       1408          ADD  WDCNT        FORM WD CNT TO EOF
  04A7  8033       1409          STA  TPPTR
  04A8  802F       1410          DLD  ENDS
  04A9  A431       1411          DST  *TPPTR       ...TO BUFFER
  04AA  5030       1412          IMO  TPPTR
  04AB  502F       1413          IMO  TPPTR
  04AC  002D       1414          LDA  ENDS+2
  04AD  842D       1415          STA  *TPPTR
  04AE  003A       1416          LDA  WDCNT        4 MORE WORD
  04AF  2704       1417          ADD  =4
  04B0  8038       1418          STA  WDCNT
  04B1              1419  WR      BSS  0
  04B1  1800       1420  LOC1    @LDM =PRBL4       WRITE INTO LUN 4
P 04B2  04E3
  04B3  CAE0       1421          SVC  0
                    1422  *       SRF
                    1423  *       DATA REGFIL
                    1424  *       BRL  WSTR
                    1425  *       LRF
                    1426  *       DATA REGFIL
                    1427  *
  04B4  0034       1428          LDA  WDCNT
  04B5  2032       1429          ADD  WD0          FIND OUT WHERE TO START NEXT TIME
  04B6  6800       1430          @CPA =128         PAST TO NEXT 'S' RECORD ?
  04B7  0080
  04B8  CD40       1431          SGT
  04B9  7802       1432          BRU  CPF5         YES
  04BA  802D       1433          STA  WD0          NO - UPDATE WORD OFFSET
  04BB  7804       1434          BRU  CPF7
                    1435  *
  04BC  2800       1436  CPF5    @SUB =128         CALCULATE WORD OFF SET IN NEXT RECORD
  04BD  0080
```

```
  04BE  0029  1437          STA  WD0
  04BF  5027  1438          IMO  REC0          BUMP RECORD # ON LUN 4
  04C0  40D9  1439  CPF7    BIX  CPF3          TRANSFER NEXT RECORD
  04C1  1800  1440          @LDM =PRBCLO       MAKE FILE PERMANENT
P 04C2  04EA
  04C3  CAE8  1441          SVC  8
  04C4  B400  1442          @DLD PRBFA+5
P 04C5  0189
  04C6  A400  1443          @DST M12+15
              0008
E 04C7  000F
  04C8  0400  1444          @LDA PRBFA+7
P 04C9  0188
  04CA  8400  1445          @STA M12+17
              0008
E 04CB  0011
  04CC  1F0C  1446          LDM  =12
  04CD  75C1  1447          BRL  *WTC0
  04CE  7C00  1448          BRU  *CPFRET       RETURN
              1449  *
              1450  *
  04CF        1451  CPFRET  BSS  1             RETURN
  04D0        1452  REGFIL  BSS  8
  04D8  A0C5  1453  ENDS    DATA >A0C5,>CEC4
  04D9  CEC4
  04DA  0D0D  1454  EOF     DATA >0D0D
  04DB        1455  TPPTR   BSS  1             TEMP PTR
              1456  *
  04DC  0003  1457  PRBL3   DATA LU3           LUN 3
  04DD  0000  1458          DATA 0             READ
  04DE  0000  1459          DATA 0
P 04DF  0556  1460          DATA OPBUF
  04E0  0000  1461          DATA 0
  04E1  0000  1462          DATA 0
  04E2  0080  1463          DATA 128
              1464  *
  04E3  0004  1465  PRBL4   DATA LU4           LUN 4
  04E4  0002  1466          DATA 2             WRITE
  04E5  0000  1467          DATA 0
P 04E6  0556  1468          DATA OPBUF
  04E7  0000  1469  REC0    DATA 0
  04E8  0000  1470  WD0     DATA 0
  04E9  0000  1471  WDCNT   DATA 0
              1472  *
              1473  *       PRB TO CLOSE AND TO MAKE FILE PERMANENT
  04EA  0004  1474  PRBCLO  DATA LU4
  04EB  0004  1475          DATA 4             CLOSE
  04EC  0000  1476          DATA 0             N/A
```

```
                    1477          PEJ
                    1478  *
                    1479  *       SUBROUTINE TO COUNT NUMBER OF WORDS IN LINE
                    1480  *       A LINE IS TERMINATED ( AND INCLUDING) BY CR
                    1481  *       IF CR IS LOCATED IN LH BYTE, A BLANK IS INSERTED BEFORE IT
                    1482  *
                    1483  *       ENTRY  : (M)= START ADDR OF BUFFER
                    1484  *       RETURN : (A)= # OF WORDS (INCLUDING CR)
                    1485  *
  04ED              1486  CC      BSS  0
  04ED  C550        1487          RMO  L,A
  04EE  801C        1488          STA  CCRET
                    1489  *
  04EF  1800        1490  LOC2    @LDM =OPBUF       (M)=START OF O/P BUFFER
P 04F0  0556
  04F1  07FF        1491          LDA  =-1          RESET CHAR COUNT
  04F2  8019        1492          STA  CCNT
                    1493  *
  04F3  5018        1494  CC5     IMO  CCNT
  04F4  0017        1495          LDA  CCNT
  04F5  75C2        1496          BRL  *GETCH@      GET CHAR
  04F6  3800        1497          @AND =>007F
  04F7  007F
  04F8  6800        1498          @CPA =>000D       CR ?
  04F9  000D
  04FA  CD20        1499          SEQ
  04FB  78F7        1500          BRU  CC5          NO - CHECK NEXT CHAR
                    1501  *                         YES
  04FC  DB2F        1502  CC7     TMBZ 15,CCNT      IS CR IS LH BYTE ?
P 04FD  050C
  04FE  7808        1503          BRU  CC10         NO
  04FF  000C        1504          LDA  CCNT         YES - THEN PLACE BLANK BEFORE CR
  0500  8000        1505          @LDE =>A00D
  0501  A00D
  0502  C841        1506          LRA  1            WD OFFSET FROM START OF BUFFER
  0503  C0B0        1507          RAD  M,A          FORM POINTER
  0504  8008        1508          STA  TPTR
  0505  8C07        1509          STE  *TPTR        PLACE BLANK + CR
  0506  5005        1510          IMO  CCNT         TO COUNT THE BLANK
  0507  0004        1511  CC10    LDA  CCNT         CHAR COUNT
  0508  2701        1512          ADD  =1           ADJUST
  0509  C841        1513          LRA  1            WORD COUNT
  050A  7C00        1514          BRU  *CCRET       ..ON RETURN
                    1515  *
  050B              1516  CCRET   BSS  1            RETURN
  050C              1517  CCNT    BSS  1            CHAR COUNT
                    1518  *
  050D              1519  TPTR    BSS  1            TEMP POINTER
```

```
                    1520          PEJ
                    1521  *
                    1522  *       FOR DEBUGGING : WRITE CONTENT OF A LOCATION
                    1523  *
  0050              1524  RLOC    BSS  0
  0050  D80E        1525          SRF
P 0051  004D        1526          DATA REGFIL
  0053  C556        1527          RMO  L,A
  0054  8010        1528          STA  RLORET
  0055  17F6        1529          LDX  =LTAB-LTABE
                    1530  *
  0057  021A        1531  DEB5    LDA  LTABE,X
  0059  9023        1532          STX  INDEX
  005B  800D        1533          STA  RLOPTR
  005D  040C        1534          LDA  *RLOPTR        GET CONTENT
  005F  0F00        1535          LDE  =0             NO ZERO SUPPRES'N
  0061  75C4        1536          BRL  *BI2HA0
  0063  A018        1537          DST  DEBUF+3
  0065  1800        1538          @LDM =PRBDE
P 0067  0052
  0069  CAE0        1539          SVC  0
  006B  101A        1540          LDX  INDEX
  006D  40F4        1541          BIX  DEB5
  006F  D8A0        1542          LRF
P 0071  004D        1543          DATA REGFIL
  0073  7C00        1544          BRU  *RLORET
                    1545  *
  0075              1546  RLORET  BSS  1
  0076              1547  RLOPTR  BSS  1
                    1548  *
P 0077  00D0        1549  LTAB    DATA CLUS0
P 0079  00CF        1550          DATA ALT0
P 007B  0092        1551          DATA WD1
P 007D  0093        1552          DATA WD2
P 007F  005A        1553          DATA LOC3
P 0081  0062        1554          DATA LOC4
P 0083  004D        1555          DATA REGFIL
P 0085  04D2        1556          DATA REGFIL+2
P 0087  00E8        1557          DATA CLSBUF
P 0089  00EC        1558          DATA CLSBUF+1
  008B              1559  LTABE   BSS  0
                    1560  *
  008B  0000        1561  PRBDE   DATA 0
  008D  0002        1562          DATA 2
  008F  000C        1563          DATA 12             CHAR CONUT
P 0091  0092        1564          DATA $+1
  0093  A0A0        1565  DEBUF   DATA '      >HHHH',LFCR
  0095  A0A0
  0097  A0BE
  0099  C8C8
  009B  C8C8
  009D  8A8D
  009F              1566  INDEX   BSS  1
                    1567  *
                    1568  *
                    1569  *       FOR DEBUGGING : WRITE STRING OF ASCIIS TO CONSOLE
                    1570  *
```

```
9800A OFF LINE ASSEMBLER                                        SHEET   46
 0539          1571  WSTR    BSS  0
 0539  1700    1572          LDX  =0          RESET INDEX
 053A  0212    1573  WS05    LDA  STRG,X      GET STARTING ADDRESS
 053B  CC00    1574          SNZ  A
 053C  C552    1575          RMO  L,P         FINISHED
 053D  0016    1576          STA  PRBAS+3     BUFFER ADDR
 053E  0210    1577          LDA  STRLTH,X    # OF CHARS
 053F  2702    1578          ADD  =2          FOR LFCR
 0540  0012    1579          STA  PRBAS+2
 0541  2F02    1580          SUB  =2
 0542  C041    1581          LRA  1           WD OFFSET
 0543  2209    1582          ADD  STRG,X
 0544  0018    1583          STA  TEMP
 0545  0000    1584          @LDA =LFCR
 0546  0A0D
 0547  040D    1585          STA  *TEMP
 0548  1800    1586          @LDM =PRBAS
P0549  0551
 054A  CAE0    1587          SVC  0
 054B  C322    1588          RIN  X,X         NEXT STRING
 054C  70ED    1589          BRU  WS05
               1590  *
P054D  0556    1591  STRG    DATA OPBUF
 054E  0000    1592          DATA 0           TERMONATOR
               1593  *
 054F  0010    1594  STRLTH  DATA 24
 0550  0000    1595          DATA 0
               1596  *
 0551  0000    1597  PRBAS   DATA 0
 0552  0002    1598          DATA 2
 0553  0000    1599          DATA 0           CHAR CNT
 0554  0000    1600          DATA 0           BUFFER ADDRESS
               1601  *
 0555          1602  TEMP    BSS  1
               1603  *
               1604  *
 0556  A0C4    1605  OPBUF   DATA ' DATA >HHHH',BBCR
 0557  C1D4
 0558  C1A0
 0559  A0BE
 055A  C8C8
 055B  C8C8
 055C  A00D
 055D          1606          BSS  128-7
               1607  *
               1608          END
```

```
   A       ####     ABLENT  ###4     ABUF    #34#     ACLBSS  ###9     ACLS#0  ###7
   ACLST#  ##77     ACLST1  ##7A     ACLST2  ##7D     ACLST3  ##7F     ACLST4  ##82
   ACLST5  ##88     ACLST6  ##8A     ACLST7  ##8B     ACLST8  ##8C     ACLST9  ##92
 R ACLST0  ###A     ACLT1#  ##95     ACLT11  ##98     ACLT12  ##A1     ACLT13  ##A7
   ACLT14  ##AD     ACLT15  ##B#     ACLT16  ##B5     ACLT17  ##BB     ACLT18  ##BE
   ACLT19  ##C2     ACOFFS  ##D6     ACTNS   ##DB   R ACTNS0  ##DA   R ACTNSE  ##EB
   AFC     #27C     AFC#5   #284     AFC1#   #28C     AFC15   #291     AFCRET  #292
   ALT0    ##CF     AOFFS   ##D5     AOPR8   #178     APTR#   ##63   R APTR#0  ##1#
   ASTER   #4#1   R B       ###6     BBBB    #1A6     BBCR    A##D     BI2DA0  ##CA
   BI2DAS  ###A     BI2HR0  ##C9     BI2HAS  ###5   R BR      ###1   R BREG    ##11
   BV      #1#5     CC      #4ED     CC1#    #5#7     CC5     #4F3   R CC7     #4FC
   CCNT    #5#C     CCRET   #5#B     CHAR    #299     CINDX   #491     CLRSV   #1F4
   CLSBUF  ##EB   R CLSMA0  ###C     CLSMAT  ##13     CLUS0   ##D#     COFFS   ##D1
   CONRET  #2F8     CPF     #495     CPF3    #49A     CPF5    #4BC     CPF7    #4C#
   CPFRET  #4CF     CPTR    #1#1     D2BCON  #2EB     DE2BI0  ##C8     DEB5    #513
   DEBUF   #532     DEC2BI  ###3     E       ###1     EMPTY   4###     ENDS    #4D8
 R EOF     #4DA     FN2     #159     FN5     #178     FNAME   #157     FNARET  #18C
   GETCH0  ##C7     GETCHA  ###2     HDER0   ###6     HLABL1  ###C     HSIZE   ###F
   ICPTR   #2D3     INDEX   #538     INDX    #3D7     INIT    #133     INITRE  #144
   IPLN    #1#A     IPLNX   #1#6     IPLNE   #132     IPLNS   #1#A     L       ###5
 R L1##    #14#   R L1#1    #143     LABEL   #252     LAPTR   #1#2     LFCR    8A8D
   LINDX   #251     LLNTH   ##23   R LOC1    #481   R LOC2    #4EF     LOC3    #45A
   LOC4    #462   R LREG    ##12     LTAB    #524     LTABE   #52E     LU1     ###1
   LU3     ###3     LU4     ###4     M       ###3     M12     ###8     M13     ###B
   MOFFS   ##D4     MPRBUF  ##D9   R MPRMA0  ###F     MPRMAT  ##4F   R NCLBUF  ##D7
 R NCLMA0  ###D     NCLMAT  ##27     NOACTS  ##CD     NOALTS  ##CC     NOCHRS  ##CE
   NOCLS   ##CB     NOFFS   ##D2     NORTRS  FFFA     NOWD    #3D6   R OCPTR   #1#3
   OFFS    ##D7   R OLAPTR  #1#4     OPBUF   #556     OPBUF2  #44#     OPTRS   #41C
   OPTRSB  #416     P       ###7   R PACT    ###B   R PACTOF  ###9     PAILN   #31F
 R PALT    ###7     PCAL    #2CA     PCL     #2CE   R PCLUST  ###6     PFR     #2D5
   PFRRET  #2DC     PIN#5   #221     PIN#7   #232     PIN3    #218     PINIT   #2#6
   PNS12   #2C5     PNSC    #29F     PNSC#2  #2A5     PNSC#5  #2AA     PNSC1#  #2B2
   PNSC15  #2C6     PNSCRE  #2C9     PRB3    #243     PRB4    #24A     PRB5    #4#3
   PRBAS   #551     PRBAST  #3FA     PRBCLO  #4EA     PRBDE   #52E     PRBFA   #184
   PRBHDR  #23C     PRBL3   #4DC     PRBL4   #4E3     PRBLU1  #145     PRBLU3  #14E
   PRBRAC  #32F     PRBSA   #1AF     PRBT    #3EE     PRBWAC  #336     PRBWCL  #2E4
   PRBWNC  #439     PRBWYN  #48A     PREDB   #29F     PROGNO  ###5     PT      #3D8
   RAIL    #18D     RAIL2   #18F     RAIL5   #198     RAIL7   #1A2     RAIRET  #1A5
   REC#    #4E7     REGFIL  #4D#     RES##   #411     RESTOR  #41#     RFC     ####
   RFC0    ##C5     RIL     #1F7     RTL5    #1FB     RILRET  #2#5   R RLOC    #5#E
   RLOPTR  #523     RLORET  #522     RP1     #186     RPI1#   #1DA   R RPI3    #18A
   RPI4    #1C9     RPI5    #1CB     RPIRET  #1EC     RPTRS   #1ED   R S       ###4
   SABICB  #447     SABO#   #3#2     SABRET  #469     SACLS   #312     SAVE    #445
   SAVEP   #4##     SAVEP#  #4#B     SBIT    #31C     SBMPR   #316     SBYNP   #2F9
   SETBIT  #2FF     SETP    #28F     SETPR   #289   R SRLNTH  ##24     STACT   #1A8
   STRG    #54D     STRLTH  #54F     SVC     CAE#     SVECTR  ###B     TEMP    #555
   TERMTR  #3F5     TPPTR   #4DB     TPTR    #5#D     WABUF   #33D     WC5     #45D
   WC7     #464     WCL     #2DD     WCM     #38D     WCM5    #3C4     WCMRET  #3D5
   WCYRET  #494     WD#     #4E8     WD1     #492     WD2     #493     WDCNT   #4E9
   WNC5    #42E     WNC7    #431     WNCN    #41C     WNCRET  #446     WR      #4B1
   WS#5    #53A   R WSTR    #539     WTC     ###1     WTC0    ##C6     WYM     #46A
   X       ###2   R XB      ###3     YNPBUF  ##D8   R YNPMA0  ###E     YNPMAT  ##3B
   YOFFS   ##D3
```

NO ERRORS NO WARNINGS

V7#08S 19/#3/84 16:12:31

```
V8#2#S    19/#3/84    16:12:38

 98#A OFF LINE ASSEMBLER                                    SHEET    1
 ROUTINES
                    1  *          SUBROUTINES
                    2  *          TI98#A
                    3  *
                    4  *          K. VO
                    5  *
                    6  *          ISSUE #
                    7  *          REVISION #
                    8  *
                    9          IDT  VA#
                   1#          HED  ROUTINES
                   11          DEF  BI2DAS,BI2HAS,WTC,HAS2BI,GETCHA,RFC,DEC2BI
                   12          DEF  HDER#,ACLS##,M12,ACL#SS,M13,HLA#L1
                   13  *
                   14          REF  IPLN,IPLNS,IPLNX,MRKIP
```

```
                    15          PEJ
                    16  *
                    17  *
    ####    18  A       EQU  #
    ###1    19  E       EQU  1
    ###2    2#  X       EQU  2
    ###3    21  M       EQU  3
    ###4    22  S       EQU  4
    ###5    23  L       EQU  5
    ###6    24  B       EQU  6
    ###7    25  P       EQU  7
                    26  *
    ###1    27  BR      EQU  1
    ###3    28  XB      EQU  3
    CAE#    29  SVC     OPD  >CAE#,3
    BA8D    3#  LFCR    EQU  >BA8D
    A##D    31  BBCR    EQU  >A##D
```

ROUTINES

```
                    32  *       PEJ
                    33  *
                    34  *
                    35  *       CONVERT FROM BINARY TO DEC ASCII
                    36  *            LEADING ZERO(S) SUPPRESSED
                    37  *
                    38  *            ENTRY  : (A)=BIN VALUE
                    39  *
                    40  *            RETURN : (X) 1ST MS DIGI
                    41  *                     (A),(E)=4 LS ASCII DIGITS
                    42  *
                    43  *
 0000  17FB         44  BI2DAS LDX  =-5
 0001  0F00         45          LDE  =0
 0002  1800         46          @LDM =>0000
 0003  0000
                    47  *
 0004  6600         48  COMP    @CPL DCONST,X      CONVERSION UNIT
P0005  0096
 0006  CDC0         49          SLE                UNIT .LE. VALUE ?
 0007  7804         50          BRU  NXUNIT        NO
 0008  2E00         51          @SUB DCONST,X      YES - SUBSTRACT
P0009  0096
 000A  C311         52          RIN  E,E                AND COUNT
 000B  78F8         53          BRU  COMP          KEEP GOING
                    54  *
 000C  C0B1         55  NXUNIT  RAD  M,E           ASCII DIGIT
 000D  8A1D         56          STE  ASBUF,X
 000E  0F00         57          LDE  =0            RESET (E)
 000F  4FF4         58          BIX  COMP          CONTINUE CONVERSION TO 5 DIGITS
                    59  *                          SUPPRESS LEADING ZERO(S)
 0010  17FB         60          LDX  =-5
 0011  0000         61          @LDA =>0000
 0012  0000
 0013  6A17         62  SUPZER  CPA  ASBUF,X
 0014  CD20         63          SEQ                ZERO ?
 0015  7806         64          BRU  PACK          NO - LEADING NON-ZERO - STOP SUPPR
 0016  CCA2         65          SNO  X             YES - LAST DIGIT ?
 0017  7804         66          BRU  PACK          YES
 0018  0000         67          @LDE =>00A0        NO - SUPPRESS ZERO
 0019  00A0
 001A  8A10         68          STE  ASBUF,X
 001B  4FF7         69          BIX  SUPZER
                    70  *
 001C  1009         71  PACK    LDX  ASBUF-5       (X)=MS DIGIT
 001D  0008         72          LDA  ASBUF-2       2 LS DIGITS
 001E  C8C8         73          LLA  8
 001F  200A         74          ADD  ASBUF-1
 0020  C501         75          RMO  A,E
 0021  0005         76          LDA  ASBUF-4       2 MIDDLE DIGITS
 0022  C8C8         77          LLA  8
 0023  2004         78          ADD  ASBUF-3
 0024  C557         79          RMO  L,P           RETURN
 0025               80  ASBUFR  BSS  0
 0025               81          BSS  6             6 ASCII DIGITS
 002B               82  ASBUF   BSS  0
```

```
                    83  *
  >002B             84  REGB   BSS  1
  >002C             85  WSRETN BSS  1
```

```
                    86          PEJ
                    87  *
                    88  *       CONVERT FROM BINARY TO HEX ASCII
                    89  *
                    9#  *            ENTRY  : (A)=BIN VALUE
                    91  *                     (E)=-1 LEADING ZERO(S) SUPPRESSED
                    92  *
                    93  *            RETURN : (A)=2 MS ASCII DIGITS
                    94  *                     (E)=2 LS ACIII DIGITS
                    95  *                     (X)=' >'
                    96  *                     (M) DESTROYED
                    97  *
  #2D               98  BI2HAS BSS  #
  #2D  8833         99         STE  ZSUPPR       SAVE ZERO SUPPRESSION
  #2E  17FC      1##           LDX  =-4
  #2F  #F##      1#1           LDE  =#
  #3#  18##      1#2           @LDM =>##B#
  #31  ##B#
                 1#3  *
  #32  622E      1#4  HCOMP    CPL  HCONST,X     CONVERSION UNIT
  #33  CDC#      1#5          SLE                UNIT .LE. VALUE ?
  #34  78#3      1#6          BRU  BI##          NO
  #35  2A28      1#7          SUB  HCONST,X      YES - SUBSTRACT
  #36  C311      1#8          RIN  E,E                 AND COUNT
  #37  78FA      1#9          BRU  HCOMP         KEEP GOING
                 11#  *
  #38  C79#      111  BI##     REX  E,A
  #39  18##      112          @LDM =>##B#        ASSUME <= 9
  #3A  ##B#
  #3B  6F#A      113          CPA  =>A           DIG OR CHAR ?
  #3C  CDC#      114          SLE
  #3D  78#3      115          BRU  BI#5          DIGIT
  #3E  2F#A      116          SUB  =>A           CHAR
  #3F  18##      117          @LDM =>##C1
  #4#  ##C1
  #41  C#B#      118  BI#5     RAD  M,A           HEX ASCII
  #42  C79#      119          REX  E,A
  #43  8AE7      12#          STE  ASBUF,X
  #44  #F##      121          LDE  =#            RESET (E)
  #45  4#EC      122          BIX  HCOMP         CONTINUE CONVERSION TO 4 DIGITS
                 123  *
                 124  *            SUPPRESS LEADING ZERO(S)
                 125  *
  #46  ##1A      126          LDA  ZSUPPR        ZERO SUPPRESSION ?
  #47  CC2#      127          SOO  A
  #48  78#C      128          BRU  BI#8          NO
  #49  17FC      129          LDX  =-4           YES
  #4A  ####      13#          @LDA =>##B#
  #4B  ##B#
  #4C  6ADE      131  SUPPZE   CPA  ASBUF,X       ZERO ?
  #4D  CD2#      132          SEQ
  #4E  78#6      133          BRU  BI#8          NO - STOP SUPPRESSION
  #4F  CCA2      134          SNO  X             YES- LAST DIGIT ?
  #5#  78#4      135          BRU  BI#8          YES - LAST ZERO IS NOT SUPPRESSED
  #51  #B##      136          @LDE =>##A#        NO - SUPPRESS
  #52  ##A#
```

```
990A OFF LINE ASSEMBLER                                          SHEET   6
ROUTINES
  0053  8AD7   137          STE   ASBUF,X
  0054  40F7   138          BIX   SUPPZE
               139  *
  0055  00D3   140  BI00    LDA   ASBUF-2      2 LS DIGITS
  0056  C8C8   141          LLA   8
  0057  20D2   142          ADD   ASBUF-1
  0058  C501   143          RMO   A,E
  0059  00CD   144          LDA   ASBUF-4      2 MS DIGITS
  005A  C8C8   145          LLA   8
  005B  20CC   146          ADD   ASBUF-3
  005C  C567   147          RMO   L,P          RETURN
               148  *
  005D  1000   149          DATA  >1000
  005E  0100   150          DATA  >100
  005F  0010   151          DATA  >10
  0060  0001   152          DATA  >1
  0061         153  HCONST  BSS   0
  0061         154  ZSUPPR  BSS   1            -1 : ZERO SUPPR. REQUESTED
```

```
                    155         PEJ
                    156  *
                    157  *      CONVERT DEC ASCII STRING INTO BINARY VALUE.
                    158  *      STRING LENGTH IN (X) OR WHEN CR OR BLANK IS ENCOUNTERED,
                    159  *      WHICHEVER COMES FIRST.
                    160  *
                    161  *      - ENTRY  : (A)= BYTE POINTER TO STRING
                    162  *                 (X)= # OF ASCIIS
                    163  *      - RETURN : (A)= CONVERTED BINARY VALUE
                    164  *                 (X)=-1 ON INVALID CHARACTER
                    165  *
0062                166  DEC2BI BSS  0
0062 0F00           167         LDE  =0             RESET ASCII COUNTER
0063 0064           168         STE  RESULT         CLEAR RESULT
0064 DB0F           169         TABZ 15             IS 1ST CHAR IN LH OR RH BYTE ?
0065 0F01           170         LDE  =1             RH
0066 0063           171         STE  COUNT
0067 C041           172         LRA  1              FORM WORD OFFSET
0068 0066           173         STA  TEMPTR
0069 C122           174         RCO  X,X            -# OF ASCIIS
006A 005F           175  DEC00  LDA  COUNT
006B C041           176         LRA  1              WORD OFFSET
006C 205C           177         ADD  TEMPTR
006D 0001           178         STA  $+2
006E 0400           179         LDA  *$+1           GET WORD CONTAINING THE ASCII
006F                180         BSS  1
0070 DB3F           181         TMBO 15,COUNT       IF IN LH BYTE
P 0071 00CA
0072 C040           182         LRA  0              ..THEN SHIFT TO RH BYTE
0073 3000           183         @AND =>00FF         KEEP THE ASCII
0074 00FF
0075 6800           184         @CPA =>000D         CR ?
0076 000D
0077 CDA0           185         SNE
0078 7814           186         BRU  DEC03          YES - STRING TERMINATES HERE
0079 6800           187         @CPA =>00A0         BLANK ?
007A 00A0
007B CDA0           188         SNE
007C 7810           189         BRU  DEC03          YES
                    190  *
                    191  *           CHECK FOR LEGAL HEX ASCII
                    192  *
007D 6800           193         @CPA =>00B0         '0' <= ASCII <= '9'
007E 00B0
007F CDC0           194         SLE
0080 780E           195         BRU  DEC04          NO - INVALID CHAR
0081 6800           196         @CPA =>00B9
0082 00B9
0083 CD80           197         SGE
0084 780A           198         BRU  DEC04          NO - INVALID CHAR
0085 2800           199         @SUB =>00B0         YES - GET BINARY VALUE
0086 00B0
0087 9A0E           200         MPY  DCONST,X
0088 C610           201         RMO  E,A
0089 203E           202         ADD  RESULT
008A 803D           203         STA  RESULT
```

```
ROUTINES
  0008  503E   204          IMO  COUNT
  000C  40DD   205          BIX  DEC00
  000D  003A   206  DEC03  LDA  RESULT     GET FINAL RESULT
  000E  C557   207          RMO  L,P       ..RETURN
               208  *
  000F  17FF   209  DEC04  LDX  =-1        SET ERROR ON RETURN
  0010  C557   210          RMO  L,P
               211  *
               212  *
  0011  2710   213          DATA 10000
  0012  03E8   214          DATA 1000
  0013  0064   215          DATA 100
  0014  000A   216          DATA 10
  0015  0001   217          DATA 1
  0016         218  DCONST BSS  0
```

```
                    219          PEJ
                    220  *
                    221  *            CONVERT FROM HEX ASCII TO BINARY VALUE
                    222  *
                    223  *            ENTRY  : (A) POINTER TO ASCII STRING
                    224  *                     (X)= # OF ASCIIS
                    225  *
                    226  *            RETURN : (A) = CONVERTED BINARY VALUE
                    227  *                     (X) = -1 ON UNVALID HEX CHARACTER
                    228  *
  0096              229  HAS2BI BSS   0
  0096  8032        230          STA   TEMPTR       SAVE ASCII POINTER
  0097  0700        231          LDA   =0           RESET ASCII COUNTER
  0098  8031        232          STA   COUNT
  0099  802E        233          STA   RESULT       CLEAR RESULT
  009A  C122        234          RCO   X,X          - # OF ASCII
  009B  C322        235          RIN   X,X
  009C  002D        236  HAS00  LDA   COUNT
  009D  C841        237          LRA   1            FORM WORD OFFSET
  009E  202A        238          ADD   TEMPTR
  009F  8001        239          STA   $+2
  00A0  0400        240          LDA   *$+1         GET WORD CONTAINING THE ASCII
  00A1              241          BSS   1
  00A2  DB3F        242          TMBO  15,COUNT     IF IN LH BYTE
P 00A3  00CA
  00A4  C848        243          LRA   8            THEN SHIFT IT TO RH BYTE
  00A5  3800        244          @AND  =>00FF       KEEP THE CURRENT ASCII
  00A6  00FF
                    245  *
                    246  *            CHECK FOR LEGAL HEX ASCII
                    247  *
  00A7  6800        248          @CPA  =>00B0       '0' <= ASCII <= '9' ?
  00A8  00B0
  00A9  CDC0        249          SLE
  00AA  7807        250          BRU   HAS01        NO
  00AB  6800        251          @CPA  =>00B9
  00AC  00B9
  00AD  CD80        252          SGE
  00AE  7803        253          BRU   HAS01        NO
  00AF  2B00        254          @SUB  =>00B0       YES - GET BINARY VALUE
  00B0  00B0
  00B1  780B        255          BRU   HAS02
                    256  *
  00B2  6800        257  HAS01  @CPA  =>00C1       'A' <= ASCII <= 'F' ?
  00B3  00C1
  00B4  CDC0        258          SLE
  00B5  7810        259          BRU   HAS04        NO
  00B6  6800        260          @CPA  =>00C6
  00B7  00C6
  00B8  CD80        261          SGE
  00B9  780C        262          BRU   HAS04        NO
  00BA  2800        263          @SUB  =>00C0       YES - GET BINARY VALUE
  00BB  00C0
  00BC  2709        264          ADD   =9
  00BD  9AA3        265  HAS02  MPY   HCONST,X
  00BE  C8CF        266          LLA   15           TO PLACE RESULT BETWEEN
```

```
98#A OFF LINE ASSEMBLER                                          SHEET   1#
ROUTINES
  ##BF   C#9#   267          RAD   E,A       ...#-FFFF INTO (A)
  ##C#   2##7   268          ADD   RESULT
  ##C1   8##6   269          STA   RESULT
  ##C2   5##7   27#          IMO   COUNT
  ##C3   4#D8   271          BIX   HAS##
  ##C4   ###3   272          LDA   RESULT
  ##C5   C557   273          RMO   L,P
                274  *
  ##C6   17FF   275  HAS#4   LDX   =-1       SET ERROR ON RETURN
  ##C7   C557   276          RMO   L,P
                277  *
  ##CB          278  RESULT  BSS   1         RESULT OF CONVERSION
  ##C9          279  TEMPTR  BSS   1         TEMP POINTER
  ##CA          28#  COUNT   BSS   1         COUNT OF ASCII
```

```
                    281           PEJ
                    282  *
                    283  *        GET A CHARACTER FROM A STRING
                    284  *
                    285  *        ENTRY  : (M)= START ADDR. OF STRING
                    286  *                 (A)= BYTE OFFSET WITHIN STRING
                    287  *
                    288  *        RETURN : (A)= THE CHARACTER, RH-JUSTIFIED
                    289  *            ALL OTHER REGISTERS PRESERVED
                    290  *
    0CB             291  GETCHA BSS  0
    0CB  0A0A       292           STA  BOFF          SAVE BYTE OFFSET
    0CC  CB41       293           LRA  1             CONVERT TO WORD OFFSET
    0CD  C800       294           RAD  M,A           FORM POINTER TO CHARACTER
    0CE  80D8       295           STA  TPTR
    0CF  A407       296           LDA  *TPTR         GET THE CHARACTER
    0D0  DB3F       297           TMBO 15,BOFF       ...IN RH BYTE
P   0D1  0D6
    0D2  C848       298           LRA  8
    0D3  3800       299           @AND =>00FF
    0D4  00FF
    0D5  C557       300           RMO  L,P           RETURN
                    301  *
    0D6             302  BOFF    BSS  1             BYTE OFFSET
    0D7             303  TPTR    BSS  1             TEMPORARY POINTER
```

```
                    3#4          PEJ
                    3#5  *
                    3#6  *       SUBROUTINE RFC (READ FROM CONSOLE) SETS UP A PRB FOR READING
                    3#7  *       FROM LUN # AND THEN ISSUES THE READ
                    3#8  *
                    3#9  *       ON RETURN
                    31#  *         (M)=NO. OF CHARS. IN I/P STRING EXCLUSIVE OF THE CR
                    311  *            =-1 FOR A READ ERROR OR A LINE TOO LONG
                    312  *         ALL OTHER REGISTERS ARE PRESERVED
                    313  *
  ##D8  8#2C        314  RFC     STA  R9+4        SAVE ENTRY (A),(L),(X)
  ##D9  C55#        315          RMO  L,A
  ##DA  8#2B        316          STA  R9+5
  ##DB  9#2B        317          STX  R9+6
                    318  *
  ##DC  #75#        319          LDA  =8#         RESET THE I/P CHAR. COUNT
  ##DD  8#25        32#          STA  R9+2
                    321  *
  ##DE  18##        322          @LDM =R9         ISSUE THE READ
P ##DF  #1#1
  ##E#  CAE#        323          SVC  #
  ##E1  DB21        324          TMBZ 1,R9        READ ERROR?
P ##E2  #1#1
  ##E3  78#B        325          BRU  R4          YES:
                    326  *
  ##E4  ##1E        327          LDA  R9+2        (A)=I/P CHAR. COUNT
  ##E5  6F5#        328          CPA  =8#
  ##E6  C7#3        329          RDE  A,M
  ##E7  CD2#        33#          SEQ              8# CHARS. READ IN?
  ##E8  78#F        331          BRU  R7          NO:
                    332  *
  ##E9  #4##        337          @LDA IPLNS+39
        ###1
E ##EA  ##27
  ##EB  678D        334          CPL  =>8D
  ##EC  CD2#        335          SEQ              WAS 8#-TH CHAR. A CR?
  ##ED  78#3        336          BRU  R6          NO:
  ##EE  78#9        337          BRU  R7
                    338  *
  ##EF  #7#F        339  R4      LDA  =15
  ##F#  78#1        34#          BRU  $+2
  ##F1  #711        341  R6      LDA  =17
                    342  *
  ##F2  1F##        343          LDM  =#
  ##F3  74##        344          BRL  *WTC###
P ##F4  #1#8        345  WTC### DATA WTC
  ##F5  C5#3        346          RMO  A,M
  ##F6  74FD        347          BRL  *WTC###
  ##F7  1FFF        348          LDM  =-1         SET BAD LINE FLAG
                    349  *
  ##F8  C33#        35#  R7      RIN  M,A         (A)=NO. OF CHARS READ
  ##F9  1C##        351          @LDM IPLNX
        ###2
E ##FA  ####
  ##FB  74##        352          BRL  *MRKIP#
        ###3        353  MRKIP# DATA MRKIP
```

```
E 00FC  0000
  00FD  C703   354          RDE  A,M
  00FE  0006   355          LDA  R9+4        RESTORE ENTRY (A),(X)
  00FF  1007   356          LDX  R9+6
  0100  7C05   357          BRU  *R9+5
               358  *
  0101  0400   359  R9       DATA >0400       0 LUN 0
  0102  0000   360          DATA 0           1 READ ASCII OPCODE
  0103         361          BSS  1           2 CHAR. COUNT
        0001   362          DATA IPLNS       3 DATA BUFFER ADDR.
E 0104  0000
  0105         363          BSS  3           4-5 ENTRY (A),(L),(X)
```

```
                    364          PEJ
                    365  *
                    366  *       SUBROUTINE TO WRITE A MESSAGE TO CONSOLE
                    367  *
                    368  *       ENTRY: (M): MESSAGE INDEX - ALL REGS PRESERVED
                    369  *
  0100              370  WTC     BSS  0
  0100  D8E0        371          @SRF REGF2          SAVE REG FILE
P 0101  011D
  0102  C7B2        372          REX  M,X
  0103  C564        373          RMO  B,S            SAVE (B)
                    374  *
  0104  0218        375          LDA  M0,X           (A)=MESSAGE STRING ADDRESS
  0105  C506        376          RMO  A,B             (B) POINTS TO COUNT + MESSAGE
  0106  0100        377          LDA  0,BR           (A)=CHARACTER COUNT
  0107  C361        378          RIN  B,E             (E)=START OF MESSAGE
  0108  A000A       379          DST  PWTC+2
                    380  *
  010A  C532        381          RMO  M,X
                    382  *
  010B  1800        383          @LDM =PWTC          CALL SUPERVISOR TO WRITE
P 010C  0119
  010D  CAE0        384          SVC  0
  010E  C546        385          RMO  S,B            RESTORE (B)
  010F  D8A0        386          @LRF REGF2          RESTORE REG FILE
P 0110  011D
  0111  C557        387          RMO  L,P
                    388  *
                    389  *       P.R.B
                    390  *
  0112  0000        391  PWTC    DATA 0
  0113  0002        392          DATA 2
  0114              393          BSS  1
  0115              394          BSS  1
                    395  *
  0116              396  REGF2   BSS  8              REGISTER FILE
P 0125  0136        397  M0      DATA M0,M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,M11
P 0126  0138
P 0127  013A
P 0128  0141
P 0129  0148
P 012A  0164
P 012B  0166
P 012C  0173
P 012D  0182
P 012E  0193
P 012F  01A0
P 0130  01A3
P 0131  01AF        398          DATA M12,M13,M14,M15,M16
P 0132  01C2
P 0133  0155
P 0134  01C6
P 0135  01D3
                    399  *
  0136  0002        400  M0      DATA 2
  0137  0A0D        401          DATA LFCR
```

```
ROUTINES
  0138  0002   402  M1     DATA 2
  0139  A000   403         DATA '  '
                404  *
  013A  000C   405  M2     DATA 12
  013B  C1C2   406         DATA 'ABL EDITOR',LFCR
  013C  CCA0
  013D  C5C4
  013E  C9D4
  013F  CFD2
  0140  8A8D
  0141  000C   407  M3     DATA 12
  0142  C6C9   408         DATA 'FILE NAME > '
  0143  CCC5
  0144  A0CE
  0145  C1CD
  0146  C5A0
  0147  BEA0
  0148  0035   409  M4     DATA 53
  0149  CDC1   410         DATA 'MACHINE SPECIFICATION ',LFCR
  014A  C3C8
  014B  C9CE
  014C  C5A0
  014D  D3D0
  014E  C5C3
  014F  C9C6
  0150  C9C3
  0151  C1D4
  0152  C9CF
  0153  CEA0
  0154  8A8D
  0155  001C   411  M14    DATA 28
  0156  A0A0   412         DATA '   ENTER SUBROUTINE NAMES',LFCR
  0157  A0C5
  0158  CED4
  0159  C5D2
  015A  A0D3
  015B  D5C2
  015C  D2CF
  015D  D5D4
  015E  C9CE
  015F  C5A0
  0160  CEC1
  0161  CDC5
  0162  D3FF
  0163  8A8D
  0164  0002   413  M5     DATA 2
  0165  A0A0   414         DATA '  '
  0166  0018   415  M6     DATA 24
  0167  A0A0   416         DATA '   NUMBER OF CLUSTERS  >'
  0168  A0CE
  0169  D5CD
  016A  C2C5
  016B  D2A0
  016C  CFC6
  016D  A0C3
  016E  CCD5
```

```
#16F  D3D4
#17#  C5D2
#171  D3A#
#172  A#BE
#173  ##1C   417  M7     DATA 28
#174  A#A#   418         DATA '   NUMBER OF ALTERNATIVES  >'
#175  A#CE
#176  D5CD
#177  C2C5
#178  D2A#
#179  CFC6
#17A  A#C1
#17B  CCD4
#17C  C5D2
#17D  CEC1
#17E  D4C9
#17F  D6C5
#18#  D3A#
#181  A#BE
#182  ##2#   419  M8     DATA 32
#183  A#A#   42#         DATA '   ALTERNATIVE SPECIFICATION :',LFCR
#184  A#C1
#185  CCD4
#186  C5D2
#187  CEC1
#188  D4C9
#189  D6C5
#18A  A#D3
#18B  D#C5
#18C  C3C9
#18D  C6C9
#18E  C3C1
#18F  D4C9
#19#  CFCE
#191  A#BA
#192  8A8D
#193  ##18   421  M9     DATA 24
#194  C5D2   422         DATA 'ERROR ON FILE OPENING ',LFCR
#195  D2CF
#196  D2A#
#197  CFCE
#198  A#C6
#199  C9CC
#19A  C5A#
#19B  CFD#
#19C  C5CE
#19D  C9CE
#19E  C7A#
#19F  8A8D
#1A#  ###4   423  M1#    DATA 4
#1A1  A#A#   424         DATA '   >'
#1A2  A#BE
#1A3  ##16   425  M11    DATA 22
#1A4  C9CD   426         DATA 'IMPROPER INPUT LINE',LFCR
#1A5  D#D2
#1A6  CFD#
```

```
ROUTINES
  #1A7  C5D2
  #1A8  A#C9
  #1A9  CED#
  #1AA  D5D4
  #1AB  A#CC
  #1AC  C9CE
  #1AD  C5FF
  #1AE  8A8D
  #1AF  ##24   427  M12    DATA 36
  #1B#  C1C2   428         DATA 'ABL PROGRAM CREATED IN FILE MMIRRS',LFCR
  #1B1  CCA#
  #1B2  D#D2
  #1B3  CFC7
  #1B4  D2C1
  #1B5  CDA#
  #1B6  C3D2
  #1B7  C5C1
  #1B8  D4C5
  #1B9  C4A#
  #1BA  C9CE
  #1BB  A#C6
  #1BC  C9CC
  #1BD  C5A#
  #1BE  CDCD
  #1BF  C9D2
  #1C#  D2D3
  #1C1  8A8D
  #1C2  ##86   429  M13    DATA 6
  #1C3  A#A#   43#         DATA '     >'
  #1C4  A#A#
  #1C5  A#BE
  #1C6  ##18   431  M15    DATA 24
  #1C7  D#D2   432         DATA 'PROGRAM SPECIFICATION ',LFCR
  #1C8  CFC7
  #1C9  D2C1
  #1CA  CDA#
  #1CB  D3D#
  #1CC  C5C3
  #1CD  C9C6
  #1CE  C9C3
  #1CF  C1D4
  #1D#  C9CF
  #1D1  CEA#
  #1D2  8A8D
  #1D3  ##14   433  M16    DATA 2#
  #1D4  A#A#   434         DATA '   PROGRAM NUMBER > ',LFCR
  #1D5  A#D#
  #1D6  D2CF
  #1D7  C7D2
  #1D8  C1C0
  #1D9  A#CE
  #1DA  D5CD
  #1DB  C2C5
  #1DC  D2A#
  #1DD  8EA#
  #1DE  8A8D
```

```
                    435          PEJ
                    436  *
                    437  *       HEADERS
                    438  *
 #1DF               439  HDER#   BSS  #
 #1DF   D#D2        44#  HLABL1  DATA 'PROGNO DATA #      ',BBCR
 #1E#   CFC7
 #1E1   CECF
 #1E2   A#C4
 #1E3   C1D4
 #1E4   C1A#
 #1E5   B#A#
 #1E6   A#A#
 #1E7   A#A#
 #1E8   A##D
 #1E9   D#C3        441          DATA 'PCLUST DATA #      ',BBCR
 #1EA   CCD5
 #1EB   D3D4
 #1EC   A#C4
 #1ED   C1D4
 #1EE   C1A#
 #1EF   B#A#
 #1F#   A#A#
 #1F1   A#A#
 #1F2   A##D
 #1F3   D#C1        442          DATA 'PALT   DATA #      ',BBCR
 #1F4   CCD4
 #1F5   A#A#
 #1F6   A#C4
 #1F7   C1D#
 #1F8   C1A#
 #1F9   B#A#
 #1FA   A#A#
 #1FB   A#A#
 #1FC   A##D
 #1FD   D#C1        443          DATA 'PACT   DATA #      ',BBCR
 #1FE   C3D4
 #1FF   A#A#
 #2##   A#C4
 #2#1   C1D4
 #2#2   C1A#
 #2#3   B#A#
 #2#4   A#A#
 #2#5   A#A#
 #2#6   A##D
 #2#7   D#C1        444          DATA 'PACTOF DATA #      ',BBCR
 #2#8   C3D4
 #2#9   CFC6
 #2#A   A#C4
 #2#B   C1D4
 #2#C   C1A#
 #2#D   B#A#
 #2#E   A#A#
 #2#F   A#A#
 #21#   A##D
 #211   C1C3        445          DATA 'ACLST# DATA #      ',BBCR
```

```
ROUTINES
  #212  CCD3
  #213  D4C#
  #214  A#C4
  #215  C1D4
  #216  C1A#
  #217  B#A#
  #218  A#A#
  #219  A#A#
  #21A  A##D
  #21B  D3D6   446        DATA 'SVECTR DATA #     ',BBCR
  #21C  C5C3
  #21D  D4D2
  #21E  A#C4
  #21F  C1D4
  #22#  C1A#
  #221  B#A#
  #222  A#A#
  #223  A#A#
  #224  A##D
  #225  C3CC   447        DATA 'CLSMA@ DATA CLSMAT',BBCR
  #226  D3CD
  #227  C1C#
  #228  A#C4
  #229  C1D4
  #22A  C1A#
  #22B  C3CC
  #22C  D3CD
  #22D  C1D4
  #22E  A##D
  #22F  CEC3   448        DATA 'NCLMA@ DATA NCLMAT',BBCR
  #23#  CCCD
  #231  C1C#
  #232  A#C4
  #233  C1D4
  #234  C1A#
  #235  CEC3
  #236  CCCD
  #237  C1D4
  #238  A##D
  #239  D9CE   449        DATA 'YNPMA@ DATA YNPMAT',BBCR
  #23A  D#CD
  #23B  C1C#
  #23C  A#C4
  #23D  C1D4
  #23E  C1A#
  #23F  D9CE
  #24#  D#CD
  #241  C1D4
  #242  A##D
  #243  CDD#   45#        DATA 'MPRMA@ DATA MPRMAT',BBCR
  #244  D2CD
  #245  C1C#
  #246  A#C4
  #247  C1D4
  #248  C1A#
  #249  CDD#
```

```
ROUTINES
  #24A  D2CD
  #24B  C1D4
  #24C  A##D
  #24D  C1D#   451          DATA 'APTR#0 DATA APTR#  ',BBCR
  #24E  D4D2
  #24F  B#C#
  #25#  A#C4
  #251  C1D4
  #252  C1A#
  #253  C1D#
  #254  D4D2
  #255  B#A#
  #256  A##D
  #257  C2D2   452          DATA 'BREG   DATA #      ',BBCR
  #258  C5C7
  #259  A#A#
  #25A  A#C4
  #25B  C1D4
  #25C  C1A#
  #25D  B#A#
  #25E  A#A#
  #25F  A#A#
  #26#  A##D
  #261  CCD2   453          DATA 'LREG   DATA #      ',BBCR
  #262  C5C7
  #263  A#A#
  #264  A#C4
  #265  C1D4
  #266  C1A#
  #267  B#A#
  #268  A#A#
  #269  A#A#
  #26A  A##D
  #26B  AAA#   454          DATA '*                  ',BBCR
  #26C  A#A#
  #26D  A#A#
  #26E  A#A#
  #26F  A#A#
  #27#  A#A#
  #271  A#A#
  #272  A#A#
  #273  A#A#
  #274  A##D
                455  *
  #275  A#C4    456  ACLS#0 DATA ' DATA ACLS##',BBCR
  #276  C1D4
  #277  C1A#
  #278  C1C3
  #279  CCD3
  #27A  B#B#
  #27B  A##D
  #27C  A#C4    457          DATA ' DATA ACLS#1',BBCR
  #27D  C1D4
  #27E  C1A#
  #27F  C1C3
  #28#  CCD3
```

```
#281  B#B1
#282  A##D
#283  A#C4   458        DATA ' DATA ACLS#2',B
#284  C1D4
#285  C1A#
#286  C1C3
#287  CCD3
#288  B#B2
#289  A##D
#28A  A#C4   459        DATA ' DATA ACLS#3',BBCR
#28B  C1D4
#28C  C1A#
#28D  C1C3
#28E  CCD3
#28F  B#B3
#29#  A##D
#291  A#C4   46#        DATA ' DATA ACLS#4',BBCR
#292  C1D4
#293  C1A#
#294  C1C3
#295  CCD3
#296  B#B4
#297  A##D
#298  A#C4   461        DATA ' DATA ACLS#5',BBCR
#299  C1D4
#29A  C1A#
#29B  C1C3
#29C  CCD3
#29D  B#B5
#29E  A##D
#29F  A#C4   462        DATA ' DATA ACLS#6',BBCR
#2A#  C1D4
#2A1  C1A#
#2A2  C1C3
#2A3  CCD3
#2A4  B#B6
#2A5  A##D
#2A6  A#C4   463        DATA ' DATA ACLS#7',BBCR
#2A7  C1D4
#2A8  C1A#
#2A9  C1C3
#2AA  CCD3
#2AB  B#B7
#2AC  A##D
#2AD  A#C4   464        DATA ' DATA ACLS#8',BBCR
#2AE  C1D4
#2AF  C1A#
#2B#  C1C3
#2B1  CCD3
#2B2  B#B8
#2B3  A##D
#2B4  A#C4   465        DATA ' DATA ACLS#9',BBCR
#2B5  C1D4
#2B6  C1A#
#2B7  C1C3
#2B8  CCD3
```

```
#2B9  B#B9
#2BA  A##D
#2BB  A#C4   466        DATA ' DATA ACLS1#',BBCR
#2BC  C1D4
#2BD  C1A#
#2BE  C1C3
#2BF  CCD3
#2C#  B1B#
#2C1  A##D
#2C2  A#C4   467        DATA ' DATA ACLS11',BBCR
#2C3  C1D4
#2C4  C1A#
#2C5  C1C3
#2C6  CCD3
#2C7  B1B1
#2C8  A##D
#2C9  A#C4   468        DATA ' DATA ACLS12',BBCR
#2CA  C1D4
#2CB  C1A#
#2CC  C1C3
#2CD  CCD3
#2CE  B1B2
#2CF  A##D
#2D#  A#C4   469        DATA ' DATA ACLS13',BBCR
#2D1  C1D4
#2D2  C1A#
#2D3  C1C3
#2D4  CCD3
#2D5  B1B3
#2D6  A##D
#2D7  A#C4   47#        DATA ' DATA ACLS14',BBCR
#2D8  C1D4
#2D9  C1A#
#2DA  C1C3
#2DB  CCD3
#2DC  B1B4
#2DD  A##D
#2DE  A#C4   471        DATA ' DATA ACLS15',BBCR
#2DF  C1D4
#2E#  C1A#
#2E1  C1C3
#2E2  CCD3
#2E3  B1B5
#2E4  A##D
#2E5  A#C4   472        DATA ' DATA ACLS16',BBCR
#2E6  C1D4
#2E7  C1A#
#2E8  C1C3
#2E9  CCD3
#2EA  B1B6
#2EB  A##D
#2EC  A#C4   473        DATA ' DATA ACLS17',BBCR
#2ED  C1D4
#2EE  C1A#
#2EF  C1C3
#2F#  CCD3
```

```
ROUTINES
  #2F1  B187
  #2F2  A##D
  #2F3  A#C4   474          DATA ' DATA ACLS18',BBCR
  #2F4  C1D4
  #2F5  C1A#
  #2F6  C1C3
  #2F7  CCD3
  #2F8  B188
  #2F9  A##D
  #2FA  A#C4   475          DATA ' DATA ACLS19',BBCR
  #2FB  C1D4
  #2FC  C1A#
  #2FD  C1C3
  #2FE  CCD3
  #2FF  B189
  #3##  A##D
  #3#1  A#C4   476          DATA ' DATA ACLS2#',BBCR
  #3#2  C1D4
  #3#3  C1A#
  #3#4  C1C3
  #3#5  CCD3
  #3#6  B2B#
  #3#7  A##D
  #3#8  A#C4   477          DATA ' DATA ACLS21',BBCR
  #3#9  C1D4
  #3#A  C1A#
  #3#B  C1C3
  #3#C  CCD3
  #3#D  B2B1
  #3#E  A##D
  #3#F  A#C4   478          DATA ' DATA ACLS22',BBCR
  #31#  C1D4
  #311  C1A#
  #312  C1C3
  #313  CCD3
  #314  B2B2
  #315  A##D
  #316  A#C4   479          DATA ' DATA ACLS23',BBCR
  #317  C1D4
  #318  C1A#
  #319  C1C3
  #31A  CCD3
  #31B  B2B3
  #31C  A##D
  #31D  A#C4   48#          DATA ' DATA ACLS24',BBCR
  #31E  C1D4
  #31F  C1A#
  #32#  C1C3
  #321  CCD3
  #322  B2B4
  #323  A##D
  #324  A#C4   481          DATA ' DATA ACLS25',BBCR
  #325  C1D4
  #326  C1A#
  #327  C1C3
  #328  CCD3
```

```
ROUTINES
  #329  B2B5
  #32A  A##D
  #32B  A#C4   482          DATA ' DATA ACLS26',BBCR
  #32C  C1D4
  #32D  C1A#
  #32E  C1C3
  #32F  CCD3
  #33#  B2B6
  #331  A##D
  #332  A#C4   483          DATA ' DATA ACLS27',BBCR
  #333  C1D4
  #334  C1A#
  #335  C1C3
  #336  CCD3
  #337  B2B7
  #338  A##D
  #339  A#C4   484          DATA ' DATA ACLS28',BBCR
  #33A  C1D4
  #33B  C1A#
  #33C  C1C3
  #33D  CCD3
  #33E  B2B8
  #33F  A##D
  #34#  A#C4   485          DATA ' DATA ACLS29',BBCR
  #341  C1D4
  #342  C1A#
  #343  C1C3
  #344  CCD3
  #345  B2B9
  #346  A##D
  #347  A#C4   486          DATA ' DATA ACLS3#',BBCR
  #348  C1D4
  #349  C1A#
  #34A  C1C3
  #34B  CCD3
  #34C  B3B#
  #34D  A##D
  #34E  A#C4   487          DATA ' DATA ACLS31',BBCR
  #34F  C1D4
  #35#  C1A#
  #351  C1C3
  #352  CCD3
  #353  B3B1
  #354  A##D
  #355  C1C3   488  ACLBSS DATA 'ACLS## BSS #',BBCR
  #356  CCD3
  #357  B#B#
  #358  A#C2
  #359  D3D3
  #35A  A#B#
  #35B  A##D
  #35C  C1C3   489          DATA 'ACLS#1 BSS #',BBCR
  #35D  CCD3
  #35E  B#B1
  #35F  A#C2
  #36#  D3D3
```

```
ROUTINES
  0361  A00B0
  0362  A00D0
  0363  C1C3   490         DATA 'ACLS02 BSS 0',BBCR
  0364  CCD3
  0365  B0B2
  0366  A0C2
  0367  D3D3
  0368  A0B0
  0369  A00D0
  036A  C1C3   491         DATA 'ACLS03 BSS 0',BBCR
  036B  CCD3
  036C  B0B3
  036D  A0C2
  036E  D3D3
  036F  A0B0
  0370  A00D0
  0371  C1C3   492         DATA 'ACLS04 BSS 0',BBCR
  0372  CCD3
  0373  B0B4
  0374  A0C2
  0375  D3D3
  0376  A0B0
  0377  A00D0
  0378  C1C3   493         DATA 'ACLS05 BSS 0',BBCR
  0379  CCD3
  037A  B0B5
  037B  A0C2
  037C  D3D3
  037D  A0B0
  037E  A00D0
  037F  C1C3   494         DATA 'ACLS06 BSS 0',BBCR
  0380  CCD3
  0381  B0B6
  0382  A0C2
  0383  D3D3
  0384  A0B0
  0385  A00D0
  0386  C1C3   495         DATA 'ACLS07 BSS 0',BBCR
  0387  CCD3
  0388  B0B7
  0389  A0C2
  038A  D3D3
  038B  A0B0
  038C  A00D0
  038D  C1C3   496         DATA 'ACLS08 BSS 0',BBCR
  038E  CCD3
  038F  B0B8
  0390  A0C2
  0391  D3D3
  0392  A0B0
  0393  A00D0
  0394  C1C3   497         DATA 'ACLS09 BSS 0',BBCR
  0395  CCD3
  0396  B0B9
  0397  A0C2
  0398  D3D3
```

ROUTINES

```
  #399  A#B#
  #39A  A##D
  #39B  C1C3   498        DATA 'ACLS1# BSS #',BBCR
  #39C  CCD3
  #39D  B1B#
  #39E  A#C2
  #39F  D3D3
  #3A#  A#B#
  #3A1  A##D
  #3A2  C1C3   499        DATA 'ACLS11 BSS #',BBCR
  #3A3  CCD3
  #3A4  B1B1
  #3A5  A#C2
  #3A6  D3D3
  #3A7  A#B#
  #3A8  A##D
  #3A9  C1C3   5##        DATA 'ACLS12 BSS #',BBCR
  #3AA  CCD3
  #3AB  B1B2
  #3AC  A#C2
  #3AD  D3D3
  #3AE  A#B#
  #3AF  A##D
  #3B#  C1C3   5#1        DATA 'ACLS13 BSS #',BBCR
  #3B1  CCD3
  #3B2  B1B3
  #3B3  A#C2
  #3B4  D3D3
  #3B5  A#B#
  #3B6  A##D
  #3B7  C1C3   5#2        DATA 'ACLS14 BSS #',BBCR
  #3B8  CCD3
  #3B9  B1B4
  #3BA  A#C2
  #3BB  D3D3
  #3BC  A#B#
  #3BD  A##D
  #3BE  C1C3   5#3        DATA 'ACLS15 BSS #',BBCR
  #3BF  CCD3
  #3C#  B1B5
  #3C1  A#C2
  #3C2  D3D3
  #3C3  A#B#
  #3C4  A##D
  #3C5  C1C3   5#4        DATA 'ACLS16 BSS #',BBCR
  #3C6  CCD3
  #3C7  B1B6
  #3C8  A#C2
  #3C9  D3D3
  #3CA  A#B#
  #3CB  A##D
  #3CC  C1C3   5#5        DATA 'ACLS17 BSS #',BBCR
  #3CD  CCD3
  #3CE  B1B7
  #3CF  A#C2
  #3D#  D3D3
```

```
#3D1  A#B#
#3D2  A##D
#3D3  C1C3   5#6        DATA 'ACLS18 BSS #',BBCR
#3D4  CCD3
#3D5  B1B8
#3D6  A#C2
#3D7  D3D3
#3D8  A#B#
#3D9  A##D
#3DA  C1C3   5#7        DATA 'ACLS19 BSS #',BBCR
#3DB  CCD3
#3DC  B1B9
#3DD  A#C2
#3DE  D3D3
#3DF  A#B#
#3E#  A##D
#3E1  C1C3   5#8        DATA 'ACLS2# BSS #',BBCR
#3E2  CCD3
#3E3  B2B#
#3E4  A#C2
#3E5  D3D3
#3E6  A#B#
#3E7  A##D
#3E8  C1C3   5#9        DATA 'ACLS21 BSS #',BBCR
#3E9  CCD3
#3EA  B2B1
#3EB  A#C2
#3EC  D3D3
#3ED  A#B#
#3EE  A##D
#3EF  C1C3   51#        DATA 'ACLS22 BSS #',BBCR
#3F#  CCD3
#3F1  B2B2
#3F2  A#C2
#3F3  D3D3
#3F4  A#B#
#3F5  A##D
#3F6  C1C3   511        DATA 'ACLS23 BSS #',BBCR
#3F7  CCD3
#3F8  B2B3
#3F9  A#C2
#3FA  D3D3
#3FB  A#B#
#3FC  A##D
#3FD  C1C3   512        DATA 'ACLS24 BSS #',BBCR
#3FE  CCD3
#3FF  B2B4
#4#   A#C2
#4#1  D3D3
#4#2  A#B#
#4#3  A##D
#4#4  C1C3   513        DATA 'ACLS25 BSS #',BBCR
#4#5  CCD3
#4#6  B2B5
#4#7  A#C2
#4#8  D3D3
```

```
ROUTINES
  #4#9  A##B#
  #4#A  A##0D
  #4#B  C1C3    514         DATA 'ACLS26 BSS #',BBCR
  #4#C  CCD3
  #4#D  B2B6
  #4#E  A#C2
  #4#F  D3D3
  #41#  A##B#
  #411  A##0D
  #412  C1C3    515         DATA 'ACLS27 BSS #',BBCR
  #413  CCD3
  #414  B2B7
  #415  A#C2
  #416  D3D3
  #417  A##B#
  #418  A##0D
  #419  C1C3    516         DATA 'ACLS28 BSS #',BBCR
  #41A  CCD3
  #41B  B2B8
  #41C  A#C2
  #41D  D3D3
  #41E  A##B#
  #41F  A##0D
  #42#  C1C3    517         DATA 'ACLS29 BSS #',BBCR
  #421  CCD3
  #422  B2B9
  #423  A#C2
  #424  D3D3
  #425  A##B#
  #426  A##0D
  #427  C1C3    518         DATA 'ACLS3# BSS #',BBCR
  #428  CCD3
  #429  B3B#
  #42A  A#C2
  #42B  D3D3
  #42C  A##B#
  #42D  A##0D
  #42E  C1C3    519         DATA 'ACLS31 BSS #',BBCR
  #42F  CCD3
  #43#  B3B1
  #431  A#C2
  #432  D3D3
  #433  A##B#
  #434  A##0D
        52#     END
```

```
9@@A OFF LINE ASSEMBLER
ROUTINES
   A        ####      ACL8SS  #355      ACLS#@  #275      ASBUF   ##2B    R ASBUFR  ##25
   B        ####6     B8CR    A##D      BI##    ##38      BI#5    ##41      BI#8    ##55
   BI2DAS   ####      BI2HAS  ##2D      BOFF    ##D6      BR      ###1      COMP    ###4
   COUNT    ##CA      DCONST  ##96      DEC##   ##6A      DEC#3   ##8D      DEC#4   ##8F
   DEC2BI   ##62      E       ###1      GETCHA  ##CB      HAS##   ##9C      HAS#1   ##B2
   HAS#2    ##BD      HAS#4   ##C6      HAS2BI  ##96      HCOMP   ##32      HCONST  ##61
   HDER@    #1DF      HLABL1  #1DF     R IPLN    ####      IPLNX   ###2      IPLNS   ###1
   L        ###5      LFCR    8A8D      M       ###3      M#      #136      M1      #138
   M1#      #1A#      M11     #1A3      M12     #1AF      M13     #1C2      M14     #155
   M15      #1C6      M16     #1D3      M2      #13A      M3      #141      M4      #148
   M5       #164      M6      #166      M7      #173      M8      #182      M9      #193
   M@       #125      MRKIP   ###3      MRKIP@  ##FC      NXUNIT  ###C      P       ###7
   PACK     ##1C      PWTC    #119      R4      ##EF      R6      ##F1      R7      ##F8
   R9       #1#1    R REGB    ##28      REGF2   #11D      RESULT  ##C8      RFC     ##D8
   S        ###4      SUPPZE  ##4C      SUPZER  ##13      SVC     CAE#      TEMPTR  ##C9
   TPTR     ##D7    R WSRETN  ##2C      WTC     #1#8      WTC@@@  ##F4      X       ###2
 R XB       ###3      ZSUPPR  ##61

    NO ERRORS      NO WARNINGS
V8#2#S     19/#3/84     16:14:35
```

```
T4#56S     19/#3/84     16:#6:15

  98#A OFF LINE ASSEMBLER                                          SHEET    1
  ABL INTERPRETOR
                        1  *            ABL INTERPRETOR
                        2  *            TI98#A
                        3  *
                        4  *            K. VO
                        5  *
                        6  *            ISSUE #
                        7  *            REVISION #
                        8  *
                        9            IDT  T5#
                       1#            HED  ABL INTERPRETOR
                       11            DEF  ABLENT,ABLRET
                       12  *
                       13            REF  BI2DAS,BI2HAS,BI2OAS,HAS2BI
                       14  *
  P          #134      15            BRS  ABLPRO
                       16  *
                       17  *       DEBUG = # : ENABLE DEBUGGING MESSAGES
                       18  *               1 : DISABLE
                       19  *
           ###1        2#  DEBUG    EQU  1
```

```
                    21          PEJ
                    22  *
                    23  *
      0000          24  A       EQU  0
      0001          25  E       EQU  1
      0002          26  X       EQU  2
      0003          27  M       EQU  3
      0004          28  S       EQU  4
      0005          29  L       EQU  5
      0006          30  B       EQU  6
      0007          31  P       EQU  7
                    32  *
      0001          33  BR      EQU  1
      0003          34  XB      EQU  3
      CAE0          35  SVC     OPD  >CAE0,3
      0A0D          36  LFCR    EQU  >0A0D
```

```
                    37          PEJ
                    38  *
                    39  *           ABL EXECUTIVE
                    40  *
                    41  *           THIS EXECUTIVE IS CALLED TO EXECUTE PAIR OF
                    42  *           ABL PROGRAM-MACHINE.
                    43  *           IT SERVES AS ENTRY AND EXIT FROM TO/FROM
                    44  *           ALL ABL PROGRAMS.
                    45  *
                    46  *           ENTRY : (B) POINT TO THE CALLING ABL PROGRAM
                    47  *                   (M) POINT TO THE CALLED ABL PROGRAM
                    48  *
                    49  *           RETURN: (B), (S) ARE RESTORED
                    50  *                   RETURN ADDRESS IS THE SAVED CONTENT OF (L)
0000                51  ABLENT BSS  0
0000  C7B6          52          REX  M,B          (B) POINTS TO NEW PROGRAM
                    53  *
0001  C783          54          REX  A,M          SAVE CALLING PROGRAM
0002  810C          55          STA  BREG
0003  C783          56          REX  A,M
                    57  *
0004  C785          58          REX  A,L          SAVE RETURN ADDRESS
0005  810D          59          STA  LREG
0006  C785          60          REX  A,L
                    61  *
                    62  *
                    63  *           ANNOUNCE ENTRY TO NEW PROGRAM
                    64  *
                    65          IF   DEBUG,1,MSGE1
                    66          LDA  PROGNO       PROGRAM NO.
                    67          @BRL B12DAS
                    68          @STE M2+6
                    69          LDM  =2
                    70          @BRL WTC
                    71          LDA  SVECTR       *'STATE VECTOR...' FOR ALT 0
                    72          @BRL B12HAS       *(ALWAYS 0000)
                    73          @DST M8+11        *
                    74          LDM  =8           *
                    75          @BRL WTC          *
0007                76  MSGE1   BSS  0
0007  0700          77          LDA  =0           ALWAYS EXECUTE ALT 0
0008  8102          78          STA  PALT
0009  8104          79          STA  PACTOF
000A  0701          80          LDA  =1           CLUSTER 1
000B  8101          81          STA  PCLUST
000C  785C          82          BRU  ALTEXE       GO EXECUTE ALTERNATIVE 0
```

```
                        83          PEJ
                        84  *
                        85  *       RETURN TO THE PREVIOUS ABL PROGRAM-MACHINE
                        86  *            FROM THE CURRENT ONE.
                        87  *            THIS ROUTINE IS CALLED WHEN ABL EXECUTION IS COMPLETED
                        88  *
                        89  *            THIS ROUTINE RESTORES (B) & (S) TO THE PREVIOUS
                        90  *            ABL PROGRAM-MACHINE WHICH ARE STORED IN THE CURRENT
                        91  *            PROGRAM
                        92  *
 0000                   93  ABLRET BSS  0
                        94          IF   DEBUG,1,MSGE2
                        95          LDA  PROGNO          *FOR DEBUGGING
                        96          @BRL BI2DAS          *
                        97          @STE M10+11          *
                        98          LDM  -10             *
                        99          @BRL WTC             *
 0000                  100  MSGE2  BSS  0
 0000  190D            101          LDM  LREG            RETURN ADDR
 0001  C535            102          RMO  M,L
                       103  *
                       104  *
 0002  190C            105          LDM  BREG                -    (B)
 0003  C536            106          RMO  M,B
                       107  *
 0004  C557            108          RMO  L,P             RETURN TO THE CALLING ABL
```

```
                    109         PEJ
                    110  *
                    111  *              PREDICATE EVALUATION
                    112  *
                    113  *              THIS ROUTINE EVALUATES CLUSTER, NEXT CLUSTER AND
                    114  *              NEXT CLUSTER MATRICES TO FIND OUT THE NEXT
                    115  *              ALTERNATIVE TO BE EXECUTED, IF ANY.
                    116  *
 0000               117  PREVA  BSS  0
 0000 0102          118         LDA  PALT          ALT JUST FINISHED
                    119  *      RMO  A,E
                    120  *      LRA  1             FORM WORD OFFSET IN NEXT CLUSTER MAT
 0003 1108          121         LDX  NCLMA@        START ADDR OF NEXT CL. MAT.
 0004 C0A0          122         RAD  X,A
 0005 8000          123         STA  NXTCLR
 0006               124  NXTCLR BSS  1
 0007 04FE          125         LDA  *NXTCLR       WORD CONTAINING NEXT CLUSTER FOR ALT
 0008 8101          126         STA  PCLUST        UPDATE NEW CLUSTER
                    127  *
                    128  *              EXIT FROM THIS ABL PROGRAM
                    129  *              IF NEXT CLUSTER 0
                    130  *
 0009 CC80          131         SNZ  A
 000A 78F2          132         BRU  ABLRET
                    133  *
 000B C700          134         RDE  A,A           ADJUSTMENT (1ST CLR IS 1)
 000C 0907          135         LDE  CLSMA@        START. ADR OF CLUSTER MAT
 000D C8C1          136         LLA  1             EACH CLUSTER TAKES 2 WORDS
 000E C090          137         RAD  E,A           ADDR OF NEXT CLUSTER
 000F 8000          138         STA  TPTER
 0010               139  TPTER  BSS  1             POINTER TO NEXT CLUSTER
 0011 04FE          140         DLD  *TPTER        (A),(E) = CLUSTER FLAGS
 0012 A05A          141         DST  CLWD1
                    142  *
                    143  *              FIND OUT ALL POSSIBLE ALTERNATIVES
                    144  *              IN THIS CLUSTER.
                    145  *
 0013 1700          146         LDX  =0            RESET FLAG FOR 16 RH BITS
 0014 905B          147         STX  SECHFT
 0015 170F          148  L050   LDX  =15           BITS 0-15
 0016 9000          149         STX  POSCNT
 0017               150  POSCNT BSS  1             BIT POSITION
                    151  *
 0018 0054          152  NXTBIT LDA  CLWD1         (A) = BITS IN CLUSTER
 0019 1855          153         LDM  TABOB@        TABO INSTRUCTION
 001A C0B2          154         RAD  M,X           SPECIFY BIT# IN TABO
 001B 9001          155         STX  TABONE        TABO (POSCNT)
 001C 7800          156         BRU  $+1
 001D               157  TABONE BSS  1             IS THIS ALT POSSIBLE IN THIS CLUSTER ?
 001E 7813          158         BRU  L100          NO - CHECK NEXT ALT
                    159  *                        YES
                    160  *              CHECK IF PREDICATES MATCH THIS ALTERNATIVE
                    161  *
 001F 10F7          162         LDX  POSCNT        (X)=ALT #
 0020 0109          163         LDA  YNPMA@        START ADR OF Y/N PREDICATE MATRIX
 0021 C0A0          164         RAD  X,A
```

```
 0032  DB20   165          TMBZ  0,SECHFT    ADJUSMENT FOR ALTS 16-31
P 0033  0080
  0034  2710   166          ADD   =16
  0035  804B   167          STA   YNPRP      POINT TO PREDICATE OF THIS ALT
                168 *
  0036  810A   169          LDA   MPRMA0     START ADR OF PRED MASK MATRIX
  0037  C0A0   170          RAD   X,A
  0038  DB20   171          TMBZ  0,SECHFT   ADJUSMENT FOR ALTS 16-31
P 0039  0080
  003A  2710   172          ADD   =16
  003B  8046   173          STA   MPRPT      POINT TO PRED MASK FOR ALT
  003C  8106   174          LDA   SVECTR     PREDICATES
  003D  0C43   175          LDE   *YNPRP     PREDICATES FOR THIS ALT
  003E  C290   176          REO   E,A        COMPARE
  003F  3C42   177          AND   *MPRPT     MASK OUT 'DONT CARE' B
  0040  CC80   178          SNZ   A          PREDICATES MATCHED ?
  0041  780D   179          BRU   L120       YES - ALT FOUND
  0042  48E4   180  L100    DMT   POSCNT     NO  - CHECK NEXT ALT TO THE LH
  0043  7800   181          BRU   $+1
  0044  10E2   182          LDX   POSCNT
  0045  CC22   183          SOO   X          ALL BITS CHECKED ?
  0046  78E1   184          BRU   NXTBIT     NO
  0047  DB20   185          TMBZ  0,SECHFT   YES - ALL 32 BITS CHECKED ?
P 0048  0080
  0049  780B   186          BRU   L110       YES
  004A  0033   187          LDA   CLWD2      NO - CHECK NEXT 16 BITS
  004B  8031   188          STA   CLWD1
  004C  DB70   189          SMBO  0,SECHFT   16 RH BITS BEING CHECKED
P 004D  0080
  004E  78D6   190          BRU   L050       GO CHECK
```

```
                    191          PEJ
                    192  *
                    193  *            ALTERNATIVE FOUND
                    194  *
  0050              195  L120     BSS  0
                    196          IF   DEBUG,1,MSGE3
                    197          LDA  SVECTR          *DEBUGGING
                    198          @BRL BI2HAS          *
                    199          @DST M8+11           *
                    200          LDM  =8              *'STATE VECTOR...'
                    201          @BRL WTC             *
  0050              202  MSGE3   BSS  0
  0050  00D7        203          LDA  POSCNT          CALCULATE ALT#
  0052  DB20        204          TMBZ 0,SECHFT
P 0054  0008
  0055  2710        205          ADD  =16             ADJUSTMENT FOR ALT 16-31
  0056  8102        206          STA  PALT
  0057  7814        207          BRU  ALTEXE          EXECUTE ALTERNATIVE
                    208  *
                    209  *            NO ALTERNATIVE FOUND - OUTPUT ERROR MESSGE
                    210  *
  0058  1F07        211  L110     LDM  =7              'NO ALTERNATIVE FOUND'
  0059  7400        212          @BRL WTC
P 005A  008D
  005B  8106        213          LDA  SVECTR          OUTPUT LAST STATUS OF PRED
  005C  7400        214          @BRL BI2HAS
                    0001
E 005E  0000
  005F  A400        215          @DST M8+11
P 0061  0112
  0062  1F08        216          LDM  =8              'STATE VECTOR : NNNN'
  0063  7400        217          @BRL WTC
P 0065  008D
  0066  8102        218          LDA  PALT            OUTPUT LAST ALT #
  0067  7400        219          BRL  *BI2AS@
                    0000  220  BI2AS@ DATA BI2DAS
E 006A  0000
  006B  8C00        221          @STE M9+16
P 006D  0125
  006E  1F09        222          LDM  =9              'LAST ALT EXECUTED NN'
  006F  7400        223          @BRL WTC
P 0071  008D
  0072  CAE5        224          SVC  5               TERMINATE UTILITY
```

```
                    225          PEJ
                    226  *
                    227  *            EXECUTE ALTERNATIVE
                    228  *
                    229  *            INPUT  : PALT
                    230  *
0069                231  ALTEXE BSS  0
                    232          IF   DEBUG,1,MSGE4
                    233          LDA  PALT          *FOR DEBUGGING*
                    234          @BRL BI2DAS        *              *
                    235          @STE M4+12         *              *
                    236          LDM  =4            *              *
                    237          @BRL WTC           *              *
0069                238  MSGE4   BSS  0
0069 0902           239          LDE  PALT          ALT #
006A 0108           240          LDA  APTR@         START ADR OF ACTION LIST POINTERS
006B C081           241          RAD  A,E
006C 8800           242          STE  TEMPR
006D                243  TEMPR   BSS  1
006E 0CFE           244          LDE  *TEMPR        START ADDR OF ACTION LIST
006F 0905           245          STE  ACLST@
0070 0700           246          LDA  =0            EXECUTE 1ST ACTION IN ALT
0071 0104           247          STA  PACTOF
                    248  *
                    249  *            EXECUTE ACTIONS IN ALTERNATIVE
                    250  *
0072 1104           251  L130    LDX  PACTOF        ACTION OFFSET
0073 0105           252          LDA  ACLST@
0074 8000           253          STA  ACTLST
0075                254  ACTLST  BSS  1             START OF ACTION LIST
                    255  *
0076 06FE           256          LDA  *ACTLST,X     ACTION ADDRESS
0077 CC80           257          SNZ  A             END OF ACTION LIST ?
0078 7C12           258          BRU  *PRDEV@       YES - GO EVALUATE PREDICATES
0079 0812           259          STA  ACTADR        NO  - FOR INDIRECT ADDR.
                    260  *       RIN  X,A           *ORDER OF ACTION WITHIN ALT.
                    261  *       @SRF ABLRF         *FOR DEBUGGING*
                    262  *       @BRL BI2DAS        *
                    263  *       @STE M5+11         *
                    264  *       LDM  =5            *'ENTER ALT #...'
                    265  *       @BRL WTC           *
                    266  *       @LRF ABLRF         *
007A 5104           267          IMO  PACTOF        NO  - BUMP ACTION OFFSET
007B 7410           268          BRL  *ACTADR       EXECUTE ACTION
007C 78F5           269          BRU  L130          GO PREPARE FOR NEXT ACTION
                    270  *
                    271  *
                    272  *            TEMPORARY DATA & POINTERS
                    273  *
                    274  *
007D                275  CLWD1   BSS  1             16 LH BITS OF CLUSTER
007E                276  CLWD2   BSS  1             "  RH    "
007F DB10           277  TABO@   TABO 0             TABO INSTRUCTION
0080 0000           278  SECHFT  DATA 0             BIT0=1 : 16 RH BITS OF CLUSTER PROCESSED
0081                279  YNPRP   BSS  1             POINTER TO PREDICATE FOR ALT
0082                280  MPRPT   BSS  1                  "     "  PRED MASK  "  "
```

```
ABL INTERPRETOR
   0083           281  ABLRF  BSS  8          ABL REGISTER FILE
P  0088  0012     282  PRDEV0 DATA PRDEVA
   008C           283  ACTADR BSS  1          ADDR. OF ACTION SUBROUTINE
```

```
                    284          PEJ
                    285  *
                    286  *       SUBROUTINE TO WRITE A MESSAGE TO CONSOLE
                    287  *
                    288  *       ENTRY: (M): MESSAGE INDEX - ALL REGS PRESERVED
                    289  *
  00BD              29#  WTC     BSS  #
  00BD  D8E#        291          @SRF REGF2         SAVE REG FILE
P 00BE  00A2
  00BF  C7B2        292          REX  M,X
  00#0  C564        293          RMO  B,S           SAVE (B)
                    294  *
  00#1  #218        295          LDA  M@,X         (A)=MESSAGE STRING ADDRESS
  00#2  C5#6        296          RMO  A,B           (B) POINTS TO COUNT + MESSAGE
  00#3  #1##        297          LDA  #,BR         (A)=CHARACTER COUNT
  00#4  C361        298          RIN  B,E           (E)=START OF MESSAGE
  00#5  A##A        299          DST  PWTC+2
                    3##  *
  00#6  C532        3#1          RMO  M,X
                    3#2  *
  00#7  18##        3#3          @LDM =PWTC         CALL SUPERVISOR TO WRITE
P 00#8  00#E
  00#9  CAE#        3#4          SVC  #
  00#A  C546        3#5          RMO  S,B           RESTORE (B)
  00#B  D8A#        3#6          @LRF REGF2         RESTORE REG FILE
P 00#C  00A2
  00#D  C557        3#7          RMO  L,P
                    3#8  *
                    3#9  *       P.R.B
                    31#  *
  00#E  0000        311  PWTC    DATA #
  00#F  0002        312          DATA 2
  00A#              313          BSS  1
  00A1              314          BSS  1
                    315  *
  00A2              316  REGF2   BSS  8               REGISTER FILE
P 00AA  00B5        317  M@      DATA M#,M1,M2,M3,M4,M5,M6,M7,M8,M9,M1#
P 00AB  00B7
P 00AC  00BA
P 00AD  00C2
P 00AE  00CD
P 00AF  00DB
P 00B#  00E8
P 00B1  00FA
P 00B2  #1#7
P 00B3  #115
P 00B4  #127
                    318  *
  00B5  0002        319  M#      DATA 2
  00B6  8A8D        32#          DATA LFCR
  00B7  0004        321  M1      DATA 4
  00B8  A#A#        322          DATA '  ',LFCR
  00B9  8A8D
  00BA  000E        323  M2      DATA 14
  00BB  A#A#        324          DATA '  PROGRAM NN',LFCR
  00BC  D#D2
```

```
  00BD  CFC7
  00BE  D2C1
  00BF  CDA0
  00C0  CECE
  00C1  8A8D
  00C2  0014   325  M3    DATA 20
  00C3  AAAA   326        DATA '***MISSING RULE***',LFCR
  00C4  AACD
  00C5  C9D3
  00C6  D3C9
  00C7  CEC7
  00C8  A0D2
  00C9  D5CC
  00CA  C5AA
  00CB  AAAA
  00CC  8A8D
  00CD  001A   327  M4    DATA 26
  00CE  A0A0   328        DATA '    ENTER ALTERNATIVE NN',LFCR
  00CF  A0A0
  00D0  C5CE
  00D1  D4C5
  00D2  D2A0
  00D3  C1CC
  00D4  D4C5
  00D5  D2CE
  00D6  C1D4
  00D7  C9D6
  00D8  C5A0
  00D9  CECE
  00DA  8A8D
  00DB  0018   329  M5    DATA 24
  00DC  A0A0   330        DATA '      ENTER ACTION  NN',LFCR
  00DD  A0A0
  00DE  A0A0
  00DF  C5CE
  00E0  D4C5
  00E1  D2A0
  00E2  C1C3
  00E3  D4C9
  00E4  CFCE
  00E5  A0A0
  00E6  CECE
  00E7  8A8D
  00E8  0022   331  M6    DATA 34
  00E9  A0A0   332        DATA '    ENTER ACTION NO NN IN ALT NN',LFCR
  00EA  A0A0
  00EB  C5CE
  00EC  D4C5
  00ED  D2A0
  00EE  C1C3
  00EF  D4C9
  00F0  CFCE
  00F1  A0CE
  00F2  CFA0
  00F3  CECE
  00F4  A0C9
```

```
ABL INTERPRETOR
  00F5  CEA0
  00F6  C1CC
  00F7  D4A0
  00F8  CECE
  00F9  8A8D
  00FA  0018    333  M7     DATA 24
  00FB  A0A0    334        DATA '  NO ALTERNATIVE FOUND',LFCR
  00FC  CECF
  00FD  A0C1
  00FE  CCD4
  00FF  C5D2
  0100  CEC1
  0101  D4C9
  0102  D6C5
  0103  A0[illegible]
  0104  CF[illegible]
  0105  CEC4
  0106  8A8D
  0107  001A    335  M8     DATA 26
  0108  A0A0    336        DATA '    STATE VECTOR :  NNNN',LFCR
  0109  A0A0
  010A  D3D4
  010B  C1D4
  010C  C5A0
  010D  D6C5
  010E  C3D4
  010F  CFD2
  0110  A0BA
  0111  A0A0
  0112  CECE
  0113  CECE
  0114  8A8D
  0115  0022    337  M9     DATA 34
  0116  A0A0    338        DATA '  LAST ALTERNATIVE EXECUTED : NN',LFCR
  0117  CCC1
  0118  D3D4
  0119  A0C1
  011A  CCD4
  011B  C5D2
  011C  CEC1
  011D  D4C9
  011E  D6C5
  011F  A0C5
  0120  D8C5
  0121  C3D5
  0122  D4C5
  0123  C4A0
  0124  8AA0
  0125  CECE
  0126  8A8D
  0127  0018    339  M10    DATA 24
  0128  A0A0    340        DATA '  EXIT FROM PROGRAM NN',LFCR
  0129  C5D8
  012A  C9D4
  012B  A0C6
  012C  D2CF
```

```
 #12D  CDA#
 #12E  D#D2
 #12F  CFC7
 #13#  D2C1
 #131  CDA#
 #132  CECE
 #133  8A8D
```

```
                    341          PEJ
                    342  *
 #134               343  ABLPRO  BSS  #
 #134  ###3         344  PROGNO  DATA 3          PROGRAM NO.
                    345  *
 #135  ####         346  PCLUST  DATA #          PRESENT CLUSTER
 #136  ####         347  PALT    DATA #             *    ALT
 #137  ####         348  PACT    DATA #             *    ACTION
 #138  ####         349  PACTOF  DATA #             *    ACTION OFFSET IN ALTERNATIVE
 #139  ####         35#  ACLST@  DATA #          PTR TO ACTION LIST FORALT BEING EXE'D
                    351  *
 #13A  ####         352  SVECTR  DATA #          STATE VECTOR
                    353  *
 #13B  ####         354  CLSMA@  DATA #       POINTER TO CLUSTER MATRIX
 #13C  ####         355  NCLMA@  DATA #          *  *  NEXT-CLUSTER VECTOR
 #13D  ####         356  YNPMA@  DATA #          *  *  PREDICATE MATRIX
 #13E  ####         357  MPRMA@  DATA #          *  *  PRED MASK MATRIX
 #13F  ####         358  APTR#@  DATA #          *  *  ACTION LIST POINTERS
                    359  *
 #14#  ####         36#  BREG    DATA #          (B)=ADDRESS OF PREVIOUS ABSTR. PROGRAM
 #141  ####         361  LREG    DATA #          (L)=RETURN ADDRESS
```

```
                    362          PEJ
                    363  *
                    364  *            5. OUTPUT ALTERNATIVE NO.
                    365  *
  #142              366  OPALT#  BSS  #
  #142  C55#        367          RMO  L,A
  #143  8##8        368          STA  RETN1        SAVE RETURN
  #144  ##F1        369          LDA  PALT         ALT #
  #145  74##        37#          @BRL BI2DAS       CONVERT TO 2 ASCII DIGITS
        ####
E #146  ####
  #147  8C##        371          @STE M4+12        PLACE IN MESSAGE
P #148  ##D9
  #149  1F#4        372          LDM  =4           WRITE MESSGE TO CONSOLE.
  #14A  74#2        373          BRL  *WTC#
  #14B  7C##        374          BRU  *RETN1
                    375  *
  #14C              376  RETN1   BSS  1
P #14D  ##8D        377  WTC#    DATA WTC
                    378  *
                    379  *
                    38#  *            6. OUTPUT ACTION OFFSET WITHIN ALT
                    381  *
  #14E              382  OPACT   BSS  #
  #14E  C55#        383          RMO  L,A
  #14F  8##D        384          STA  RETN2        RETURN ADDR
  #15#  ##E7        385          LDA  PACTOF       ACTION OFFSET WITHIN ALTERNATIVE
  #151  74##        386          @BRL BI2DAS
        ####
E #152  ####
  #153  8C##        387          @STE M6+11        PLACE IN MESSAGE
P #154  ##F3
  #155  ##E#        388          LDA  PALT         ALT ##
  #156  74##        389          @BRL BI2DAS       CONVERT TO 2 ASCII DIGITS
        ####
E #157  ####
  #158  8C##        39#          @STE M6+16
P #159  ##F8
  #15A  1E#6        391          LDM  =6
  #15B  74F1        392          BRL  *WTC#        OUTPUT MESSGE TO CONSOLE
  #15C  7C##        393          BRU  *RETN2
                    394  *
  #15D              395  RETN2   BSS  1
                    396  *
                    397  *       1#. OUTPUT PROGRAM # TO CONSOLE
  #15E  C55#        398  OPPRO#  RMO  L,A
  #15F  8##8        399          STA  RETURN
  #16#  ##D3        4##          LDA  PROGNO
  #161  74##        4#1          @BRL BI2DAS
        ####
E #162  ####
  #163  8C##        4#2          @STE M2+6
P #164  ##C#
  #165  1F#2        4#3          LDM  =2
  #166  74E6        4#4          BRL  *WTC#
  #167  7C##        4#5          BRU  *RETURN
```

```
 0168          486  RETURN BSS  1
               487  *
               488       END
```

```
     A       0000    ABLENT  0000    ABLPRO  0134    ABLRET  000D  R ABLRF   0083
     ACLST@  0139    ACTADR  008C    ACTLST  0075    ALTEXE  0069    APTR#@  013F
     B       0006    BI2AS@  0062    BI2DAS  0000    BI2HAS  0001  R BI20AS  0002
     BR      0001    BREG    014#    CLSMA@  013B    CLWD1   007D    CLWD2   007E
     DEBUG   0001    E       0001  R HAS2BI  0003    L       0005    L05#    0025
     L1##    0042    L11#    0055    L12#    004F    L13#    0072    LFCR    0A0D
     LREG    014I    M       0003    M#      00B5    M1      00B7    M1#     0127
     M2      00BA    M3      00C2    M4      00CD    M5      00DB    M6      00E8
     M7      00FA    M8      01#7    M9      0115    M@      00AA    MPRMA@  013E
     MPRPT   0082    MSGE1   0007    MSGE2   000D    MSGE3   004F    MSGE4   0069
     NCLMA@  013C    NXTBIT  0028    NXTCLR  0016  R OPACT   014E  R OPALT@  0142
   R OPPRO@  015E    P       0007  R PACT    0137    PACTOF  0138    PALT    0136
     PCLUST  0135    POSCNT  0027    PRDEV@  008B    PRDEVA  0012    PROGNO  0134
     PWTC    009E    REGF2   00A2    RETN1   014C    RETN2   015D    RETURN  0168
     S       0004    SECHFT  0080    SVC     CAE#    SVECTR  013A    TABOB@  007F
     TABONE  002D    TEMPR   006D    TPTER   0020    WTC     008D    WTC@    014D
     X       0002  R XB      0003    YNPMA@  013D    YNPRP   0081

     NO ERRORS       NO WARNINGS
T40#6S    19/03/84      16:07:33
```

APPENDIX 3

A3. Listings of utility FUP/G :

- V1# : main module
- V3# : syntax check
- V4# : disassembler - part 1
- V9# : disassembler - part 2
- VD# : common subroutines

```
V1#15S    19/#3/84    16:14:36

  99#A OFF LINE ASSEMBLER                                        SHEET    1
  FUP/G
                    1  *          FUP/G  (GRAPHIC 7 FOREGROUND UTILITY PROGRAM)
                    2  *          TI99#A
                    3  *
                    4  *          K. VO
                    5  *
                    6  *          ISSUE #
                    7  *          REVISION #
                    8  *
                    9          IDT  V1#
                   1#          HED  FUP/G
                   11  *
                   12  *
                   13  *          FUP/G IS USED TO DEBUG GRAPHIC 7 DISPLAY FILE.
                   14  *          IT ALLOWS THE USER TO EXAMINE AND MODIFY CONTENTS
                   15  *          OF GRAPHIC 7 DISPLAY MEMORY, TO GET A HARD COPY
                   16  *          ON LINE PRINTER OF DISPLAY MEMORY, LINK/UNLINK
                   17  *          SOME PORTION OF DISPLAY MEMORY...
                   18  *
                   19          REF  A3,A5,A6,INIT,UNPACK,MRKIP
                   2#          REF  ABLENT,ABLRET,BI2DAS,BI2HAS,BI2OAS,WTC,PAGENO
                   21  *
                   22          DEF  MPRED,CMDLN&,CMDLNS,ASSTR1,ASSTR2,ASSTR3,ASSTR4,ASSTR5
                   23          DEF  ASSTR6,ASSTR7,IPVAL2,IPVAL3,IPVAL4,IPVAL5
                   24          DEF  IPVAL6,IPVAL7,NOWORD,NOWDP,NOSTRS
                   25          DEF  GBUFF,GBUFF@,SADDR,EADDR,NWMOD,HBUF1,HBUF2
                   26          DEF  BHBUF2,GI,RPRB
```

```
                    27                PEJ
                    28  *
                    29  *
        0000        30  A       EQU   0
        0001        31  B       EQU   1
        0002        32  X       EQU   2
        0003        33  M       EQU   3
        0004        34  S       EQU   4
        0005        35  L       EQU   5
        0006        36  R       EQU   6
        0007        37  F       EQU   7
                    38  *
        0000        39  P0      EQU   0
        0001        40  P1      EQU   1
        0002        41  P2      EQU   2
        0003        42  P3      EQU   3
        0004        43  P4      EQU   4
        0005        44  P5      EQU   5
        0006        45  P6      EQU   6
        0007        46  P7      EQU   7
        0008        47  P8      EQU   8
        0009        48  P9      EQU   9
        000A        49  P10     EQU   10
        000B        50  P11     EQU   11
        000C        51  P12     EQU   12
        000D        52  P13     EQU   13
        000E        53  P14     EQU   14
        000F        54  P15     EQU   15
                    55  *
        8D00        56  CR      EQU   >8D00
        8000        57  FULL    EQU   >8000
        4000        58  EMPTY   EQU   >4000
                    59  *
        0001        60  BR      EQU   1
        0003        61  XB      EQU   3
        CAE0        62  SVC     OPD   >CAE0,3
        8A8D        63  LFCR    EQU   >8A8D
                    64  *
```

```
990A OFF LINE ASSEMBLER                                    SHEET   3
FUP/G
                        65          PEJ
P       0006            66          BRS   PROGNO
                        67  *
                        68  *
  0000  0000            69          @LDA =PROGNO
P 0001  0006
  0002  C503            70          RMO   A,H          M REG CONTAINS ABL PROGRAM ADDR.
  0003  7400            71          @BRL  ABLENT       CALL DRIVER TO EXECUTE FUP/G PROGRAM
        0006
E 0004  0000
  0005  CAE5            72          SVC   5            * TERMINATE FUP/G
```

```
                PEJ
*
C1    V                                     INIT
C2       V                                  READ & CHECK COMMAND
C3          V V V V V V                     PROCESS COMMAND
*
N     2 3 2 2 2 2 # 2
*
P#    - - N N N N N -                       .
P1    - - N N N Y Y -                       . COMMAND NO.
P2    - - N Y Y N N -                       .
P3    - - Y N Y N Y -                       .
P4    - - N N N N N Y                       INVALID SYNTAX/PARAMETER
*
      # 1 2 3 4 5 6 7
*                                           11  13  15
*
*
ALT # : INITIALIZATION
ALT 1 : CLEAR PRED. - READ VALID COMMAND
ALT 2 : 'L'
ALT 3 : 'H'
ALT 4 : 'M'
ALT 5 : 'W'
ALT 6 : 'Q'
ALT 7 : INVALID SYTAX/PARAMETER(S)
```

```
                    99          PEJ
                   100  *
                   101  *              ABL PROGRAM SECTION
                   102  *              ===================
                   103  *
                   104  *
    0006  0003     105  PROGNO DATA 3                    PROGRAM NO.
                   106  *
    0007  0000     107  PCLUST DATA 0                    PRESENT CLUSTER
    0008  0000     108  PALT   DATA 0                       "   ALT
    0009  0000     109  PACT   DATA 0                       "   ACTION
    000A  0000     110  PACTOF DATA 0                       "   ACTION OFFSET IN ALTERNATIVE
    000B  0000     111  ACLST@ DATA 0                    PTR TO ACTION LIST FORALT BEING EXE'D
                   112  *
    000C  0000     113  SVECTR DATA 0                    STATE VECTOR
                   114  *
P   000D  0014     115  CLSMA@ DATA CLSMAT          POINTER TO CLUSTER MATRIX
P   000E  001A     116  NCLMA@ DATA NCLMAT              "   "  NEXT-CLUSTER VECTOR
P   000F  0022     117  YNPMA@ DATA YNPMAT              "   "  PREDICATE MATRIX
P   0010  002A     118  MPRMA@ DATA MPRMAT              "   "  PRED MASK MATRIX
P   0011  0032     119  APTR00 DATA APTR00              "   "  ACTION LIST POINTERS
                   120  *
    0012  0000     121  BREG   DATA 0                    (B)=ADDRESS OF PREVIOUS ABSTR. PROGRAM
    0013  0000     122  LREG   DATA 0                    (L)=RETURN ADDRESS
```

```
                    123           PEJ
                    124  *
                    125  *             CLUSTER MATRIX - BITS0-31
                    126  *
   0014  8000       127  CLSMAT DATA >8000          C1
   0015  0000       128         DATA >0000
   0016  4000       129         DATA >4000          C2
   0017  0000       130         DATA >0000
   0018  3F00       131         DATA >3F00          C3
   0019  0000       132         DATA >0000
                    133  *
                    134  *             NEXT CLUSTER -
                    135  *
   001A  0002       136  NCLMAT DATA 2              0
   001B  0003       137         DATA 3              1
   001C  0002       138         DATA 2              2
   001D  0002       139         DATA 2              3
   001E  0002       140         DATA 2              4
   001F  0002       141         DATA 2              5
   0020  0000       142         DATA 0              6
   0021  0002       143         DATA 2              7
                    144  *
                    145  *             Y/N PREDICATE MATRIX
                    146  *
   0022  0000       147  YNPMAT DATA >0000          ALT 0
   0023  0000       148         DATA >0000              1
   0024  1000       149         DATA >1000              2
   0025  2000       150         DATA >2000              3
   0026  3000       151         DATA >3000              4
   0027  4000       152         DATA >4000              5
   0028  5000       153         DATA >5000              6
   0029  0800       154         DATA >0800              7
                    155  *
                    156  *             MASK FOR PREDICATE MATRIX
                    157  *
   002A  0000       158  MPRMAT DATA >0000          ALT 0
   002B  0000       159         DATA >0000              1
   002C  F800       160         DATA >F800              2
   002D  F800       161         DATA >F800              3
   002E  F800       162         DATA >F800              4
   002F  F800       163         DATA >F800              5
   0030  F800       164         DATA >F800              6
   0031  0800       165         DATA >0800              7
                    166  *
                    167  *             ACTION MATRIX
                    168  *
P  0032  003A       169  APTR00 DATA ACLST0       POINTERS TO ALT ACTION LISTS
P  0033  003C       170         DATA ACLST1
P  0034  003F       171         DATA ACLST2
P  0035  0042       172         DATA ACLST3
P  0036  0045       173         DATA ACLST4
P  0037  0047       174         DATA ACLST5
P  0038  0049       175         DATA ACLST6
P  0039  004B       176         DATA ACLST7
```

```
                    177          PEJ
                    178  *
  0000              179  ACLST0 BSS  0
                    180  *            - ALT 0 : INITIALIZATION
P 0000  0081        181          DATA INITI         INITIALIZATION
  0002  0000        182          DATA 0
                    183  *
  0004              184  ACLST1 BSS  0
                    185  *            - ALT 1 : READ AND CHECK COMMAND
                    186  *
P 0004  008C        187          DATA RDCMD         READ COMMAND & SET PREDICATES
        0000        188          DATA A3            CHECK COMMAND SYNTAX
E 0006  0000
  0008  0000        189          DATA 0
                    190  *
  000A              191  ACLST2 BSS  0
                    192  *            - ALT 2 : 'L'
P 000A  010C        193          DATA GRTI1         READ GR7 INST. TO TI BUFFER #1
        0001        194          DATA A5            DISASSEMBLE GR7 INST. TO CONSOLE
E 000C  0000
  000E  0000        195          DATA 0
                    196  *
  0010              197  ACLST3 BSS  0
                    198  *            - ALT 3 : 'H'
P 0010  010F        199          DATA GRTI2         READ 34 WORDS TO TI BUFFER
        0002        200          DATA A6            DISASSEMBLE GR7 INST. TO LP
E 0012  0000
  0014  0000        201          DATA 0
                    202  *
  0016              203  ACLST4 BSS  0
                    204  *            - ALT 4 : 'M'
P 0016  0119        205          DATA MODD          MODIFY CONTENTS IN BUFFER
  0018  0000        206          DATA 0
                    207  *
  001A              208  ACLST5 BSS  0
                    209  *            - ALT 5 : 'W'
P 001A  012E        210          DATA WBTGR7        WRITE BUFFER TO GR7 MEMORY
  001C  0000        211          DATA 0
                    212  *
  001E              213  ACLST6 BSS  0
                    214  *            - ALT 6 : 'Q'
P 001E  0148        215          DATA TERMES        OUTPUT TERMINATION MESSAGE
  0020  0000        216          DATA 0
                    217  *
  0022              218  ACLST7 BSS  0
                    219  *            - ALT 7 : INVALID COMMAND
P 0022  0141        220          DATA ERMESG        WRITE ERROR MEESAGE TO CONSOLE
  0024  0000        221          DATA 0
```

```
                    222          PEJ
                    223  *
                    224  *            DATA SECTION
                    225  *            ============
                    226  *
          0003      227  INIT@  DATA INIT
E 004D    0000
P         000C      228  MPRED  EQU  SVECTR
                    229  *
                    230  *
P 004E    004F      231  CMDLN%  DATA CMDLN&
  004F    4050      232  CMDLN&  DATA EMPTY+80      80 CHARS
  0050    0000      233          DATA 0,0
  0051    0000
  0052              234  CMDLNS  BSS  80/2
  007A    0A0D      235          DATA LFCR
                    236  *
  007B              237  ASSTR@  BSS  1             POINTER TO ONE OF THESE STRINGS
  007C    4001      238          DATA EMPTY+1       ASCII STRING 1
  007D    0000      239          DATA 0,0
  007E    0000
  007F              240  ASSTR1  BSS  1
  0080    4006      241          DATA EMPTY+6                  2
  0081    0000      242          DATA 0,0
  0082    0000
  0083              243  ASSTR2  BSS  3
  0086    4006      244          DATA EMPTY+6                  3
  0087    0000      245          DATA 0,0
  0088    0000
  0089              246  ASSTR3  BSS  3
  008C    4006      247          DATA EMPTY+6                  4
  008D    0000      248          DATA 0,0
  008E    0000
  008F              249  ASSTR4  BSS  3
  0092    4006      250          DATA EMPTY+6                  5
  0093    0000      251          DATA 0,0
  0094    0000
  0095              252  ASSTR5  BSS  3
  0098    4006      253          DATA EMPTY+6                  6
  0099    0000      254          DATA 0,0
  009A    0000
  009B              255  ASSTR6  BSS  3
  009E    4006      256          DATA EMPTY+6                  7
  009F    0000      257          DATA 0,0
  00A0    0000
  00A1              258  ASSTR7  BSS  6/2
  00A4              259  NOSTRS  BSS  1             # OF ASCII STRINGS IN CMD LINE
                    260  *
  00A5              261  IPVAL2  BSS  1                        2
  00A6              262  IPVAL3  BSS  1                        3
  00A7              263  IPVAL4  BSS  1                        4
  00A8              264  IPVAL5  BSS  1                        5
  00A9              265  IPVAL6  BSS  1                        6
  00AA              266  IPVAL7  BSS  1                        7
                    267  *
  00AB              268  PRECMD  BSS  1             PREVIOUS COMMAND IN LH BYTE, RH BYTE ZERO
```

```
  ØØAC          269  NOWORD BSS  1          # OF WORD IN GR7 MEM TO BE PROCESSED
  ØØAD          27Ø  NOWDP  BSS  1          # WORDS ALREADY PROCESSED
  ØØAE          271  SADDR  BSS  1          START BYTE ADDR OF BUFFER
  ØØAF          272  EADDR  BSS  1           END
  ØØBØ          273  NWMOD  BSS  1          # OF WORDS TO MODIFY
```

```
SUP/G
                    274          PEJ
                    275  *
                    276  *            ACTION SECTION
                    277  *            **************
                    278  *
                    279  *       --   INITIALIZATION
                    28#  *
  ##B1  C55#        281  INITI   RMO  L,A          SAVE RETURN ADDR
  ##B2  8###        282          STA  A1RETN
  ##B3  1F##        283          LDM  =#           SKIP A LINE
  ##B4  74##        284          BRL  *WTC@@
        ####B       285  WTC@@   DATA WTC
E ##B5  ####
  ##B6  1F#A        286          LDM  =1#          LET MFUP ANNOUNCE ITSELF
  ##B7  74FD        287          BRL  *WTC@@
  ##B8  1F##        288          LDM  =#           SKIP A LINE
  ##B9  74FB        289          BRL  *WTC@@
  ##BA  7C##        29#          BRU  *A1RETN
                    291  *
  ##BB              292  A1RETN  BSS  1
                    293  *
```

```
                    294          PEJ
                    295  *       --    - INITIALIZE ALL INPUT STRINGS
                    296  *             - READ COMMAND LINE FROM CONSOLE
                    297  *             - RECOGNIZE VALID COMMAND AND SET ITS PRED..
                    298  *             UNPACK THE COMMAND LINE INTO SETS OF STRINGS
                    299  *
                    300  *
  00BC-             301  RDCMD  BSS   0
  00BC  07A0        302          LDA   =0          CLEAR PREDICATES
  00BD  8036        303          STA   SVECTR
  00BE  C550        304          RMO   L,A         SAVE RETURN
  00BF  8036        305          STA   A2RETN
                    306  *
  00C0  0700        307          LDA   =0          INIT
  00C1  80EA        308          STA   NOWORD
  00C2  80EA        309          STA   NOWDP
  00C3  1F00        310  A2.0    LDM   =0
  00C4  74F0        311          BRL   *WTC@@      NEW LINE
  00C5  1F0B        312          LDM   =11
  00C6  74EE        313          BRL   *WTC@@      PROMPT FOR COMMAND
  00C7  7400        314          BRL   *RFC@       READ COMMAND LINE TO BUFFER
P 00C8  0151        315  RFC@    DATA  RFC
  00C9  CCA3        316          SNO   M           VALID RESPONSE ?
  00CA  7823        317          BRU   A2.12       NO - ERROR MESSGE
                    318  *
  00CB  CC03        319          SZE   M           JUST A CR ENTERED ?
  00CC  7805        320          BRU   A2.2        NO  - GO UNPACK CMDLN INTO STRINGS
  00CD  00DD        321          LDA   PRECMD      YES - RECALL PREVIOUS CMD
  00CE  2000        322          @ADD  =>00A0      PLACE BLANK IN RH BYTE
  00CF  00A0
  00D0  8081        323          STA   CMDLNS      AND FAKE THE INPUT COMMAND
  00D1  7813        324          BRU   A200        GO REPEAT THE PREVIOUS COMMAND
                    325  *
  00D2  1800        326  A2.2    @LDM  =CMDLN&     UNPACK INTO STRINGS OF ASCII
P 00D3  004F
  00D4  7422        327          BRL   *UNPCK@
  00D5  0003        328          DATA  3           # OF DELIMITERS
  00D6  A0FF        329          DATA  ' '
  00D7  8D00        330          DATA  CR
  00D8  ACFF        331          DATA  ','
  00D9  0007        332          DATA  7           # OF DESTINATION STRINGS
P 00DA  007C        333          DATA  ASSTR1-3
P 00DB  0080        334          DATA  ASSTR2-3
P 00DC  0086        335          DATA  ASSTR3-3
P 00DD  008C        336          DATA  ASSTR4-3
P 00DE  0092        337          DATA  ASSTR5-3
P 00DF  0098        338          DATA  ASSTR6-3
P 00E0  009E        339          DATA  ASSTR7-3
                    340  *
  00E1  CCA2        341          SNO   X           IS THERE OVERFLOW IN ANY STRING ?
  00E2  7808        342          BRU   A2.12       YES - UNVALID COMMAND
  00E3  C722        343          RDE   X,X         *UNPACK NOT WORKING AS CLAIMED
  00E4  900F        344          STX   NOSTRS      NO - SAVE # OF STRINGS
  00E5  17FB        345  A200    LDX   =NOCOMM     # OF COMMANDS
  00E6  0098        346  A210    LDA   ASSTR1      (A) LH BYTE = COMMAND
  00E7  A01F        347          LDE   COMMD,X     LOOK UP COMMAND REPERTOIRE
```

```
FUP/C
   00E8  C290    348          REQ  E,A
   00E9  3800    349          @AND =>7F00       ONLY LH BYTE
   00EA  7F00
   00EB  CC80    350          SNZ  A            INSTRUCTION MATCHED ?
   00EC  7805    351          BRU  A215         YES
   00ED  40F8    352          BIX  A210         NO - TRY NEXT INSTRUCTION
                 353  *
   00EE  1F0C    354  A2.12    LDM  =12          'UNVALID COMMAND'
   00EF  7400    355          BRL  *$+1
         0008    356          DATA WTC
E  00F0  0000
   00F1  7801    357          BRU  A2.0         PROMPT FOR NEW COMMAND
                 358  *
   00F2  8080    359  A215    STA  PRECMD       SAVE THE COMMAND FOR POSSIBLE CR NEXT TIME
   00F3  0210    360          LDA  PREDIC,X     GET PREDICATE CORRESPONDING TO THIS COMMAND
   00F4  8106    361          STA  SVECTR
   00F5  7C00    362          BRU  *A2RETN      RETURN
                 363  *
   00F6          364  A2RETN  BSS  1
         0004    365  UNPCK0  DATA UNPACK
E  00F7  0000
   00F8          366          BSS  10
```

```
                    367          PEJ
                    368  *
                    369  *       COMMAND LIST
                    370  *
 0102  CCFF         371  BCOMMD  DATA 'L'          LIST
 0103  C8FF         372          DATA 'H'          HARD COPY
 0104  CDFF         373          DATA 'M'          MODIFY
 0105  D7FF         374          DATA 'W'          WRITE
 0106  D1FF         375          DATA 'Q'          QUIT
 0107               376  COMMD   BSS  0
       FFFB         377  NOCOMM  EQU  BCOMMD-COMMD # OF COMMANDS
                    378  *
                    379  *       PREDICATES FOR COMMANDS
                    380  *
 0107  1000         381          DATA >1000        L
 0108  2000         382          DATA >2000        H
 0109  3000         383          DATA >3000        M
 010A  4000         384          DATA >4000        W
 010B  5000         385          DATA >5000        Q
 010C               386  PREDIC  BSS  0
```

```
                    387          PEJ
                    388  *
                    389  *       --    READ GR7 MEMORY TO TI MEMORY
                    39#  *
                    391  *             INPUT  : IPVAL2,NOWORD
                    392  *             OUTPUT : GBUFF
                    393  *
  #1#C              394  GRTI1   BSS   #
  #1#C  ##98        395          LDA   IPVAL2        STARTING BYTE ADDR
  #1#D  #89E        396          LDE   NOWORD        # OF WORDS
  #1#E  78#1        397          BRU   GRTI
                    398  *
                    399  *       READ 38 WORDS FROM GR7 TO TI (FOR 'H')
                    4##  *
  #1#F              4#1  GRTI2   BSS   #
  #1#F  ##95        4#2          LDA   IPVAL2        START BYTE ADDR
  #11#  84##        4#3  GRTI    @STA  GI+1          FOR 'GI' MESSAGE
P #111  #18C
  #112  18##        4#4          @LDM  =RPRB         RECEIVE 'RI' FROM GRB
P #113  #185
  #114  CAE#        4#5          SVC   #
  #115  54##        4#6          @IMO  PAGENO        BUMP PAGE NO FOR 'H'
        ###C
E #116  ####
  #117  C557        4#7          RMO   L,P           RETURN
                    4#8  *
P #118  #194        4#9  GBUFF@  DATA  GBUFF         POINTER
```

```
                    410          PEJ
                    411  *
                    412  *       --   MODIFY GBUFF WITH THE CONTENTS SPECIFIED
                    413  *
  0119              414  MODD    BSS  0
  0119  C562        415          RMO  B,X          BACK-UP (B)
  011A  0008A       416          LDA  IPVAL2
  011B  2892        417          SUB  SADDR
  011C  C841        418          LRA  1            OFFSET IN WORDS FR. START OF GBUFF
  011D  2000        419          @ADD =GBUFF       POINT TO LOC. IN BUFFER
P 011E  0194
  011F  C501        420          RMO  A,E
  0120  0000        421          @LDA =IPVAL3      START ADDR OF INPUT
P 0121  00A6
  0122  C506        422          RMO  A,B
                    423  *
  0123  8100        424  UPDATE  LDA  0,BR         NEW CONTENT
  0124  C796        425          REX  E,B
  0125  8100        426          STA  0,BR         UPDATE INTO BUFFER
  0126  C796        427          REX  E,B
  0127  C311        428          RIN  E,E          POINT TO NEXT WORD
  0128  C366        429          RIN  B,B
  0129  4C03        430          DMT  *NVMOD@      UPDATE ALL SPECIFIED LOCATIONS
  012A  78F8        431          BRU  UPDATE
  012B  C526        432          RMO  X,B          RESTORE (B)
  012C  C557        433          RMO  L,P
                    434  *
                    435  *
P 012D  0008        436  NVMOD@  DATA NVMOD
```

```
                    437          PEJ
                    438  *
                    439  *       WRITE BUFFER TO GR7 MEMORY
                    440  *
  #12E              441  W8TGR7  BSS  #
  #12E  8###        442          LDA  8+1          'SU'
  #12F  5355        443          DATA >5355
  #13#  84##        444          @STA SUSU
P #131  #191
  #133  84##        446          @STA S8SU+1
P #134  #192
  #135  #1A9        447          LDA  EADDR        NUMBER..
  #136  29A8        448          SUB  SADDR
  #137  C841        449          LRA  1                 .. OF WORDS TO XFER
                    45#  *
  #138  8#5A        451          STA  $USU+2
  #139  C8C1        452          LCA  1
  #13A  27#6        453          ADD  =6
  #13B  84##        454          @STA SUPRB+2      CHAR COUNT
P #13C  #183
  #13D  18##        455          @LDM =SUPRB       WRITE TO GR7 MEMORY
P #13E  #181
  #13F  CAE#        456          SVC  #
  #14#  C557        457          RMO  L,P
```

```
                    458          PEJ
                    459  *
                    460  *       --   OUTPUT 'INVALID SYNTAX' TO CONSOLE
                    461  *
  0141              462  ERMESG  BSS  0
  0141  C550       463          RMO  L,A          SAVE RETURN
  0142  0004       464          STA  A16RET
                    465  *
  0143  1F0F       466          LDM  =15
  0144  7400       467          @BRL WTC
        0008
E 0145  0000
  0146  7C00       468          BRU  *A16RET
                    469  *
  0147              470  A16RET BSS  1            RETURN
                    471  *
                    472  *
                    473  *       --   OUTPUT 'FUP/G TERMIANTES' TO CONSOLE
                    474  *
  0148  C550       475  TERMES  RMO  L,A          SAVE RETURN
  0149  0006       476          STA  TERRTN
                    477  *
  014A  1F00       478          LDM  =0           SKIP A LINE
  014B  7402       479          BRL  *WTCO
  014C  1F10       480          LDM  =16          'FUP/G TERMINATES'
  014D  7400       481          BRL  *WTCO
        0008       482  WTCO    DATA WTC
E 014E  0000
  014F  7C00       483          BRU  *TERRTN      RETURN
                    484  *
  0150              485  TERRTN BSS  1            RETURN
```

```
                    486          PEJ
                    487  *
                    488  *       SUBROUTINE RFC (READ FROM CONSOLE) SETS UP A PRB FOR READING
                    489  *       FROM LUN # AND THEN ISSUES THE READ
                    490  *
                    491  *       ON RETURN
                    492  *         (M)=NO. OF CHARS. IN I/P STRING EXCLUSIVE OF THE CR
                    493  *             =-1 FOR A READ ERROR OR A LINE TOO LONG
                    494  *           ALL OTHER REGISTERS ARE PRESERVED
                    495  *
  0151  0020        496  RFC     STA  R9+4          SAVE ENTRY (A),(L),(X)
  0152  C560        497          RMO  L,A
  0153  0028        498          STA  R9+5
  0154  9028        499          STX  R9+6
                    500  *
  0155  0750        501          LDA  =80           RESET THE I/P CHAR. COUNT
  0156  0025        502          STA  R9+2
                    503  *
  0157  1800        504          @LDM =R9           ISSUE THE READ
P 0158  017A
  0159  CAE0        505          SVC  0
  015A  DB21        506          TMBZ 1,R9          READ ERROR?
P 015B  017A
  015C  780B        507          BRU  R4            YES:
                    508  *
  015D  001E        509          LDA  R9+2          (A)=I/P CHAR. COUNT
  015E  6F50        510          CPA  =80
  015F  C703        511          RDE  A,M
  0160  CD20        512          SEQ                80 CHARS. READ IN?
  0161  780F        513          BRU  R7            NO:
                    514  *
  0162  0400        515          @LDA CMDLNS+39
P 0163  0079
  0164  678D        516          CPL  =>8D
  0165  CD20        517          SEQ                WAS 80-TH CHAR. A CR?
  0166  7803        518          BRU  R6            NO:
  0167  7809        519          BRU  R7
                    520  *
  0168  070F        521  R4      LDA  =15
  0169  7801        522          BRU  $+2
  016A  0711        523  R6      LDA  =17
                    524  *
  016B  1F00        525          LDM  =0,
  016C  7400        526          BRL  *WTC000
        000B        527  WTC000  DATA WTC
E 016D  0000
  016E  C503        528          RMO  A,M
  016F  74FD        529          BRL  *WTC000
  0170  1FFF        530          LDM  =-1           SET BAD LINE FLAG
                    531  *
  0171  C330        532  R7      RIN  M,A           (A)=NO. OF CHARS READ
  0172  1C00        533          @LDM CMDLNX
P 0173  004E
  0174  7400        534          BRL  *MRKIP0
        0005        535  MRKIP0  DATA MRKIP
E 0175  0000
```

```
FWP/G
  #176  C7#3   536          RDE   A,M
  #177  ##86   537          LDA   R9+4        RESTORE ENTRY (A),(X)
  #178  1##7   538          LDX   R9+6
  #179  7C#5   539          BRU   *R9+5
               54#  *
  #17A  #4##   541  R9       DATA  >4##        # LUN #
  #17B  ####   542          DATA  #           1 READ ASCII OPCODE
  #17C         543          BSS   1           2 CHAR. COUNT
P #17D  ##52   544          DATA  CMDLNS      3 DATA BUFFER ADDR.
  #17E         545          BSS   3           4-5 ENTRY (A),(L),(X)
```

```
                    546          PEJ
                    547  *
                    548  *       GR7 MESSAGES TO READ MEMORY
                    549  *
                    550  *
  0181  0002        551  SUPRB   DATA 2              GR8
  0182  0003        552          DATA 3              WRITE OBJECT
  0183              553          BSS  1              CHAR COUNT
P 0184  0191        554          DATA SUSU           BUFFER
  0185  0002        555  RPRB    DATA 2              GR8
  0186  0001        556          DATA 1              READ OBJECT
  0187  0006        557          DATA 6              CHAR COUNT FOR 'GI'
P 0188  0188        558          DATA GI             BUFFER ADDR
  0189  0050        559          DATA 12+38*2        CHAR COUNT FOR 'RI'
P 018A  018E        560          DATA RI             BUFFER
  018B  4749        561  GI      DATA >4749          'GIVE IMAGE'
  018C              562          BSS  1              STARTING BYTE ADDR
  018D  0026        563          DATA 38             WORD COUNT
  018E  5249        564  RI      DATA >5249          'RETURN IMAGE'
  018F              565          BSS  1              STARTING BYTE ADDR
  0190              566          BSS  1              # OF WORDS
  0191  0000        567  SUSU    DATA 0              NOT USED
  0192  564C        568  VL      DATA >564C          'VARIABLE LENGTH'
  0193              569          BSS  1              # OF WORDS FOLLOWING
  0194              570  GBUFF   BSS  0              GRAPHIC BUFFER - TO STORE GR7 INST.
  0194              571  HBUF1   BSS  0              'H' SHARES THE SAME BUFFER AS 'L'
  0194              572          BSS  38
  018A              573  HBUF2   BSS  38
P 01E0  0000        574  BHBUF2  BYTE HBUF2          BYTE ADDR OF HBUF2
  01E1  0374
                    575  *
                    576          END
```

FUP/G

	A	0000		A16RET	0147		A1RETN	00BB		A2.0	00C3		A2.12	00EE
	A2.2	00D2		A2#8	00E5		A21#	00E6		A215	00F2		A2RETN	00F6
	A3	0000		A5	0001		A6	0002		ABLENT	0006	R	ABLRET	0007
	ACLST#	003A		ACLST1	003C		ACLST2	003F		ACLST3	0042		ACLST4	0045
	ACLST5	0047		ACLST6	0049		ACLST7	004B	R	ACLST@	000B		APTR00	0032
R	APTR#@	0011		ASSTR1	007F		ASSTR2	0083		ASSTR3	0089		ASSTR4	008F
	ASSTR5	0095		ASSTR6	009B		ASSTR7	00A1	R	ASSTR@	007B		B	0006
	BCOMMD	01#2		BHBUF2	01E#	R	BI2DAS	0008	R	BI2HAS	0009	R	BI2OAS	000A
	BR	0001	R	BREG	0012	R	CLSMA@	000D		CLSMAT	0014		CMDLNX	004E
	CMDLN&	004F		CMDLNS	0052		COMMD	01#7		CR	8D00		E	0001
	EADDR	00AF		EMPTY	4000		ERMESG	0141	R	FULL	8000		GBUFF	0194
	GBUFF@	0118		GI	018B		GRTI	011#		GRTI1	01#C		GRTI2	01#F
	HBUF1	0194		HBUF2	01BA		INIT	0003	R	INIT@	004D		INITI	00B1
	IPVAL2	00A5		IPVAL3	00A6		IPVAL4	00A7		IPVAL5	00A8		IPVAL6	00A9
	IPVAL7	00AA		L	0005		LFCR	8A8D	R	LREG	0013		M	0003
	MODD	0119		MPRED	000C	R	MPRMA@	0010		MPRMAT	002A		MRKIP	0005
	MRKIP@	0175	R	NCLMA@	000E		NCLMAT	001A		NOCOMM	FFFB		NOSTRS	00A4
	NOWDP	00AD		NOWORD	00AC		NWMOD	00B0		NWMOD@	012D		P	0007
R	P0	0000	R	P1	0001	R	P1#	000A	R	P11	000B	R	P12	000C
R	P13	000D	R	P14	000E	R	P15	000F	R	P2	0002	R	P3	0000
R	P4	0004	R	P5	0005	R	P6	0006	R	P7	0007	R	P8	0008
R	P9	0009	R	PACT	0009	R	PACTOF	000A		PAGENO	000C	R	PALT	0008
R	PCLUST	0007		PRECMD	00AB		PREDIC	01#C		PROGNO	0006		R4	0168
	R6	016A		R7	0171		R9	017A		RDCMD	00BC		RFC	0151
	RFC@	00C8		RI	018E		RPRB	0185	R	S	0004		SADDR	00AE
	SUPRB	0181		SUSU	0191		SVC	CAE0		SVECTR	000C		TERMES	0148
	TERRTN	015#		UNPACK	0004		UNPCK@	00F7		UPDATE	0123	R	VL	0192
	WBTGR7	012E		WTC	0008		WTC@@	00B5		WTC@@@	016D		WTCO	014E
	X	0002	R	XB	0003	R	YNPMA@	000F		YNPMAT	0022			

NO ERRORS NO WARNINGS

V1@15S 19/03/84 16:16:1#

```
V3@@4S    19/@3/84    15:57:28

  9@@A OFF LINE ASSEMBLER                                        SHEET    1
  SYNTAX CHECK
                    1  *          SYNTAX CHECK FOR FUP/G
                    2  *          TI9@@A
                    3  *
                    4  *          K. VO
                    5  *
                    6  *          ISSUE #
                    7  *          REVISION #
                    8  *
                    9          IDT  V3@
                   1@          HED  SYNTAX CHECK
                   11  *
                   12  *          THIS MODULE CHECKS ON VALIDITY OF PARAMETERS
                   13  *          USED IN COMMANDS
                   14  *
                   15          REF  BI2HAS,ABLENT,ABLRET,ASSTR1,ASSTR2
                   16          REF  ASSTR3,ASSTR4,ASSTR5,ASSTR6,ASSTR7
                   17          REF  HAS2BI,NOWORD,MPRED,NOSTRS,SADDR,EADDR,NWMOD
                   18          REF  IPVAL2,IPVAL3,IPVAL4,IPVAL5,IPVAL6,IPVAL7
                   19  *
                   2@          DEF  IPVAL@,A3,ASSTR@
```

```
                    21          PEJ
                    22  *
                    23  *
          0000      24  A       EQU  0
          0001      25  E       EQU  1
          0002      26  X       EQU  2
          0003      27  M       EQU  3
          0004      28  S       EQU  4
          0005      29  L       EQU  5
          0006      30  B       EQU  6
          0007      31  P       EQU  7
                    32  *
          0001      33  BR      EQU  1
          0003      34  XB      EQU  3
          CAE0      35  SVC     OPD  >CAE0,3
          0A0D      36  LFCR    EQU  >0A0D
                    37  *
P         0008      38          BRS  PROGMO
                    39  *
                    40  *
          0000      41  P0      EQU  0
          0001      42  P1      EQU  1
          0002      43  P2      EQU  2
          0003      44  P3      EQU  3
          0004      45  P4      EQU  4
          0005      46  P5      EQU  5
          0006      47  P6      EQU  6
          0007      48  P7      EQU  7
          0008      49  P8      EQU  8
          0009      50  P9      EQU  9
          000A      51  P10     EQU  10
          000B      52  P11     EQU  11
          000C      53  P12     EQU  12
          000D      54  P13     EQU  13
          000E      55  P14     EQU  14
          000F      56  P15     EQU  15
                    57  *
          8000      58  BIT0    EQU  >8000
          4000      59  BIT1    EQU  >4000
          2000      60  BIT2    EQU  >2000
          1000      61  BIT3    EQU  >1000
          0800      62  BIT4    EQU  >0800
          0400      63  BIT     EQU  >0400
          0200      64  BIT6    EQU  >0200
          0100      65  BIT7    EQU  >0100
          0080      66  BIT8    EQU  >0080
          0040      67  BIT9    EQU  >0040
```

```
                    68          PEJ
                    69  *
                    70  *       A3*  COMMAND SYNTAX CHECK
                    71  *
                    72  *            CHECK FOR VALIDITY OF ARGUMENTS ASSOCIATED WITH COMMAND
                    73  *            (EXCEPT FOR 'M' WITH 'I'). HEX ASCII STRINGS ARE
                    74  *            CONVERTED INTO BINARY VALUES
                    75  *            - FOR L,H,M,S : CALCULATES STARTING GR7 ADDR AND
                    76  *            AND # OF WORDS TO PROCESS. INDICATES IF MASK VALUE
                    77  *            IS USED IN S COMMAND
                    78  *
                    79  *            INPUTS : ASSTR1,...ASSTR7
                    80  *                     P0/P1.../P10
                    81  *
                    82  *            OUTPUT : IPVAL2,...IPVAL7,IPVAL0,NOWORD
                    83  *
  0000              84  A3      BSS  0
  0000  C550        85          RMO  L,A            SAVE RETURN
  0001  8005        86          STA  A3RETN
                    87  *
  0002  1000        88          @LDM =PROGNO        M REG CONTAINS PROGRAM ADDR.
P 0003  0008
  0004  7400        89          @BRL ABLENT         CALL ABL DRIVER TO EXECUTE SYNTAX CHECK PROGRAM
        0001
E 0005  0000
  0006  7C00        90          BRU  *A3RETN        RETURN
                    91  *
  0007              92  A3RETN  BSS  1              RETURN
```

```
                    93          PEJ
                    94  *
                    95  *           A3 (SYNTAX CHECK)
                    96  *
                    97  *
                    98  *           ALB PROGRAM SECTION
                    99  *           ===================
                   1##  *
   ###8  ###3      1#1  PROGNO DATA 3                    PROGRAM NO.
                   1#2  *
   ###9  ####      1#3  PCLUST DATA #                    PRESENT CLUSTER
   ###A  ####      1#4  PALT   DATA #                       "    ALT
   ###B  ####      1#5  PACT   DATA #                       "    ACTION
   ###C  ####      1#6  PACTOF DATA #                       "    ACTION OFFSET IN ALTERNATIVE
   ###D  ####      1#7  ACLST@ DATA #                    PTR TO ACTION LIST FORALT BEING EXE'D
                   1#8  *
   ###E  ####      1#9  SVECTR DATA #                    STATE VECTOR
                   11#  *
P  ###F  ##16      111  CLSMA@ DATA CLSMAT       POINTER TO CLUSTER MATRIX
P  ##1#  ##1A      112  NCLMA@ DATA NCLMAT          "     "  NEXT-CLUSTER VECTOR
P  ##11  ##24      113  YNPMA@ DATA YNPMAT         "     "  PREDICATE MATRIX
P  ##12  ##2E      114  MPRMA@ DATA MPRMAT         "     "  PRED MASK MATRIX
P  ##13  ##38      115  APTR#@ DATA APTR#          "     "  ACTION LIST POINTERS
                   116  *
   ##14  ####      117  BREG   DATA #                    (B)=ADDRESS OF PREVIOUS ABSTR. PROGRAM
   ##15  ####      118  LREG   DATA #                    (L)=RETURN ADDRESS
```

```
                    119          PEJ
                    120  *
                    121  *       C1   V                                  INIT
                    122  *       C2     V V V V V V V V V V              CHECK ARGUMENTS
                    123  *
                    124  *       N    2 0 0 2 0 0 0 0 0 0
                    125  *
                    126  *       P0   - Y Y - - - - - - -    L,H
                    127  *       P1   - - - Y Y - - - - -    M
                    128  *       P2   - - - - - Y Y - - -    S
                    129  *       P3   - - - - - - - Y - -    C,U,K
                    130  *       P4   - N Y - - - - - - -    TO IN 3RD ASCII STRG
                    131  *       P5   - - - N Y - - - - -    "
                    132  *       P6   - - - - - N Y - - -    MASK VALUE SPECIFIED IN 'S'
                    133  *       P7   - - - - - - - - Y -    'R','I','C'
                    134  *       P8   - - - N - - - - - Y    ALL 'M' PAR'S CHECKED OR NO ARG CMD
                    135  *
                    136  *            0 1 2 3 4 5 6 7 8 9
                    137  *
                    138  *
                    139  *
                    140  *
                    141  *            CLUSTER MATRIX  BITS0-31
                    142  *
 0016  8000  -      143  CLSMAT DATA >8000,0       C1
 0017  0000  -
 0018  7FC0  -      144          DATA >7FC0,0       C2
 0019  0000  -
                    145  *
                    146  *            NEXT CLUSTER
                    147  *
 001A  0002  -      148  NCLMAT DATA 2             ALT 0
 001B  0000  -      149          DATA 0                 1
 001C  0000  -      150          DATA 0                 2
 001D  0002  -      151          DATA 2                 3
 001E  0000  -      152          DATA 0                 4
 001F  0000  -      153          DATA 0                 5
 0020  0000  -      154          DATA 0                 6
 0021  0000  -      155          DATA 0                 7
 0022  0000  -      156          DATA 0                 8
 0023  0000  -      157          DATA 0                 9
                    158  *
                    159  *            Y/N PREDICATE MATRIX
                    160  *
 0024  0000  -      161  YNPMAT DATA >0000        ALT 0
 0025  8000  -      162          DATA >8000             1
 0026  8800  -      163          DATA >8800             2
 0027  4000  -      164          DATA >4000             3
 0028  4400  -      165          DATA >4400             4
 0029  2000  -      166          DATA >2000             5
 002A  2200  -      167          DATA >2200             6
 002B  1000  -      168          DATA >1000             7
 002C  0100  -      169          DATA >0100             8
 002D  0080  -      170          DATA >0080             9
                    171  *
                    172  *            MASK FOR PREDICATE MATRIX
```

```
                    173  *
  0020  0000        174  MPRMAT DATA >0000        ALT 0
  0021  8000        175       DATA >8000             1
  0022  8000        176       DATA >8000             2
  0023  4410        177       DATA >4410             3
  0024  4400        178       DATA >4400             4
  0025  2200        179       DATA >2200             5
  0026  2200        180       DATA >2200             6
  0027  1000        181       DATA >1000             7
  0028  0100        182       DATA >0100             8
  0029  0008        183       DATA >0008             9
                    184  *
                    185  *          ACTION MATRIX
                    186  *
P 002A  0042        187  APTR0 DATA ACLST0
P 002B  0044        188       DATA ACLST1
P 002C  004A        189       DATA ACLST2
P 002D  0051        190       DATA ACLST3
P 002E  0055        191       DATA ACLST4
P 002F  0057        192       DATA ACLST5
P 0030  0058        193       DATA ACLST6
P 0031  0059        194       DATA ACLST7
P 0032  005A        195       DATA ACLST8
P 0033  005C        196       DATA ACLST9
```

```
                    197          PEJ
                    198  *
                    199  *             - ALT 0 : INITIALIZATION - CLASSIFICATION
P 0042  0060        200  ACLST0 DATA INIT          INIT - CLASS.
  0043  0000        201          DATA 0
                    202  *
                    203  *             - ALT 1 :  L,H  FORMAT 1
P 0044  00F8        204  ACLST1 DATA HXS2BI        CONVERT HEX VALUE TO BINARY
P 0045  00D0        205          DATA BPTRB1       BUMP POINTERS BY 1
P 0046  00F8        206          DATA HXS2BI       CONVERT HEX VALUE TO BINARY
P 0047  013F        207          DATA CHMENW       CHECK MEM LOC 1 & CALC. # OF WORDS
P 0048  0154        208          DATA SAVSE        SAVE START & END ADDRESS
  0049  0000        209          DATA 0
                    210  *
                    211  *             - ALT 2 :  L,H  FORMAT 2
P 004A  00F8        212  ACLST2 DATA HXS2BI        CONVERT HEX VALUE TO BINARY
P 004B  0119        213          DATA CHKMEM       CHECK VALID GR7 MEM LOC.
P 004C  00E8        214          DATA BPTRB2       BUMP POINTER BY 2
P 004D  00F8        215          DATA HXS2BI       CONVERT HEX VALUE TO BINARY
P 004E  012E        216          DATA CHM1&2       CHECK MEM LOC 1 & LOC 2
P 004F  0154        217          DATA SAVSE        SAVE START & END ADDRESS
  0050  0000        218          DATA 0
                    219  *
                    220  *             - ALT 3 :  M
P 0051  00F8        221  ACLST3 DATA HXS2BI        CONVERT HEX VALUE TO BINARY
P 0052  00D0        222          DATA BPTRB1       BUMP POINTERS BY 1
P 0053  015E        223          DATA CHKALC       CHECK IF ALL ARG'S ARE CONVERTED
  0054  0000        224          DATA 0
                    225  *
                    226  *             - ALT 4 :  M WITH !    ** TO BE DEVELOPED
  0055  0000        227  ACLST4 DATA 0
  0056  0000        228          DATA 0
                    229  *
                    230  *             - ALT 5 :  S           ** TO BE DEVELOPED
  0057  0000        231  ACLST5 DATA 0
                    232  *
                    233  *             - ALT 6 :  S WITH MASK ** TO BE DEVELOPED
                    234  *
  0058  0000        235  ACLST6 DATA 0
                    236  *
                    237  *             - ALT 7 :  C,U,K       ** TO BE DEVELOPED
  0059  0000        238  ACLST7 DATA 0
                    239  *
                    240  *             - ALT 8 :  INVALID SYNTAX
P 005A  00F8        241  ACLST8 DATA INVALD          OUTPUT 'INVALID SYNTAX'
  005B  0000        242          DATA 0
                    243  *
                    244  *             - ALT 9 : EXIT FROM A3
  005C  0000        245  ACLST9 DATA 0
```

SYNTAX CHECK

```
                    246          PEJ
                    247  *
                    248  *            DATA SECTION
                    249  *            ************
                    250  *
  ##5D              251  IPVAL@ BSS   1          POINT TO IPVAL1,IPVAL2...
  ##5E              252  ASSTR@ BSS   1          POINT TO ONE OF ASCII STRINGS
  ##6F              253  NOARGS BSS   1          # OF ARGUMENTS
```

SYNTAX CHECK

```
                        254         PEJ
                        255  *
                        256  *          ACTION SECTION
                        257  *          **************
                        258  *
                        259  *
                        260  *     --   RESET ASCII STRING POINTER TO 1ST STRING
                        261  *          DETERMINE CLASS OF COMMAND
                        262  *          CHECK FOR 'TO', 'L', MASK VALUE FOR 'M'
                        263  *
   0060                 264  INIT   BSS  0
                        265  *
                        266  *          RESET ASCII & IPVAL POINTERS
                        267  *
   0060  0004           268          @LDA =ASSTR2      ASSTR0 = ASSTR2
         0064
E  0062  0000
   0063  0460           269          @STA ASSTR0
P  0065  005E
   0066  0008           270          @LDA =IPVAL2
         0011
E  0068  0000
   0069  00F6           271          STA  IPVAL0
                        272  *
                        273  *          RECOGNIZE CLASS OF COMMAND
   006A  0F00           274          LDE  =0           CLEAR PREDICATES
   006B  00A5           275          STE  SVECTR
   006C  17FD           276          LDX  =CMDPR-CMDPRE          # OF CMDS REQUIRING SYNTAX CHECK
   006D  0400           277          @LDA MPRED        MAIN PRED.
         006C
E  006F  0000
   0070  3000           278          @AND =>F000       P0-P3 OF MAIN PRED : CMD NO.
   0072  F000
   0073  C501           279          RMO  A,E
   0074  026A           280  A31.0   LDA  CMDPRE,X     CHECK CMD CLASS
   0076  C410           281          RCA  E,A
   0077  CDA0           282          SNE               THIS CLASS ?
   0078  7804           283          BRU  A31.2        YES
   007A  4FF8           284          BIX  A31.0        NO - CHECK NEXT CLASS
   007B  DB70           285          SMBO 0,SVECTR     COMMAND DOES NOT NEED ARGUMENT
P  007C  000E
   007D  C557           286          RMO  L,P
                        287  *
   007E  0065           288  A31.2   LDA  CMDCLS,X     SET PRED
   007F  0095           289          STA  SVECTR
                        290  *
                        291  *          CHECK FOR CORRECT # OF STRGS IN COMMAND LINE
                        292  *
   0080  0094           293          LDA  SVECTR
   0082  DB10           294          TABO 0
   0083  7814           295          BRU  A312C
   0085  0400           296          @LDA NOSTRS       'L','H'
         0080
E  0087  0000
   0088  6F03           297          CPA  =3
   0089  CDA0           298          SNE               CORRECT # OF STRINGS ?
```

```
900A OFF LINE ASSEMBLER                                          SHEET   1#
SYNTAX CHECK
   ###0   C557   299          RMO  L,P          YES
   ###1   6F#4   3##          CPA  =4           L LOC1 TO LOC2 ?
   ###2   CD2#   3#1          SEQ
   ###3   782D   3#2          BRU  A3129        NO - INVALID # OF STRINGS
   ###4   #4##   3#3          @LDA ASSTR3       YES
          ###5
E  ###5   ####
   ###6   68##   3#4          CPA  $+1          3RD STRING IS 'TO' ?
   ###7   D4CF   3#5          DATA 'TO'
   ###8   CD2#   3#6          SEQ
   ###9   7827   3#7          BRU  A3129        NO - UNVALID STRING
   ###A   #4##   3#8          @LDA SVECTR       SET PRED
P  ###B   ###E
   ###C   D854   3#9          SABO P4
   ###D   84##   31#          @STA SVECTR
P  ###E   ###E
   ###F   C557   311          RMO  L,P          RETURN
                 312  *
   ##9#   DB11   313  A312#   TABO 1
   ##91   78#A   314          BRU  A3122
   ##92   #4##   315          @LDA NOSTRS       'M'
          ###D
E  ##93   ####
   ##94   2F#1   316          SUB  =1
   ##95   8#C9   317          STA  NOARGS       # OF ARGUMENTS FOR CONVERSION
   ##96   2F#1   318          SUB  =1
   ##97   84##   319          @STA NWMOD        # OF WORDS TO MODIFY IN FUP/G
          ##1#
E  ##98   ####
   ##99   CCE#   32#          SPL  A
   ##9A   7816   321          BRU  A3129        INVALID # OF STRGS
   ##9B   7817   322          BRU  A31.3        CHECK FOR 'TO','I'
                 323  *
   ##9C   DB12   324  A3122   TABO 2
   ##9D   78#E   325          BRU  A3124
   ##9E   #4##   326          @LDA NOSTRS       'S'
          ###D
E  ##9F   ####
   ##A#   6F#6   327          CPA  =6
   ##A1   CDA#   328          SNE
   ##A2   C557   329          RMO  L,P
   ##A3   6F#7   33#          CPA  =7
   ##A4   CD2#   331          SEQ
   ##A5   78#B   332          BRU  A3129        UNVALID # OF STRGS
   ##A6   #4##   333          @LDA SVECTR
P  ##A7   ###E
   ##A8   D856   334          SABO P6           FOR MASK VALUE
   ##A9   84##   335          @STA SVECTR
P  ##AA   ###E
   ##AB   78#7   336          BR@  A31.3
                 337  *
   ##AC   #4##   338  A3124   @LDA NOSTRS
          ###D
E  ##AD   ####
   ##AE   6F#2   339          CPA  =2
   ##AF   CDA#   34#          SNE
```

```
9900A OFF LINE ASSEMBLER                                        SHEET   11
SYNTAX CHECK
   0080  C557    341          RMO  L,P
                 342  *
   0081          343  A3129  BSS  0
   0081  DB74    344          SMBO P4,MPRED     SET 'INVALID' PREDICATE IN MAIN STATE VECTOR
         008C
E  0082  0000
                 345  *
                 346  *       CHECK FOR 'TO', 'I'
                 347  *
   0083  0400    348  A31.3  @LDA ASSTR3        TO OR I MUST BE IN STRING 3
         0005
E  0084  0000
   0085  6800    349          CPA  $+1
   0086  D4CF    350          DATA 'TO'
   0087  CD20    351          SEQ                'TO' ?
   0088  7006    352          BRU  A31.4         NO - GO CHECK 'I'
   0089  0400    353          @LDA SVECTR        YES- SET PRED P4
P  008A  000E
   008B  DB54    354          SABO P4
   008C  8400    355          @STA SVECTR
P  008D  000E
   008E  7000    356          BRU  A31.4         CHECK IF MASK VALUE SPECIFIED
                 357  *
   008F  0000    358  A31.4  @LDE =>00FF         SET RH BYTE TO 1
   00C0  00FF
   00C1  C490    359          ROR  E,A
   00C2  6800    360          CPA  $+1           'I'
   00C3  A1FF    361          DATA 'I'
   00C4  CD20    362          SEQ
   00C5  C557    363          RMO  L,P           NO  - RETURN
   00C6  0400    364          @LDA SVECTR        YES - SET PRED P5
P  00C7  000E
   00C8  DB55    365          SABO P5
   00C9  8400    366          @STA SVECTR
P  00CA  000E
   00CB  C557    367          RMO  L,P           RETURN
                 368  *
   00CC  0400    369  A31.6  @LDA NOSTRS
         000D
E  00CD  0000
   00CE  6F07    370          CPA  -7
   00CF  CD20    371          SEQ                MASK VALUE SPECIFIED ?
   00D0  C557    372          RMO  L,P           NO
   00D1  0400    373          @LDA SVECTR
P  00D2  000E
   00D3  DB56    374          SABO P6
   00D4  8400    375          @STA SVECTR
P  00D5  000E
   00D6  C557    376          RMO  L,P
                 377  *
```

```
                    378          PEJ
                    379  *
                    380  *               MAIN PREDICATES
0097  1000          381  CMDPR   DATA >1000          L
0098  2000          382          DATA >2000          H
0099  3000          383          DATA >3000          H
009A                384  CMDPRE  BSS  0
                    385  *
                    386  *               A3 PREDICATES
009A  8000          387          DATA BIT0          L, CLASS 0
009B  8000          388          DATA BIT0          H, CLASS 0
009C  4000          389          DATA BIT1          H, CLASS 1
009D                390  CMDCLS  BSS  0
```

```
                    391          PEJ
                    392  *
                    393  *       --   BUMP ASCII STRING PTR AND I/P VALUE PTR BY 1
                    394  *
                    395  *            INPUTS : ASSTR@,IPVAL@
                    396  *            OUTPUTS: ASSTR@,IPVAL@
                    397  *
  0@0D              398  BPTRB1 BSS  @
  0@0D  54@@        399          @IMO IPVAL@
P 0@DE  0@5D
  0@DF  17F9        4@@          LDX  *-7
  0@E@  @4@@        4@1          @LDA ASSTR@
P 0@E1  0@5E
  0@E2  6215        4@2  A32.@   CPL  ASSTR,X
  0@E3  CDA@        4@3          SNE
  0@E4  70@1        4@4          BRU  *+2
  0@E5  4@FC        4@5          BIX  A32.@
  0@E6  C322        4@6          RIN  X,X
  0@E7  @21@        4@7          LDA  ASSTR,X
  0@E8  84@@        4@@          @STA ASSTR@
P 0@E9  0@5E
  0@EA  C557        4@9          RMO  L,P
                    41@  *
                    411  *       --   BUMP ASCII STRING PTR AND I/P VALUE PTR BY 2
                    412  *
                    413  *            INPUTS : ASSTR@,IPVAL@
                    414  *            OUTPUTS: ASSTR@,IPVAL@
                    415  *
  0@EB  C55@        416  BPTRB2 RMO  L,A
  0@EC  8@@3        417          STA  A33RET
  0@ED  7@EF        418          BRL  BPTRB1
  0@EE  7@EE        419          BRL  BPTRB1
  0@EF  7C@@        42@          BRU  *A33RET
                    421  *
  0@F@              422  A33RET BSS  1
                    423  *
        0@@3        424          DATA ASSTR1
E 0@F1  0@@@
        0@@4        425          DATA ASSTR2
E 0@F2  0@@@
        0@@5        426          DATA ASSTR3
E 0@F3  0@@@
        0@@6        427          DATA ASSTR4
E 0@F4  0@@@
        0@@7        428          DATA ASSTR5
E 0@F5  0@@@
        0@@@        429          DATA ASSTR6
E 0@F6  0@@@
        0@@9        43@          DATA ASSTR7
E 0@F7  0@@@
  0@F@              431  ASSTR  BSS  @
                    432  *
                    433  *
                    434  *       -- SET INVALID PAR. IN MAIN PREDICATES
                    435  *
  0@F@  DB74        436  INVALD SMBO P4,MPRED
```

```
SYNTAX CHECK
        0000C
E 00F8  0000
  00FA  C557   437          RMO  L.P          RETURN
```

SYNTAX CHECK

```
                    438          PEJ
                    439  *
                    440  *       --   CHECK FOR HEX ASCII AND CONVERT TO BINARY VALUE
                    441  *
  0FFB              442  HXS2BI  BSS  0
  0FFB  C550        443          RMO  L,A          SAVE RETURN
  0FFC  8018        444          STA  A34RET
                    445  *
  0FFD  0400        446          @LDA ASSTR@
P 0FFE  005E
  0FFF  2F02        447          SUB  =2           POINT TO OS
  1000  0000        448          STA  TEMP
  1001              449  TEMP    BSS  1
  1002  04FE        450          LDA  *TEMP        IS
  1003  3823        451          AND  RHBYTE       ...IN RH BYTE
  1004  C502        452          RMO  A,X
  1005  50FB        453          IMO  TEMP         OS
  1006  04FA        454          LDA  *TEMP        ...IN RH BYTE
  1007  381F        455          AND  RHBYTE
  1008  C002        456          RSU  A,X          # OF ASCIIS IN (X)
  1009  C322        457          RIN  X,X          BECAUSE UNPACK IS NOT WORKING AS CLAIMED
  100A  0400        458          @LDA ASSTR@       (A)=START OF ASCII STRING
P 100B  005E
  100C  7400        459          @BRL HAS2BI       CONVERT ASCII STRING TO BINARY
                    000A
E 100D  0000
  100E  CC22        460          SOO  X            CONVERSION SUCCESSFUL ?
  100F  7802        461          BRU  HX2BI0       YES
  1010  DB74        462          SMBO 4,MPRED      NO - SET 'INVALID'IN MAIN PREDICATE
                    000C
E 1011  0000
  1012  0C00        463  HX2BI0  @LDE IPVAL@
P 1013  005D
  1014  8000        464          STE  IPVL@
  1015              465  IPVL@   BSS  1
  1016  04FE        466          STA  *IPVL@       SAVE CONVERTED BIN VALUE
  1017  7C00        467          BRU  *A34RET      RETURN
                    468  *
  1018              469  A34RET  BSS  1
```

```
                    470  *       PEJ
                    471  *
                    472  *       --    CHECK FOR VALID GR7 MEMMV LOCATION
                    473  *             DETERMINED BY POINTER IPVAL0
                    474  *
  0119              475  CHKMEM  BSS   0
  0119  0C00        476          @LDE  IPVAL0
P 011A  0050
  011B  00F9        477          STE   IPVL0
  011C  04F8        478          LDA   *IPVL0
  011D  6002        479          CPL   GR7LO       GR7 LO < MEM LOC < GR7 HI ?
  011E  CDC0        480          SLE
  011F  7804        481          BRU   A35.0       NO - NO GOOD
  0120  6004        482          CPL   GR7HI
  0121  CD00        483          SGE
  0122  7801        484          BRU   A35.0       NO     "
  0123  C557        485          RMO   L,P         RETURN
                    486  *
  0124  DB74        487  A35.0   SMBO  P4,MPRED    INVALID SYNTAX IN MAIN PRED.
        000C
E 0125  0000
  0126  C557        488          RMO   L,P
                    489  *
  0127  00FF        490  RHBYTE  DATA  >00FF
  0128  0000        491  GR7LO   DATA  0
  0129  FFFF        492  GR7HI   DATA  >FFFF
                    493  *
        0011        494  IPVA20  DATA  IPVAL2
E 012A  0000
        0012        495  IPVA30  DATA  IPVAL3
E 012B  0000
        0013        496  IPVA40  DATA  IPVAL4
E 012C  0000
        0014        497  IPVA50  DATA  IPVAL5
E 012D  0000
```

```
                    498          PEJ
                    499  *
                    5##  *       --    CHECK MEM LOC 1 < MEM LOC 2
                    5#1  *             CALCULATE # OF WORDS
                    5#2  *
   #12E             5#3  CHM1&2  BSS   #
   #12E   #4FD      5#4          LDA   *IPVA4#
   #12F   6#F9      5#5          CPL   GR7HI          MEM LOC 2 <= GR7 HI ?
   #13#   CD8#      5#6          SGE
   #131   78#A      5#7          BRU   A36.2          NO
   #132   2CF7      5#8          SUB   *IPVA2#        MEM LOC 1 < MEM LOC 2 ?
   #133   CCE#      5#9          SPL   A
   #134   78#7      51#          BRU   A36.2          NO
   #135   CC8#      511          SNZ   A
   #136   78#5      512          BRU   A36.2          NO
   #137   C841      513          LRA   1              YES - CONVERT TO # OF WORDS
   #138   27#1      514          ADD   =1             INCLUDE THE LAST WORD
   #139   84##      515          @STA  NOWORD         # OF WORDS TO UPDATE
          ####B
E  #13A   ####
   #13B   C557      516          RMO   L,P            RETURN
                    517  *
   #13C   DB74      518  A36.2    SMBO  P4,MPRED       SET P13 (INVALID SYNTAX) IN MAIN PREDICATES
          ###C
E  #13D   ####
   #13E   C557      519          RMO   L,P
```

```
                    52#          PEJ
                    521
                    522  *       --    CHECK IF MEM LOC 1 + # OF WORDS <= GR7 HI
                    523  *             CALCULATE # OF WORDS
                    524  *
  #13F              525  CHMEMW  BSS   #
  #13F  #4EA        526          LDA   *IPVA2#       BYTE LOC 1
  #14#  24EA        527          ADD   *IPVA3#       # OF BYTES
  #141  6#E7        528          CPL   GR7HI         <= GR7 HI LIMIT ?
  #142  CD4#        529          SGT
  #143  78#5        53#          BRU   A37.2         NO
  #144  #4E6        531          LDA   *IPVA3#       # OF BYTES
  #145  C841        532          LRA   1             CONVERT TO # OF WORDS
  #146  84#5        533          #STA  NOWORD
        ####
E #147  ####
  #148  C557        534          RMO   L,P
                    535  *
  #149  DB74        536  A37.2    SMBO  P4,MPRED      SET P4 IN MAIN PRED.
        ###C
E #14A  ####
  #14B  C557        537          RMO   L,P
```

SYNTAX CHECK

```
                    538          PEJ
                    539  *
                    540  *       --   CHECK FOR VALID GRAPHIC UNIT FOR C,U,K CMDS
                    541  *
  0140  0400        542  CHGUNI @LDA IPVAL3       GRAPHIC UNIT
        0012
E 0142  0000
  0144  6F05        543          CPA  =5
  0146  CD00        544          SLT              VALID UNIT# ?
  0148  C557        545          RMO  L,P         YES
  014A  DB7C        546          SMBO P12,MPRED   NO - SET ERROR PRED
        000C
E 014C  0000
  014E  C557        547          RMO  L,P
```

```
9900A OFF LINE ASSEMBLER                                    SHEET  2#
SYNTAX CHECK

                        548          PEJ
                        549  *
                        55#  *       --   SAVE START & END GR7 ADDRESSES OF BUFFER
                        551  *
  #154                  552  SAVSE   BSS  #
  #154  #4##            553          @LDA IPVAL2        START BYTE ADDR
        ##11
E #155  ####
  #156  04##            554          @STA SADDR
        ###E
E #157  ####
  #158  24##            555          ADD  *NWORD@
        ###B            556  NWORD@  DATA NOWORD
E #159  ####
  #15A  24FE            557          ADD  *NWORD@       END BYTE ADDR
  #15B  04##            558          @STA EADDR
        ###F
E #15C  ####
  #15D  C557            559          RMO  L,P
                        56#  *
                        561  *       --   CHECK IF ALL WORDS IN 'M' ARE CONVERTED
                        562  *            INTO BINARY VALUES
                        563  *
  #15E                  564  CHKALC  BSS  #
  #15E  4C#4            565          DMT  *NOARG@
  #15F  78#2            566          BRU  CHKAL9
  #16#  DB78            567          SMBO 8,SVECTR      ALL CONVERTED
P #161  ###E
  #162  C557            568  CHKAL9  RMO  L,P
                        569  *
P #163  ##5F            57#  NOARG@  DATA NOARGS
                        571  *
                        572          END
```

SYNTAX CHECK

```
   A        ####     A3       ####     A31.#    ##6F     A31.2    ##77     A31.3    ##B3
   A31.4    ##8F   R A31.6    ##CC     A312#    ##9#     A3122    ##9C     A3124    ##AC
   A3129    ##B1     A32.#    ##E2     A33RET   ##F#     A34RET   #118     A35.#    #124
   A36.2    #13C     A3?2     #149     A3RETN   ###7     ABLENT   ###1   R ABLRET   ###2
   ACLST#   ##42     ACLST1   ##44     ACLST2   ##4A     ACLST3   ##51     ACLST4   ##55
   ACLST5   ##57     ACLST6   ##58     ACLST7   ##59     ACLST8   ##5A     ACLST9   ##5C
 R ACLST#   ###D     APTR#    ##38   R APTR#0   ##13     ASSTR    ##F8     ASSTR1   ###3
   ASSTR2   ###4     ASSTR3   ###5     ASSTR4   ###6     ASSTR5   ###7     ASSTR6   ###8
   ASSTR7   ###9     ASSTR#   ##5E   R B        ###6   R BI2HAS   ####   R BIT      #4##
   BIT#     8###     BIT1     4###   R BIT2     2###   R BIT3     1###   R BIT4     ####
 R BIT6     #2##   R BIT7     #1##   R BIT8     ##8#   R BIT9     ##4#     BPTR81   ##DD
   BPTR82   ##E8   R BR       ###1   R BREG     ##14   R CHGUNI   #14C     CHKAL9   #162
   CHKALC   #15E     CHKMEM   #119     CHM1&2   #12E     CHMENV   #13F   R CLSMA0   ###F
   CLSMAT   ##16     CMDCLS   ##DD     CMDPR    ##D7     CMDPRE   ##DA     E        ###1
   EADDR    ###F     GR7HI    #129     GR7LO    #128     HAS2BI   ###A     HX2BI#   #112
   HXS2BI   ##F8     INIT     ##6#     INVALD   ##F8     IPVA2@   #12A     IPVA3@   #12B
   IPVA4@   #12C   R IPVA5@   #12D     IPVAL2   ##11     IPVAL3   ##12     IPVAL4   ##13
   IPVAL5   ##14   R IPVAL6   ##15   R IPVAL7   ##16     IPVAL@   ##5D     IPVL@    #115
   L        ###5   R LFCR     8A8D   R LREG     ##18   R M        0##3     MPRED    ###C
 R MPRMA@   ##12     MPRMAT   ##2E   R NCLMA@   ##1#     NCLMAT   ##1A     NOARG@   #163
   NOARGS   ##5F     NOSTRS   ###D     NOWORD   ###B     NWMOD    ##1#     NWORD@   #159
   P        ###7   R P#       ####   R P1       ###1   R P1#      ###A   R P11      ###B
   P12      ###C   R P13      ###D   R P14      ###E   R P15      ###F   R P2       ###2
 R P3       ###3     P4       ###4     P5       ###5     P6       ###6   R P7       ###7
 R P8       ###8   R P9       ###9   R PACT     ###B   R PACTOF   ###C   R PALT     ###A
 R PCLUST   ###9     PROGNO   ###8     RHBYTE   #127   R S        ###4     SADDR    ###E
   SAVSE    #154   R SVC      CAE#     SVECTR   ###E     TEMP     #1#1     X        ###2
 R XB       ###3   R YNPMA@   ##11     YNPMAT   ##24
```

NO ERRORS NO WARNINGS

V3##4S 19/#3/84 15:58:#7

```
                1  *          GR7 DISASSEMBLER
                2  *          TI9#AA
                3  *
                4  *          K. VO
                5  *
                6  *          ISSUE #
                7  *          REVISION #
                8  *
                9          IDT  V4#
               1#          HED  GR7 DISASSEMBLER
               11  *
               12          DEF  A5,A6,PAGENO
               13          REF  ABLENT,BI2HAS,BI2DAS,BI2OAS
               14          REF  ASSTR1,ASSTR2,ASSTR3,ASSTR4,ASSTR5,ASSTR6,ASSTR7
               15          REF  IPVAL2,IPVAL3,IPVAL4,IPVAL5,IPVAL6,IPVAL7
               16          REF  MOVORD,GBUFF,GBUFF#
               17          REF  OPEFLD,B####3,B#7FF,B#1FF,B3F##,B###3F,B#FFF,B####7
               18          REF  B7F##,B##7F,BFFFF,B7FFF,B###3#
               19          REF  GRMASK,GRINST,GRFORM,GRMNE,WTC
               2#          REF  RPRB,GI,HBUF1,HBUF2,BHBUF2,NOINS
```

```
                    21          PEJ
                    22  *
                    23  *
          0000      24  A       EQU  0
          0001      25  E       EQU  1
          0002      26  X       EQU  2
          0003      27  M       EQU  3
          0004      28  S       EQU  4
          0005      29  L       EQU  5
          0006      30  B       EQU  6
          0007      31  P       EQU  7
                    32  *
          0001      33  BR      EQU  1
          0003      34  XB      EQU  3
          CAE0      35  SVC     OPD  >CAE0,3
          0A0D      36  LFCR    EQU  >0A0D
                    37  *
P         007D      38          BRS  PROGNO
                    39  *
          0000      40  P0      EQU  0
          0001      41  P1      EQU  1
          0002      42  P2      EQU  2
          0003      43  P3      EQU  3
          0004      44  P4      EQU  4
          0005      45  P5      EQU  5
          0006      46  P6      EQU  6
          0007      47  P7      EQU  7
          0008      48  P8      EQU  8
          0009      49  P9      EQU  9
          000A      50  P10     EQU  10
          000B      51  P11     EQU  11
          000C      52  P12     EQU  12
          000D      53  P13     EQU  13
          000E      54  P14     EQU  14
          000F      55  P15     EQU  15
```

```
                    56          PEJ
                    57  *
                    58  *       A6   DISASSEMBLE GR7 INSTRUCTIONS INTO MNEMONIC
                    59  *            FORMS AND OUTPUT TO LINE PRINTER
                    60  *
   0000             61  A6      BSS  0
   0000  1800       62          @LDM =PRBAO         OPEN LP SPOOL ON LUN 3
P  0001  037F
   0002  CAE8       63          SVC  8
   0003  0701       64          LDA  =1             FLAG TO DIRECT OUTPUT TO LP
   0004  8400       65          @STA LORH
P  0005  00A3
                    66  *
   0006  0000       67          @LDA =HBUF1         INIT POINTERS TO H BUFFER
         0028
E  0007  0000
   0008  8400       68          @STA IGBUF@
P  0009  00A1
   000A  8400       69          @STA AGBUF@
P  000B  00A2
   000C  0701       70          LDA  =1             RESET PAGE NO. TO 1ST PAGE
   000D  8400       71          @STA PAGENO
P  000E  00A6
   000F  1800       72          @LDM =HD1PRL
P  0010  0427
   0011  CAE0       73          SVC  0
   0012  1800       74          @LDM =HD2PRL
P  0013  0444
   0014  CAE0       75          SVC  0
   0015  1800       76          @LDM =HD3PRL
P  0016  0462
   0017  CAE0       77          SVC  0
   0018  781F       78          BRU  L007           'H' SHARES THE SAME ABL WITH 'L'
                    79  *
                    80  *       A5   DISASSEMBLE GR7 INSTRUCTIONS INTO MNEMONIC FORM
                    81  *            AND OUTPUT TO CONSOLE
                    82  *
   0019             83  A5      BSS  0
   0019  0700       84          LDA  =0
   001A  8400       85  L005    @STA HD1PRB
P  0018  042E
   001C  8400       86          @STA HD2PRB
P  001D  044B
   001E  8400       87          @STA HD3PRB
P  001F  0469
   0020  8400       88          @STA OPCPRB
P  0021  02CD
   0022  8400       89          @STA LORH
P  0023  00A3
                    90  *
                    91  *            OUTPUT HEADER TO CONSOLE
                    92  *
   0024  DB20       93          TMBZ 0,FSTL         1ST 'L' COMMAND ?
P  0025  00A7
   0026  780B       94          BRU  L006           NO
   0027  1800       95          @LDM =HD1PRB        1ST LINE OF HEADER
```

```
9845A OFF LINE ASSEMBLER                                        SHEET  4
GR7 DISASSEMBLER
P 0028  042E
  0029  CAE0     96          SVC  0
  002A  1800     97          @LDM =HD2PRB      2ND  "   "   "
P 002B  044B
  002C  CAE0     98          SVC  0
  002D  1800     99          @LDM =HD3PRB      3RD  "   "   "
P 002E  0469
  002F  CAE0     100         SVC  0
  0030  DB70     101         SMB0 0,FSTL       RESET 1ST 'L' FLAG
P 0031  00A7
                  102  *
  0032  0000     103  L006  @LDA =GBUFF        INIT BUFFER POINTER
                  0012
E 0033  0000
  0034  8400     104         @STA AGBUF@
P 0035  00A2
  0036  8400     105         @STA IGBUF@
P 0037  00A1
                  106  *
  0038  C550     107  L007  RMO  L,A           SAVE RETURN
  0039  8016     108         STA  A5RETN
                  109  *
  003A  0700     110         LDA  =0           RESET # OF WORDS PROCESSED
  003B  8400     111         STA  *S+1
P 003C  00A4     112         DATA NOWDP
  003D  8400     113         @STA LINENO       RESET LINE #
P 003E  00A5
                  114  *
  003F  0400     115         @LDA IPVAL2       INIT GR7 MEM LOCATION
                  0008
E 0040  0000
  0041  8400     116         @STA GRLOC1
P 0042  00A0
  0043  B00E     117         DLD  8888         BLANK OUT LINE BUFFER
  0044  17D8     118         LDX  =-80/2
  0045  A600     119  A51.2  @DST OPBUF+40,X
P 0046  007C
  0047  C322     120         RIN  X,X
  0048  40FC     121         BIX  A51.2
```

```
                    122             PEJ
                    123   *
                    124   *         CALL ABL DRIVER TO DISASSEMBLE (NOWORD) GR7 INSTRUCTIONS
                    125   *         TO CONSOLE ('L') OR TO LP ('H')
                    126   *
                    127   *         INPUT : NOWORD
                    128   *
  0049              129   DISAS0    BSS   0
  0049  0700        130             LDA   =0          CLEAR PREDICATES FOR ABL PROGRAM
  004A  0038        131             STA   SVECTR
  004B  1800        132             @LDM  =PROGNO     ABL PROGRAM
P 004C  007D
  004D  7400        133             @BRL  ABLENT
        0000
E 004E  0000
  004F  7C00        134             BRU   *A5RETN     YES - RETURN
                    135   *
  0050              136   A5RETN    BSS   1
        0025        137   WTC@      DATA  WTC
E 0051  0000
  0052  A0A0        138   BBBB      DATA  '  '
  0053  A0A0
  0054              139   OPBUF     BSS   80/2        OUTPUT BUFFER
  007C  0A0D        140             DATA  LECR
```

```
                    141          PEJ
                    142  *
                    143  *            ABL PROGRAM SECTION
                    144  *            ===================
                    145  *
  ##7D   ###3       146  PROGNO  DATA 3                    PROGRAM NO.
                    147  *
  ##7E   ####       148  PCLUST  DATA #                    PRESENT CLUSTER
  ##7F   ####       149  PALT    DATA #                      *     ALT
  ###8   ####       15#  PACT    DATA #                      *     ACTION
  ###1   ####       151  PACTOF  DATA #                      *     ACTION OFFSET IN ALTERNATIVE-
  ###2   ####       152  ACLST@  DATA #                    PTR TO ACTION LIST FORALT BEING EXE'D
                    153  *
  ###3   ####       154  SVECTR  DATA #                    STATE VECTOR
                    155  *
P ###4   ##A8       156  CLSMA@  DATA CLSMAT          POINTER TO CLUSTER MATRIX
P ###5   ##AE       157  NCLMA@  DATA NCLMAT             *     *  NEXT-CLUSTER VECTOR
P ###6   ##C4       158  YNPMA@  DATA YNPMAT             *     *  PREDICATE MATRIX
P ###7   ##DA       159  MPRMA@  DATA MPRMAT             *     *  PRED MASK MATRIX
P ###8   ##F#       16#  APTR#@  DATA APTR##              *     *  ACTION LIST POINTERS
                    161  *
  ###9   ####       162  BREG    DATA #                    (B)=ADDRESS OF PREVIOUS ABSTR. PROGRAM
  ###A   ####       163  LREG    DATA #                    (L)=RETURN ADDRESS
```

```
                    164          PEJ
                    165  *
                    166  *              DATA SECTION
                    167  *              ------------
                    168  *
P ##8B   ##8C       169  MNEM#   DATA MNEM1
  ##8C              17#  MNEM1   BSS  2          MNEMONIC STRING 1
  ##8E              171  MNEM2   BSS  3          MNEMONIC STRING 2
  ##91   A#AC       172          DATA ' ,'
  ##92              173  MNEM3   BSS  3          MNEMONIC STRING 3
  ##95   A#AC       174          DATA ' ,'
  ##96              175  MNEM4   BSS  2          MNEMONIC STRING 4
  ##9B              176  MNEM5   BSS  #
  ##98   ####       177  MNEPTR  DATA #          POINTER USED TO POINT MNEM# TO NEXT STRG
P ##99   ##8C       178  MNEM1#  DATA MNEM1
P ##9A   ##8E       179          DATA MNEM2
P ##9B   ##92       18#          DATA MNEM3
P ##9C   ##96       181          DATA MNEM4
                    182  *
  ##9D              183  IBUFF#  BSS  1          PTR TO INST BUFFER
  ##9E              184  IBUFF   BSS  2
  ##A#              185  GRLOC1  BSS  1          MEM LOC 1
  ##A1              186  IGBUF#  BSS  1          POINTER TO INST. WD.
  ##A2              187  AGBUF#  BSS  1          "   "  ADDR. WORD
  ##A3              188  LORH    BSS  1          BIT#=# : FOR 'L', =1 FOR 'H'
  ##A4              189  NOWDP   BSS  1          NO. OF WORDS PROCESSED
  ##A5              19#  LINENO  BSS  1          LINE # WITHIN THE PAGE
       ##26        191  LPERP   EQU  38         NO. OF LINES / PAGE
  ##A6              192  PAGENO  BSS  1          PAGE NO. (FOR 'H')
  ##A7   ####       193  FSTL    DATA #          #=# ; 1ST 'L' COMMAND
```

```
                194         PEJ
                195  *
                196  *C1    V V                                                 GET OPCODE,FORMAT
                197  *C2        V V V V V V V V V V V V V V V V                  DISASS. INSTR. WD
                198  *C3                                    V V V V            DISASS. 2ND WD
                199  *
                2##  *N     2 # 1 1 1 1 1 1 1 1 1-3 3 3 3 3 3 1 1 1 1 1
                2#1  *
                2#2  *P#    - - N N N N N N N N Y Y Y Y Y Y Y - - - - -
                2#3  *P1    - - N N N N Y Y Y Y N N N N Y Y Y - - - - -    . FORMAT NO.
                2#4  *P2    - - N N Y Y N N Y Y N N Y Y N N Y - - - - -    .
                2#5  *P3    - - N Y N Y N Y N Y N Y N Y N Y N - - - - -    .
                2#6  *P4    - - N N N N N N N N N N N N N N Y - - - -    INVALID OP-CODE
                2#7  *P5    N Y- - - - - - - - - - - - - - - - N N - Y   ALL WORDS PROCESSED
                2#8  *P6    - - - - - - - - - - - - - - - - - - Y N - -   IND. BIT IN ADDR.
                2#9  *P7    - - - - - - - - - - - - - - - - - - - - Y -   2ND WD REAL INC
                21#  *
                211  *      # 1 2 3 4 5 6 7 8 9 1# 12 14 16 18 2#
                212  *                          11 13 15 17 19 21
                213  *
                214  *
                215  *      ALT# INIT $ RECORGNIZE OP-CODE
                216  *         1 ALL LOCATIONS PROCESSED
                217  *         2 FORMAT #  NO OPERAND
                218  *         3    "     1  I B14-15
                219  *         4    "     2  I B5-15
                22#  *         5    "     3, INST B#-B6; REAL B7-15
                221  *         6    "     4, INST B#-1&B8-9; R1 B2-7  R2B1#-15
                222  *         7    "     5, INST B#-4; R B5-15
                223  *         8    "     6, INST B#-B9; I B1#-15
                224  *         9    "     7, INST B#&B8; I1 B1-7  I2 B9-15
                225  *         1#   "     8, INST B#-8; I B9-15
                226  *         11   "     9, INST B#-9
                227  *                     ADDR B#-15
                228  *         12   "    1#, INST B#-9
                229  *                     ADDR B1-15; IND B#
                23#  *         13   "    11, INST B#-B9; I1 B1#-12  I2 B13-15
                231  *                     DATA B4-15
                232  *         14   "    12, INST B#-13; I B14-15
                233  *                     ADDR B#-15
                234  *         15   "    13, INST B#-B9
                235  *                     INC REAL B#-B15
                236  *         16   "    14, INST B#-B13, I B14-B15
                237  *                     INC REAL B#-B15
                238  *         17 UNVALID OP-CODE
                239  *         18 2ND WORD ADDRESS, BIT # INDIRECT
                24#  *         19 2ND WORD ADDRESS
                241  *         2# 2ND WORD REAL INCREMENT
                242  *         21 1ST WORD OF D.W. INST. THE LAST WORD
```

```
                    243          PEJ
                    244  *
                    245  *            CLUSTER MATRIX  BITS#-31
                    246  *
  ##A8  C###        247  CLSMAT DATA >C###,#       C1
  ##A9  ####
  ##AA  3FFF        248          DATA >3FFF,>C###  C2
  ##AB  C###
  ##AC  ####        249          DATA >####,>3C##  C3
  ##AD  3C##
                    25#  *
                    251  *            NEXT CLUSTER
                    252  *
                    253  *                          ALT
  ##AE  ###2        254  NCLMAT DATA 2               #
  ##AF  ####        255          DATA #              1
  ##B#  ###1        256          DATA 1              2
  ##B1  ###1        257          DATA 1              3
  ##B2  ###1        258          DATA 1              4
  ##B3  ###1        259          DATA 1              5
  ##B4  ###1        26#          DATA 1              6
  ##B5  ###1        261          DATA 1              7
  ##B6  ###1        262          DATA 1              8
  ##B7  ###1        263          DATA 1              9
  ##B8  ###1        264          DATA 1              1#
  ##B9  ###3        265          DATA 3              11
  ##BA  ###3        266          DATA 3              12
  ##BB  ###3        267          DATA 3              13
  ##BC  ###3        268          DATA 3              14
  ##BD  ###3        269          DATA 3              15
  ##BE  ###3        27#          DATA 3              16
  ##BF  ###1        271          DATA 1              17
  ##C#  ###1        272          DATA 1              18
  ##C1  ###1        273          DATA 1              19
  ##C2  ###1        274          DATA 1              2#
  ##C3  ###1        275          DATA 1              21
                    276  *
                    277  *            Y/N PREDICATE MATRIX
                    278  *
                    279  *                          ALT
  ##C4  ####        28#  YNPMAT DATA >####           #
  ##C5  #4##        281          DATA >#4##          1
  ##C6  ####        282          DATA >####          2
  ##C7  1###        283          DATA >1###          3
  ##C8  2###        284          DATA >2###          4
  ##C9  3###        285          DATA >3###          5
  ##CA  4###        286          DATA >4###          6
  ##CB  5###        287          DATA >5###          7
  ##CC  6###        288          DATA >6###          8
  ##CD  7###        289          DATA >7###          9
  ##CE  8###        29#          DATA >8###          1#
  ##CF  9###        291          DATA >9###          11
  ##D#  A###        292          DATA >A###          12
  ##D1  B###        293          DATA >B###          13
  ##D2  C###        294          DATA >C###          14
  ##D3  D###        295          DATA >D###          15
```

```
9900A OFF LINE ASSEMBLER                                        SHEET   10
GR7 DISASSEMBLER
    0000  E000    296          DATA >E000          16
    0002  0800    297          DATA >0800          17
    0004  0200    298          DATA >0200          18
    0006  0000    299          DATA >0000          19
    0008  0100    300          DATA >0100          20
    000A  0400    301          DATA >0400          18
                  302  *
                  303  *            MASK FOR PREDICATE MATRIX
                  304  *
                  305  *                                ALT
    000C  0400    306  MPRMAT  DATA >0400          0
    000E  0400    307          DATA >0400          1
    0010  F800    308          DATA >F800          2
    0012  F800    309          DATA >F800          3
    0014  F800    310          DATA >F800          4
    0016  F800    311          DATA >F800          5
    0018  F800    312          DATA >F800          6
    001A  F800    313          DATA >F800          7
    001C  F800    314          DATA >F800          8
    001E  F800    315          DATA >F800          9
    0020  F800    316          DATA >F800          10
    0022  F800    317          DATA >F800          11
    0024  F800    318          DATA >F800          12
    0026  F800    319          DATA >F800          13
    0028  F800    320          DATA >F800          14
    002A  F800    321          DATA >F800          15
    002C  F800    322          DATA >F800          16
    002E  0800    323          DATA >0800          17
    0030  0600    324          DATA >0600          18
    0032  0600    325          DATA >0600          19
    0034  0100    326          DATA >0100          20
    0036  0400    327          DATA >0400          21
                  328  *
                  329  *            ACTION MATRIX
                  330  *
P   0038  0106    331  APTR00  DATA ACLST0
P   003A  010A    332          DATA ACLST1
P   003C  010C    333          DATA ACLST2
P   003E  0111    334          DATA ACLST3
P   0040  0118    335          DATA ACLST4
P   0042  011F    336          DATA ACLST5
P   0044  0126    337          DATA ACLST6
P   0046  012F    338          DATA ACLST7
P   0048  0136    339          DATA ACLST8
P   004A  013D    340          DATA ACLST9
P   004C  0147    341          DATA ACLS10
P   004E  014E    342          DATA ACLS11
P   0050  0152    343          DATA ACLS12
P   0052  0157    344          DATA ACLS13
P   0054  015F    345          DATA ACLS14
P   0056  0165    346          DATA ACLS15
P   0058  016A    347          DATA ACLS16
P   005A  0171    348          DATA ACLS17
P   005C  0176    349          DATA ACLS18
P   005E  0177    350          DATA ACLS19
P   0060  0180    351          DATA ACLS20
```

```
                    353          PEJ
                    354  *
                    355  *            - ALT 0   : RECORGNIZE OP-CODE
                    356  *
P 0180  0180        357  ACLST0 DATA INIT          INITIALIZATION
P 0181  01A4        358          DATA REOPCO       RECOGNIZE OP-CODE & SET PRED.
P 0182  035C        359          DATA INCWP
  0183  0000        360          DATA 0
                    361  *
                    362  *            - ALT 1   : ALL WORDS PROCESSED
                    363  *
P 0184  0378        364  ACLST1 DATA WLP           WRITE TO LP
  0185  0000        365          DATA 0
                    366  *
                    367  *            - ALT 2   : FORMAT 0
                    368  *
P 0186  023E        369  ACLST2 DATA FORLN         FORMAT ADDR & CONTENT IN DEC,HEX,OCT
P 0187  0271        370          DATA MNE2LN       PLACE MNE STRS IN BUFF, O.P. TO CONS
P 0188  033C        371          DATA SWADV
P 0189  0353        372          DATA CHKALL
  018A  0000        373          DATA 0
                    374  *
                    375  *            - ALT 3   : FORMAT 1
                    376  *
          0015      377  ACLST3 DATA B000F3
E 018B  0000
P 018C  01C5        378          DATA INT2HX       CONVERT BIN. INT. TO HEX
P 018D  023E        379          DATA FORLN        FORMAT ADDR & CONTENT IN DEC,HEX,OCT
P 018E  0271        380          DATA MNE2LN       PLACE MNE STRS IN BUFF, O.P. TO CONS
P 018F  033C        381          DATA SWADV
P 0190  0353        382          DATA CHKALL
  0191  0000        383          DATA 0
                    384  *
                    385  *            - ALT 4   : FORMAT 2
                    386  *
          0016      387  ACLST4 DATA B007FF
E 0192  0000
P 0193  01C5        388          DATA INT2HX       CONVERT BIN. INT. TO HEX
P 0194  023E        389          DATA FORLN        FORMAT ADDR & CONTENT IN DEC,HEX,OCT
P 0195  0271        390          DATA MNE2LN       PLACE MNE STRS IN BUFF, O.P. TO CONS
P 0196  033C        391          DATA SWADV
P 0197  0353        392          DATA CHKALL
  0198  0000        393          DATA 0
                    394  *
                    395  *            - ALT 5   : FORMAT 3
                    396  *
          0017      397  ACLST5 DATA B001FF
E 0199  0000
P 019A  01EA        398          DATA RL2HX        CONVERT REAL BIN. TO HEX
P 019B  023E        399          DATA FORLN        FORMAT ADDR & CONTENT IN DEC,HEX,OCT
P 019C  0271        400          DATA MNE2LN       PLACE MNE STRS IN BUFF, O.P. TO CONS
P 019D  033C        401          DATA SWADV
P 019E  0353        402          DATA CHKALL
  019F  0000        403          DATA 0
                    404  *
                    405  *            - ALT 6   : FORMAT 4
```

```
                    406  *
          0018      407  ACLST6 DATA B3F00
E 0126    0000
P 0127    01EA      408         DATA RL2HX        CONVERT REAL BIN. TO HEX
          0019      409         DATA B00#3F
E 0128    0000
P 0129    01EA      410         DATA RL2HX        CONVERT REAL BIN. TO HEX
P 012A    023E      411         DATA FORLN        FORMAT ADDR & CONTENT IN DEC,HEX,OCT
P 012B    0271      412         DATA MNE2LN       PLACE MNE STRS IN BUFF, O.P. TO CONS
P 012C    033C      413         DATA SWADV
P 012D    0353      414         DATA CHKALL
  012E    0000      415         DATA 0
                    416  *
                    417  *           - ALT 7  : FORMAT 5
                    418  *
          0016      419  ACLST7 DATA B07FF
E 012F    0000
P 0130    01EA      420         DATA RL2HX        CONVERT REAL BIN. TO HEX
P 0131    023E      421         DATA FORLN        FORMAT ADDR & CONTENT IN DEC,HEX,OCT
P 0132    0271      422         DATA MNE2LN       PLACE MNE STRS IN BUFF, O.P. TO CONS
P 0133    033C      423         DATA SWADV
P 0134    0353      424         DATA CHKALL
  0135    0000      425         DATA 0
                    426  *
                    427  *           - ALT 8  : FORMAT 6
                    428  *
          0019      429  ACLST8 DATA B00#3F
E 0136    0000
P 0137    01C5      430         DATA INT2HX       CONVERT BIN. INT. TO HEX
P 0138    023E      431         DATA FORLN        FORMAT ADDR & CONTENT IN DEC,HEX,OCT
P 0139    0271      432         DATA MNE2LN       PLACE MNE STRS IN BUFF, O.P. TO CONS
P 013A    033C      433         DATA SWADV
P 013B    0353      434         DATA CHKALL
  013C    0000      435         DATA 0
                    436  *
                    437  *           - ALT 9  : FORMAT 7
                    438  *
P 013D    02F3      439  ACLST9 DATA SWAPB        SWAP AROUND BYTES IN TEXT INST.
          001C      440         DATA B7F00
E 013E    0000
P 013F    0226      441         DATA ASCII        EXTRACT ASCII
          001D      442         DATA B00#7F
E 0140    0000
P 0141    0226      443         DATA ASCII        EXTRACT ASCII
P 0142    023E      444         DATA FORLN        FORMAT ADDR & CONTENT IN DEC,HEX,OCT
P 0143    02F7      445         DATA WRLWCB       PLACE MNE STRS IN BUFF, O.P. TO CONS
P 0144    033C      446         DATA SWADV
P 0145    0353      447         DATA CHKALL
  0146    0000      448         DATA 0
                    449  *
                    450  *           - ALT 10 : FORMAT 8
                    451  *
          001D      452  ACLS10 DATA B00#7F
E 0147    0000
P 0148    0226      453         DATA ASCII        EXTRACT ASCII
P 0149    023E      454         DATA FORLN        FORMAT ADDR & CONTENT IN DEC,HEX,OCT
```

```
P 014A  0271    455          DATA MNE2LN      PLACE MNE STRS IN BUFF, O.P. TO CONS
P 014B  033C    456          DATA SWADV
P 014C  0353    457          DATA CHKALL
  014D  0000    458          DATA 0
                459 *
                460 *          - ALT 11 : FORMAT 9
                461 *
  014E          462 ACLS11 BSS 0
P 014E  0313    463          DATA ADJPTR      ADJUST INST & ADDR PTRS FOR DW INST.
        001E    464          DATA BFFFF
E 014F  0000
P 0150  0353    465          DATA CHKALL      CHECK ALL WORDS IN BUFFER PROCESSED
  0151  0000    466          DATA 0
                467 *
                468 *          - ALT 12 : FORMAT 10
                469 *
  0152          470 ACLS12 BSS 0
P 0152  0313    471          DATA ADJPTR
        001F    472          DATA B7FFF       PICK UP OPERAND FIELD
E 0153  0000
P 0154  0353    473          DATA CHKALL
P 0155  0375    474          DATA INDIRT      SET PREDICATE FOR INDIRECT ADDR.
  0156  0000    475          DATA 0
                476 *
                477 *          - ALT 13 : FORMAT 11
                478 *
  0157          479 ACLS13 BSS 0
P 0157  0313    480          DATA ADJPTR
        0020    481          DATA B0038
E 0158  0000
P 0159  01C5    482          DATA INT2HX      CONVERT BIN. INT. TO HEX
        0018    483          DATA B00007
E 015A  0000
P 015B  01C5    484          DATA INT2HX      CONVERT BIN. INT. TO HEX
        001A    485          DATA B0FFF
E 015C  0000
P 015D  0353    486          DATA CHKALL
  015E  0000    487          DATA 0
                488 *
                489 *          - ALT 14 : FORMAT 12
                490 *
  015F          491 ACLS14 BSS 0
P 015F  0313    492          DATA ADJPTR
        0015    493          DATA B00003
E 0160  0000
P 0161  01C5    494          DATA INT2HX      CONVERT BIN. INT. TO HEX
        001E    495          DATA BFFFF
E 0162  0000
P 0163  0353    496          DATA CHKALL
  0164  0000    497          DATA 0
                498 *
                499 *          - ALT 15 : FORMAT 13
                500 *
  0165          501 ACLS15 BSS 0
P 0165  0313    502          DATA ADJPTR
        001E    503          DATA BFFFF
```

```
E #166  ####
P #167  #353    5#4          DATA CHKALL
P #168  #26E    5#5          DATA SETP7
  #169  ####    5#6          DATA #
                5#7  *
                5#8  *
                5#9  *            - ALT 16 : FORMAT 14
                51#  *
  #16A          511  ACLS16 BSS  #
P #16A  #313    512          DATA ADJPTR
        ##15    513          DATA B####3
E #16B  ####
P #16C  #1C5    514          DATA INT2HX
        ##1E    515          DATA BF,FF
E #16D  ####
P #16E  #353    516          DATA CHKALL
P #16F  #26E    517          DATA SETP7
  #17#  ####    518          DATA #
                519  *
                52#  *            - ALT 17 : INVALID OP-CODE
                521  *
P #171  #23E    522  ACLS17 DATA FORLN        FORMAT ADDR & CONTENT IN DEC,HEX,OCT
P #172  #38C    523          DATA ILLCDE      OUTPUT 'ILLEGAL OP-CODE'
P #173  #33C    524          DATA SWADV
P #174  #353    525          DATA CHKALL      CHECK IF ALL WORDS PROCESSED
  #175  ####    526          DATA #
                527  *
                528  *            - ALT 18 : 2ND WORD ADDRESS, BIT# INDIRECT
                529  *
  #176          53#  ACLS18 BSS  #
P #176  #231    531          DATA INDBIT
                532  *
                533  *            - ALT 19 : 2ND WORD ADDRESS
                534  *
  #177          535  ACLS19 BSS  #
P #177  #1C#    536          DATA AIN2HX
P #178  #23E    537          DATA FORLN
P #179  #271    538          DATA MNE2LN
P #17A  #35C    539          DATA INCWP
P #17B  #239    54#          DATA AFORLN      FORMAT ADDR & CONTENT OF ADDR WORD
P #17C  #32#    541          DATA OPADVA      OUTPUT ADDR & VALUE
P #17D  #333    542          DATA DWADV
P #17E  #353    543          DATA CHKALL      CHECK IF ALL WORD PROCESSED
  #17F  ####    544          DATA #
                545  *
                546  *            - ALT 2# : 2ND WORD REAL INCREMENT
                547  *
  #18#          548  ACLS2# BSS  #
P #18#  #1E7    549          DATA ARL2HX
P #181  #23E    55#          DATA FORLN
P #182  #271    551          DATA MNE2LN
P #183  #35C    552          DATA INCWP
P #184  #239    553          DATA AFORLN      FORMAT ADDR & CONTENT OF ADDR WORD
P #185  #32#    554          DATA OPADVA      OUTPUT ADDR & VALUE
P #186  #333    555          DATA DWADV
P #187  #353    556          DATA CHKALL      CHECK IF ALL WORD PROCESSED
```

```
  #188  ####    557          DATA #
                558  *
                559  *            -)ALT 21 : 1ST WORD OF DW INST. IS LAST WORD )
                56#  *
P #189  #23E    561  ACLS21  DATA FORLN
P #18A  #271    562          DATA MNE2LN
  #18B  ####    563          DATA #
```

```
                    564           PEJ
                    565   *
                    566   *            ACTION SECTION
                    567   *            **************
                    568   *
                    569   *
                    570   *       --   INITIALIZATION
                    571   *
  0180              572   INIT    BSS  0
  0180  B015        573           DLD  DDBB          BLANK OUT MNEM STRINGS
  0180  A400        574           @DST MNEM1
P 018E  0080
  018F  A111        575           DST  MNEM2
  0190  8113        576           STA  MNEM2+2
  0191  A115        577           DST  MNEM3
  0192  8117        578           STA  MNEM3+2
  0193  A119        579           DST  MNEM4
  0194  8400        580           @STA OPBUF+21
P 0195  0069
                    581   *
                    582   *                        PLACE 'LFCR' AT LAST POSITION
  0196  0000        583           @LDA =LFCR
  0197  8A8D
  0198  8400        584           @STA OPBUF+34
P 0199  0076
  019A  0000        585           @LDA =MNEM1@       INITIALIZE POINTERS
P 019B  0099
  019C  8118        586           STA  MNEPTR
  019D  0000        587           @LDA =MNEM1
P 019E  008C
  019F  810E        588           STA  MNEM@
  01A0  C557        589           RMO  L,P
                    590   *
  01A1              591   A51RET  BSS  1             RETURN
  01A2  A0A0        592   DDBB    DATA '  '
  01A3  A0A0
```

```
                    593          PEJ
                    594  *
                    595  *       --   RECORGNIZE OP-CODE, MNEMONIC
                    596  *            SET PRED FOR FORMAT TYPE
                    597  *
                    598  *       INPUT  : A5CNT,GBUFF@
                    599  *       OUTPUT : MNEM1, P3/.../P12,MNEM@
                    600  *
  01A4              601  REOPCO BSS  0
                    602  *                                RECORGNIZE OP-CODE
  01A4  1400        603          @LDX NOINS              NO. OF INSTRUCTIONS IN REPERTOIRE (-VE)
        0028
E 01A5  0000
  01A6  0524        604  A52.0   LDA  *IGBUF@           GR7 INSTRUCTION
  01A7  3E00        605          AND  *$+1,X             KEEP OP-CODE
        0021        606          DATA GRMASK
E 01A8  0000
  01A9  6E00        607          CPA  *$+1,X
        0022        608          DATA GRINST
E 01AA  0000
  01AB  CDA0        609          SNE                     OP-CODE MATCHED ?
  01AC  7804        610          BRU  A52.2              YES
  01AD  40F8        611          BIX  A52.0              NO - TRY NEXT
  01AE  DB74        612          SMBO P4,SVECTR          NO VALID OP-CODE FOUND
P 01AF  0083
  01B0  C557        613          RMO  L,P                RETURN
                    614  *
                    615  *                               VALID OP-CODE
                    616  *
  01B1  6600        617  A52.2   LDA  *$+1,X             GET FORMAT #
        0023        618          DATA GRFORM
E 01B2  0000
  01B3  CA04        619          CRA  4                  CHANGE FORMAT# TO A5 PRED
  01B4  8106        620          STA  SVECTR
                    621  *                               GET MNEMOMIC INTO MNEM1
  01B5  C0A2        622          RAD  X,X                DOUBLE INDEX
  01B6  B600        623          DLD  *$+1,X             GET MNEMONICS FOR OP-CODE
        0024        624          DATA GRMNE
E 01B7  0000
  01B8  A400        625          @DST MNEM1
P 01B9  008C
  01BA  A50E        626          DST  *MNEM@             PLACE IT IN 1ST STRING
                    627  *
  01BB  511B        628          IMO  MNEPTR             POINT TO NEXT MNEM STRG
  01BC  0518        629          LDA  *MNEPTR
  01BD  810E        630          STA  MNEM@
  01BE  C557        631          RMO  L,P                RETURN
                    632  *
  01BF  DB50        633  SABOIN SABO 0
```

```
                    634          PEJ
                    635  *
                    636  *       --   CONVERT INT BIN VALUE IN ADDR WORD, SPECIFIED
                    637  *            BY 'OPEFLD' INTO HEX ASCII
                    638  *
                    639  *            INPUT  : OPEFLD, AGBUF@, MNEM@
                    640  *            OUTPUT : MNEM(N), MNEM@
                    641  *
  01C0              642  AIN2HX BSS  0
  01C0  0525        643          LDA  *AGBUF@
  01C1  0023        644          STA  VALUE
  01C2  0F00        645          LDE  =0              DO NOT SUPPRESS ZEROS IN ADDRESS
  01C3  0022        646          STE  ZSUPP
  01C4  7804        647          BRU  IN2HX0
                    648  *
                    649  *       --   CONVERT INT BIN VALUE IN INST. WORD SPECIFIED BY 'OPEFLD'
                    650  *            INTO HEX ASCII
                    651  *
                    652  *            INPUT  : OPEFLD, IGBUF@, MNEM@
                    653  *            OUTPUT : MNEM(N), MNEM@
                    654  *
  01C5              655  INT2HX BSS  0
  01C5  0524        656          LDA  *IGBUF@
  01C6  001E        657          STA  VALUE
  01C7  0FFF        658          LDE  =-1             SUPPRESS ZEROS IN OPERAND
  01C8  001D        659          STE  ZSUPP
  01C9  C550        660  IN2HX0 RMO  L,A            SAVE RETURN
  01CA  0019        661          STA  A5.15R
                    662  *
  01CB  0019        663          LDA  VALUE          GET THE CONTENT
  01CC  3C00        664          @AND OPEFLD         KEEP SPECIFIED FIELD
        0014
E 01CD  0000
  01CE  C501        665          RMO  A,E
  01CF  0400        666          @LDA OPEFLD         DETERMINE FIELD POSITION
        0014
E 01D0  0000
  01D1  C90F        667          RTO  15
  01D2  0000        668          LDA  $+1            SHIFT IT TO RIGHT
  01D3  C840        669          LRA  0
  01D4  C0A0        670          RAD  X,A            FORM SHIFT INST.
  01D5  0001        671          STA  $+2
  01D6  C510        672          RMO  E,A            (A)= FIELD
  01D7              673          BSS  1              LRA (X)
  01D8  000D        674          LDE  ZSUPP          SUPPRESS LEADING ZERO(S) OR NOT
                    675  *
  01D9  7400        676          BRL  *$+1           CONVERT BIN INTO HEX ASCII INT
        0001        677          DATA BI2HAS
E 01DA  0000
  01DB  1400        678          @LDX MNEM@          PLACE IN MNE STRG
P 01DC  0008
  01DD  9001        679          STX  MNEM@@
  01DE  A400        680          DST  *MNEM@@        DST *MNEM@
  01DF              681  MNEM@@ BSS  1
                    682  *
  01E0  5118        683          IMO  MNEPTR         POINT TO NEXT MNEM STRG
```

```
GR7 DISASSEMBLER
  01E1  0518    684          LDA   *MNEPTR
  01E2  810E    685          STA   MNEM0
  01E3  7C00    686          BRU   *A5.15R
                687  *
  01E4          688  A5.15R  BSS   1
  01E5          689  VALUE   BSS   1          VALUE OF CURRENT WORD IN BUFFER
  01E6          690  ZSUPP   BSS   1          -1: SUPPRESS ZERO
```

```
                    691         PEJ
                    692   *
                    693   *     --   CONVERT REAL VALUE IN ADDRESS WORD, SPECIFIED
                    694   *          BY 'OPEFLD' INTO HEX ASCII
                    695   *
                    696   *          INPUT  : OPEFLD,AGBUF@,MNEM@
                    697   *          OUTPUT : MNEM(N),MNEM@
                    698   *
  Ø1E7              699   ARL2HX BSS  Ø
  Ø1E7  Ø525        7ØØ          LDA  *AGBUF@
  Ø1E8  8ØFC        7Ø1          STA  VALUE
  Ø1E9  78Ø2        7Ø2          BRU  RL2HXØ
                    7Ø3   *
                    7Ø4   *     --   CONVERT REAL BIN VLAUE IN INST. WORD SPECIFIED BY 'OPEFLD'
                    7Ø5   *          INTO HEX ASCII
                    7Ø6   *
                    7Ø7   *          INPUT  : OPEFLD, GBUFF@, MNEM@
                    7Ø8   *          OUTPUT : MNEM(N), MNEM@
                    7Ø9   *
  Ø1EA              71Ø   RL2HX  BSS  Ø
  Ø1EA  Ø524        711          LDA  *IGBUF@       CONTENT OF INST. WD
  Ø1EB  8ØF9        712          STA  VALUE
  Ø1EC  C55Ø        713   RL2HXØ RMO  L,A           SAVE RETURN
  Ø1ED  8Ø36        714          STA  A5.16R
                    715   *
  Ø1EE  Ø4ØØ        716          @LDA OPEFLD        DETERMINE FIELD POSITION
        ØØ14
E Ø1EF  ØØØØ
  Ø1FØ  C9ØF        717          RTO  15
  Ø1F1  9Ø2B        718          STX  FDPOS
  Ø1F2  DB5F        719          SABO 15
  Ø1F3  C2ØØ        72Ø          RIV  A,A           -VE CONVERSION MASK
  Ø1F4  8Ø29        721          STA  NMASK
  Ø1F5  C2ØØ        722          RIV  A,A           DETERMINE RELATIVE SIGN BIT
  Ø1F6  C98F        723          LTO  15
  Ø1F7  9Ø27        724          STX  RSBIT
                    725   *
  Ø1F8  ØØEC        726          LDA  VALUE
  Ø1F9  3CØØ        727          @AND OPEFLD
        ØØ14
E Ø1FA  ØØØØ
  Ø1FB  C5Ø1        728          RMO  A,E
  Ø1FC  ØØ25        729          LDA  LRA@          'LRA @'
  Ø1FD  2Ø1F        73Ø          ADD  FDPOS         SHIFT COUNT
  Ø1FE  8ØØ1        731          STA  $+2
  Ø1FF  C51Ø        732          RMO  E,A           SHIFT FIELD TO RHE RIGHT
  Ø2ØØ              733          BSS  1             LRA @
                    734   *
  Ø2Ø1  DB6Ø        735          SMBZ Ø,NEG@        ASSUME +VE, BØ=Ø
P Ø2Ø2  Ø221
  Ø2Ø3  C5Ø1        736          RMO  A,E           FIELD RIGHT JUSTIFIED NOW
  Ø2Ø4  ØØ1E        737          LDA  TABO@         'TABO'
  Ø2Ø5  2Ø19        738          ADD  RSBIT         SPECIFY BIT
  Ø2Ø6  8ØØ1        739          STA  $+2
  Ø2Ø7  C51Ø        74Ø          RMO  E,A
  Ø2Ø8              741          BSS  1
```

```
  0209  7805   742          BRU  A5.162      +VE
  020A  DB70   743          SMBO 0,NEG0      -VE
P 020B  0221
  020C  2011   744    *     ADD  NMASK       SET ALL BITS TO THE LEFT TO 1
  020D  C200   745          RIV  A,A         TAKE ITS MAGNITUDE
  020E  2701   746          ADD  =1
  020F  0FFF   747  A5.162  LDE  =-1         SUPPRESS LEADING ZEROS
  0210  7400   748          BRL  *$+1        CONVERT +VE VALUE TO HEX ASCII
        0001   749          DATA BI2HAS
E 0211  0000
  0212  DB30   750          TMBO 0,NEG0      TEST FOR SIGN
P 0213  0221
  0214  7803   751          BRU  A5.164      +VE 0
  0215  1A0A   752          LDX  NEGSGN      -VE 0 - PLACE '-' SIGN
  0216  950E   753          STX  *MNEM0
  0217  510E   754          IMO  MNEM0       POINT TO NEXT WORD WITHIN STRING
  0218  A50E   755  A5.164  DST  *MNEM0      PLACE IN MNE STRG
                756    *
  0219  5118   757          IMO  MNEPTR      POINT TO NEXT STRING
  021A  0518   758          LDA  *MNEPTR
  021B  810E   759          STA  MNEM0
  021C  7C07   760          BRU  *A5.16R     RETURN
                761    *
  021D        1 762  FDPOS  BSS  1           FIELD POSITION (# OF BITS FROM BIT15)
  021E        1 763  NMASK  BSS  1           NEG MASK
  021F        1 764  RSBIT  BSS  1           REL SIGN BIT
  0220  ADA0   765  NEGSGN DATA >ADA0        NEGATIVE SIGN
  0221        1 766  NEG0   BSS  1           B0=1 : NEG
  0222  CB40   767  LRA0   LRA  0
  0223  DB10   768  TABO0  TABO 0
  0224        1 769  A5.16R BSS  1
  0225  DB40   770  SABZ0  SABZ 0
```

```
                    771          PEJ
                    772  *
                    773  *       --   TAKE ASCII FROM FIELD SPECIFIED BY 'OPEFLD'
                    774  *            AND PLACE IT IN APPROPRIATE MNE STRG
                    775  *
                    776  *            INPUT  :OPEFLD,MNEM@,IGBUF@
                    777  *            OUTPUT :MNEM(N)
                    778  *
  #226              779  ASCII  BSS  #
  #226  #524        78#          LDA  *IGBUF@       GET INSTRUCTION
  #227  3C##        781          AND  *OPE@FL       ISOLATE ASCII
        ##14        782  OPE@FL  DATA OPEFLD
E #228  ####
  #229  C5#3        783          RMO  A,M
  #22A  #D#E        784          LDE  *MNEM@        GET MNE STRG
  #22B  #4FC        785          LDA  *OPE@FL       BLANK OUT THE POSITION FIRST
  #22C  C2##        786          RIV  A,A
  #22D  C69#        787          RAN  E,A
  #22E  C#B#        788          RAD  M,A           PLACE ASCII IN
  #22F  85#E        789          STA  *MNEM@        ... THE SAME MNEM STRG
                    79#  *
  #23#  C557        791          RMO  L,P
                    792  *
                    793  *       --   ASSEMBLE INDIRECT BIT
                    794  *
  #231              795  INDBIT BSS  #
  #231  #525        796          LDA  *AGBUF@       (=LDA *GBUFF@)
  #232  DB1#        797          TABO #             INDIRECT BIT = 1 ?
  #233  C557        798          RMO  L,P           NO
  #234  ####        799          LDA  $+1           YES - PLACE '*' IN FRONT OF MNEMONICS
  #235  A#AA        8##          DATA '*'
  #236  84##        8#1          @STA OPBUF+2#
P #237  ##6B
  #238  C557        8#2          RMO  L,P           RETURN
```

```
                    803         PEJ
                    804  *
                    805  *      --   FORMAT ADDRESS & CONTENT OF ADDRESS WORD
                    806  *
                    807  *           INPUT  : IPVAL2,A5CNT,AGBUF0
                    808  *           OUTPUT : CMDLN
                    809  *
  0239  0400        810  AFORLN @LDA AGBUF0
P 023A  00A2
  023B  8400        811          @STA GBUF00
P 023C  026D
  023D  7804        812          BRU  FORLN0
                    813  *
  023E              814  FORLN   BSS  0
  023E  0400        815          @LDA IGBUF0
P 023F  00A1
  0240  8400        816          @STA GBUF00
P 0241  026D
  0242  C550        817  FORLN0 RMO  L,A          SAVE RETURN
  0243  8027        818          STA  A5.19R
                    819  *
                    820  *                       FORMAT ADDRESS
  0244  0400        821          @LDA IPVAL2      CURRENT GR7 ADDR
        0008
E 0245  0000
  0246  2401        822          ADD  *NOWDP0     # OF BYTES PROCESSED
  0247  2400        823          ADD  *NOWDP0
P 0248  00A4        824  NOWDP0  DATA NOWDP
  0249  C700        825          RDE  A,A         ADJUSTMENT
  024A  C700        826          RDE  A,A
  024B  8020        827          STA  ADDRSS
                    828  *
  024C  7400        829          BRL  *$+1        CONVERT TO DEC ADDR
        0002        830          DATA BI2DAS
E 024D  0000
  024E  A400        831          @DST OPBUF+2
P 024F  0056
  0250  9400        832          @STX OPBUF+1
P 0251  0055
                    833  *
  0252  0019        834          LDA  ADDRSS
  0253  7400        835          BRL  *$+1        CONVERT TO HEX ADDR
        0001        836          DATA BI2HAS
E 0254  0000
  0255  A400        837          @DST OPBUF+5
P 0256  0059
                    838  *
  0257  0014        839          LDA  ADDRSS
  0258  7400        840          BRL  *$+1        CONVERT TO OCTAL ASCII
        0003        841          DATA BI2OAS
E 0259  0000
  025A  A400        842          @DST OPBUF+9
P 025B  005D
  025C  9400        843          @STX OPBUF+8
P 025D  005C
                    844  *
```

```
                        845  *                        FORMAT CONTENT
  #25E  #4#E    846           LDA  *GBUF@@
  #25F  74##    847           BRL  *$+1                CONVERT TO HEX ASCII
        ##81    848           DATA BI2HAS
E #26#  ####
  #261  A4##    849           @DST OPBUF+13
P #262  ##61
  #263  #4#9    85#           LDA  *GBUF@@
  #264  74##    851           BRL  *$+1                CONVERT TO OCTAL ASCII
        ###3    852           DATA BI2OAS
E #265  ####
  #266  94##    853           @STX OPBUF+16
P #267  ##64
  #268  A4##    854           @DST OPBUF+17

P #269  ##65
  #26A  7C##    855           BRU  *A5.19R             RETURN
                856  *
  #26B          857  A5.19R BSS  1
  #26C          858  ADDRSS BSS  1                     CURRENT GR7 ADDRESS
  #26D          859  GBUF@@ BSS  1                     EITHER IGBUF@ OR AGBUF@
```

```
                    860           PEJ
                    861  *
                    862  *             SET PREDICATE P7
                    863  *
  #26E              864  SETP7   BSS  #
  #26E   DB77       865          SMBO 7,SVECTR
P #26F   ##83
  #27#   C557       866          RMO  L,P
```

```
                    867          PEJ
                    868  *
                    869  *       --   PLACE MNEMONICS FOR INST. INTO 'CMDLN'
                    870  *            AND OUTPUT TO CONSOLE
                    871  *
                    872  *            INPUT  : MNEM1,...,MNEM4
                    873  *            OUTPUT : CMDLN
                    874  *
  0271              875  MNE2LN BSS   0
  0271  C550       876          RMO   L,A          SAVE RETURN
  0272  0050E      877          STA   MNERET
                    878  *
                    879  *            PACK MNEM2,..,MNEM4 INTO ONE CONTIGUOUS STRING
                    880  *
  0273              881  PCKSTR BSS   0
  0273  B700       882          DLD   =0           RESET POINTERS
  0274  A078       883          DST   TPTR
  0275  C564       884          RMO   B,S
                    885  *
  0276  B400       886          @DLD MNEM1         PLACE OPCODE MNEM IN OP BUFFER
P 0277  008C
  0278  A400       887          @DST OPBUF+21
P 0279  0069
                    888  *
  027A  B400       889          @DLD MNEM2
P 027B  008E
  027C  A400       890          @DST OPBUF+24
P 027D  006C
  027E  B400       891          @DLD MNEM2+2
P 027F  0090
  0280  A400       892          @DST OPBUF+26
P 0281  006E
                    893  *
  0282  B400       894          @DLD MNEM3
P 0283  0092
  0284  A400       895          @DST OPBUF+28
P 0285  0070
  0286  B400       896          @DLD MNEM3+2
P 0287  0094
  0288  A400       897          @DST OPBUF+30
P 0289  0072
                    898  *
  028A  B400       899          @DLD MNEM4
P 028B  0096
  028C  A400       900          @DST OPBUF+32
P 028D  0074
  028E  1800       901          @LDM =OPBUF+24     START OF STRING
P 028F  006C
  0290  7041       902  PCKS00 BRL   GETCHA        GET CHAR. POINTED BY FPTR TO (A)
                    903  *
  0291  6800       904          @CPA =>00A0        BLANK ?
  0292  00A0
  0293  CD20       905          SEQ
  0294  7801       906          BRU   PCKS02       NO  - MOVE IT
  0295  780D       907          BRU   PCKS10       YES
                    908  *
```

```
 #296  8#58   9#9  PCKS#2 STA  CHAR        SAVE CHAR. TO BE MOVED
 #297  B#58   91#        DLD  TPTR        ERASE 'FROM' CHAR.
 #298  C781   911        REX  A,E
 #299  A#56   912        DST  TPTR
 #29A  ####   913        @LDA =>##A#
 #29B  ##A#
 #29C  7#4#   914        BRL  PUTCHA
 #29D  B#52   915        DLD  TPTR
 #29E  C781   916        REX  A,E
 #29F  A#5#   917        DST  TPTR
 #2A#  ##51   918        LDA  CHAR        PLACE CHAR. INTO POSITION
 #2A1  7#3B   919        BRL  PUTCHA
              92#  *
 #2A2  5#4D   921        IMO  TPTR        BUMP TO POINTER
 #2A3  5#4D   922  PCKS1# IMO  FPTR        BUMP FROM POINTER
 #2A4  ##4C   923        LDA  FPTR
 #2A5  6F14   924        CPA  =2#         ALL CHARACTERS PACKED ?
 #2A6  CDC#   925        SLE
 #2A7  78E8   926        BRU  PCKS##      NO
              927  *
 #2A8  B#47   928  L##12  DLD  TPTR        YES - ERASE LAST COMMAS
 #2A9  CC8#   929        SNZ  A
 #2AA  78#F   93#        BRU  L##15       NOTHING TO ERASE
 #2AB  C7##   931        RDE  A,A         POINT TO PREVIOUS CHAR.
 #2AC  C781   932        REX  A,E
 #2AD  A#42   933        DST  TPTR
 #2AE  7#23   934        BRL  GETCHA
 #2AF  68##   935        @CPA =>##AC      ',' ?
 #2B#  ##AC
 #2B1  CD2#   936        SEQ
 #2B2  78#7   937        BRU  L##15       NO  - GO WRITE BUFFER TO CONSOLE
 #2B3  B#3C   938        DLD  TPTR        YES - ERASE ','
 #2B4  C781   939        REX  A,E
 #2B5  A#3A   94#        DST  TPTR
 #2B6  ####   941        @LDA =>##A#
 #2B7  ##A#
 #2B8  7#24   942        BRL  PUTCHA
 #2B9  78EE   943        BRU  L##12
              944  *
 #2BA  C546   945  L##15  RMO  S,B         RESTORE ABL"(B)
 #2BB  DB2F   946        TMBZ 15,LORH
P #2BC  ##A3
 #2BD  78#4   947        BRU  L##17       OUTPUT TO LP SPOOL
 #2BE  18##   948        @LDM =OPCPRB     OUTPUT LINE BUFFER TO CONSOLE
P #2BF  #2CD
 #2C#  CAE#   949        SVC  #
 #2C1  7C#F   95#        BRU  *MNERET
              951  *
 #2C2  18##   952  L##17  @LDM =LPSPRB
P #2C3  #2C6
 #2C4  CAE#   953        SVC  #
 #2C5  7C#8   954        BRU  *MNERET
              955  *
              956  *
              957  *      OUTPUT LINE TO LP SPOOL
 #2C6  #2#3   958  LPSPRB DATA 3###
```

```
   02C7  0002    959          DATA 2
   02C8  0000    960          DATA 0
P  02C9  0054    961          DATA OPBUF
   02CA  0000    962          DATA 0            REC #
   02CB  0000    963          DATA 0            WORD#
   02CC  0023    964          DATA 35           WORD COUNT
                 965  *
                 966  *       OUTPUT LINE TO CONSOLE
   02CD  0000    967  OPCPRB  DATA 0            CONSOLE
   02CE  0002    968          DATA 2            WRITE
   02CF  0046    969          DATA 70           CHAR COUNT
P  02D0  0054    970          DATA OPBUF        BUFFER
                 971  *
                 972  *
   02D1          973  MNERET BSS  1
                 974  *
                 975  *            GET CHAR POINTED BY FPTR
                 976  *
   02D2  C536    977  GETCHA RMO  M,B
   02D3  001D    978          LDA  FPTR         OFFSET IN WORD
   02D4  C841    979          LRA  1
   02D5  C086    980          RAD  A,B
   02D6  1100    981          LDA  0,BR         GET WORD CONTAINING CHAR.
   02D7  DB3F    982          TMBO 15,FPTR      ISOLATE CHAR. IN RH BYTE
P  02D8  02F1
   02D9  C848    983          LRA  8
   02DA  3800    984          @AND =>00FF
   02DB  00FF
   02DC  C557    985          RMO  L,P
                 986  *
                 987  *            PUT CHAR IN (A) TO POS. POINTED BY TPTR
                 988  *
   02DD  C501    989  PUTCHA RMO  A,E            (E)=CHAR.
   02DE  C536    990          RMO  M,B          START OF STRINGS
   02DF  001A    991          LDA  TPTR         OFFSET IN WORD
   02E0  C841    992          LRA  1
   02E1  C086    993          RAD  A,B
   02E2  1100    994          LDA  0,BR         DESTN WORD
   02E3  DB3F    995          TMBO 15,TPTR      INTO WHICH BYTE ?
P  02E4  02F0
   02E5  7804    996          BRU  PCKS05       LH
   02E6  3800    997          @AND =>FF00       RH
   02E7  FF00
   02E8  C081    998          RAD  A,E
   02E9  7804    999          BRU  PCKS08
   02EA  3800    1000 PCKS05 @AND =>00FF
   02EB  00FF
   02EC  CA28    1001         CRE  8
   02ED  C081    1002         RAD  A,E
   02EE  8900    1003 PCKS08 STE  0,BR          PLACE CHAR. IN BYTE
   02EF  C557    1004         RMO  L,P
                 1005 *
   02F0  0000    1006 TPTR   DATA 0             TO POITER
   02F1  0000    1007 FPTR   DATA 0             FROM POINTER
   02F2          1008 CHAR   BSS  1             CHAR TO BE MOVED
```

```
                    1009        PEJ
                    1010  *
                    1011  *     --   SWAP AROUND 2 ASCIIS IN 'TEXT' INSTRUCTION
                    1012  *
  02F3              1013  SWAPB BSS  0
  02F3  0524        1014        LDA  *IGBUF0      GET INSTRUCTION IN BUFFER
  02F4  CA00        1015        CRA  0            SWAP BYTES
  02F5  8524        1016        STA  *IGBUF0
  02F6  C557        1017        RMO  L,P
```

```
                    1#18          PEJ
                    1#19  *
                    1#2#  *       PLACE MNEMONICS INTO 'CMDLN' WITHOUT COMPRESSING
                    1#21  *       OUT BLANK(S) AND WRITE TO CONSOLE OR LP
                    1#22  *       (FOR 'TEXT' INSTRUCTION)
                    1#23  *
                    1#24  *       INPUT  : MNEM1,MNEM2
                    1#25  *       OUTPUT : CMDLN
                    1#26  *
  #2F7              1#27  WRLWCB  BSS  #
  #2F7  C55#        1#28          RMO  L,A          SAVE RETURN
  #2F8  8#D8        1#29          STA  MNERET
                    1#3#  *
  #2F9  B4##        1#31          @DLD DBBB         BLANK OUT MNEM FIELDS IN O/P BUFFER
P #2FA  #331
  #2FB  17F6        1#32          LDX  =-1#
  #2FC  A6##        1#33          @DST OPBUF+34,X
P #2FD  ##76
  #2FE  C322        1#34          RIN  X,X
  #2FF  4#FC        1#35          BIX  $-3
  #3##  B4##        1#36          @DLD MNEM1        PLACE OPCODE MNEM IN OP BUFFER
P #3#1  ##8C
  #3#2  A4##        1#37          @DST OPBUF+21
P #3#3  ##69
                    1#38  *
  #3#4  B4##        1#39          @DLD MNEM2
P #3#5  ##8E
  #3#6  A4##        1#4#          @DST OPBUF+24
P #3#7  ##6C
  #3#8  DB2F        1#41          TMBZ 15,LORH
P #3#9  ##A3
  #3#A  78#4        1#42          BRU  L##LP        OUTPUT TO LP SPOOL
  #3#B  18##        1#43          @LDM =OPCPRB      OUTPUT LINE BUFFER TO CONSOLE
P #3#C  #2CD
  #3#D  CAE#        1#44          SVC  #
  #3#E  7CC2        1#45          BRU  *MNERET
                    1#46  *
  #3#F  18##        1#47  L##LP   @LDM =LPSPRB
P #31#  #2C6
  #311  CAE#        1#48          SVC  #
  #312  7CBE        1#49          BRU  *MNERET
```

```
                    1#5#          PEJ
                    1#51  *
                    1#52  *       --   SET ADDRESS POINTER ONE LOCATION AFTER
                    1#53  *            INSTRUCTION POINTER (FOR 2-WORD INST.)
                    1#54  *
  #313   #124   1#55  ADJPTR LDA  IGBUF@
  #314   C3##   1#56        RIN  A,A
  #315   DB3F   1#57        TMBO 15,LORH      FOR 'L' OR 'H' ?
P #316   ##A3
  #317   78#5   1#58        BRU  L1##         'L'
  #318   C551   1#59        RMO  L,E          'H' - SAVE RETURN
  #319   88#5   1#6#        STE  ADJRET
  #31A   74##   1#61        @BRL BUMPA        ADJUST AGBUF@
P #31B   #3B9
  #31C   7C#2   1#62        BRU  *ADJRET
  #31D   8125   1#63  L1##  STA  AGBUF@
  #31E   C557   1#64        RMO  L,P
                1#65  *
  #31F          1#66  ADJRET BSS  1
```

```
                    1067          PEJ
                    1068  *
                    1069  *       --   OUTPUT ONLY ADDRESS & CONTENT TO CONSOLE
                    1070  *
  0320  B400  1071  OPADVA @DLD  DBBB          BLANK OUT MNEM FIELDS
P 0321  0331
  0322  17F2  1072          LDX  =-14
  0323  A600  1073          @DST OPBUF+34,X
P 0324  0076
  0325  C322  1074          RIN  X,X
  0326  40FC  1075          BIX  $-3
                    1076  *
  0327  D82F  1077          TMBZ 15,LORH      FOR WHICH COMMAND ?
P 0328  00A3
  0329  7803  1078          BRU  L0150        'H'
  032A  1800  1079          @LDM =OPCPRB      'L'
P 032B  02CD
  032C  7802  1080          BRU  L0155
                    1081  *
  032D  1800  1082  L0150  @LDM =LPSPRB
P 032E  02C6
  032F  CAE0  1083  L0155  SVC  0
  0330  C557  1084          RMO  L,P
                    1085  *
  0331  A0A0  1086  DBBB   DATA '  '
  0332  A0A0
```

```
                    1#87          PEJ
                    1#88  *
                    1#89  *       --   ADVANCE BUFFER POINTERS AFTER DW INST. PROCESSED
                    1#9#  *
  #333              1#91  DWADV   BSS  #
  #333  DB3F        1#92          TMBO 15,LORH       FOR 'L' OR 'H' ?
P #334  ##A3
  #335  7813        1#93          BRU  DWADVL        'L'
                    1#94  *
  #336  #124        1#95          LDA  IGBUF@        'H'
  #337  C3##        1#96          RIN  A,A           POINT TO NEXT WORD
  #338  6817        1#97          CPA  OVER          PAGE FULL ?
  #339  CDA#        1#98          SNE
  #33A  ##16        1#99          LDA  SHBUF1        YES - RESET IT TO START OF BUFFER 1
  #33B  8124        11##          STA  IGBUF@
                    11#1  *
                    11#2  *       --   ADVANCE BUFFER POINTERS AFTER SW INST. PROCESSED
                    11#3  *
  #33C              11#4  SWADV   BSS  #
  #33C  DB3F        11#5          TMBO 15,LORH       FOR 'L' OR 'H'
P #33D  ##A3
  #33E  78#C        11#6          BRU  SWADVL        'L'
                    11#7  *
  #33F  #124        11#8          LDA  IGBUR@        'H'
  #34#  C3##        11#9          RIN  A,A           POINT TO NEXT WORD
  #341  68#E        111#          CPA  OVER          PAGE FULL ?
  #342  CDA#        1111          SNE
  #343  ###D        1112          LDA  SHBUF1        YES - RESET IT TO START OF BUFFER 1
  #344  #124        1113          STA  IGBUF@
  #345  C551        1114          RMO  L,E           SAVE RETURN
  #346  88#B        1115          STE  ADVRET
  #347  7#71        1116          BRL  BUMPA         BUMP ADDR PTR
  #348  7C#9        1117          BRU  *ADVRET       RETURN
                    1118  *
  #349  54##        1119  DWADVL  @IMO  IGBUF@
P #34A  ##A1
  #34B  54##        112#  SWADVL  @IMO  IGBUF@
P #34C  ##A1
  #34D  54##        1121          @IMO  AGBUF@
P #34E  ##A2
  #34F  C557        1122          RMO  L,P
                    1123  *
        ##28        1124  OVER    DATA HBUF1+LPERP*2        ONE WORD AFTER LAST WORD IN BUFFER 2
E #35#  ##4C
        ##28        1125  SHBUF1  DATA HBUF1
E #351  ####
  #352              1126  ADVRET  BSS  1             RETURN ADDR
```

```
                1127          PEJ
                1128  *
                1129  *
                1130  *       --    CHECK IF ALL WORDS IN BUFFER PROCESSED
                1131  *
  0353  0400    1132  CHKALL @LDA NOWDP
P 0354  00A4
  0355  6C00    1133         @CPA NOWORD
        0011
E 0356  0000
  0357  CDC0    1134         SLE              ALL WORDS PROCESSED ?
  0358  C557    1135         RMO  L,P
  0359  DB75    1136         SMBO 5,SVECTR    YES - SET PREDICATE
P 035A  0083
  035B  C557    1137         RMO  L,P
                1138  *
                1139  *       --    INCREMENT # OF WORDS PROCESSED
                1140  *
  035C  5400    1141  INCWP  @IMO NOWDP
P 035D  00A4
  035E  DB3F    1142         TMBO 15,LORH     CHECK FOR NEW PAGE FOR 'H'
P 035F  00A3
  0360  C557    1143         RMO  L,P
  0361  0128    1144         LDA  LINENO      LINE # WITHIN PAGE
  0362  C300    1145         RIN  A,A
  0363  6F26    1146         CPA  =38
  0364  CD20    1147         SEQ              START NEW PAGE ?
  0365  780D    1148         BRU  L160        NO
  0366  1800    1149         @LDM =LPRB       YES - LFCR
P 0367  041F
  0368  CAE0    1150         SVC  0
  0369  1800    1151         @LDM =HD1PRL     HEADERS
P 036A  0427
  036B  CAE0    1152         SVC  0
  036C  1800    1153         @LDM =HD2PRL
P 036D  0444
  036E  CAE0    1154         SVC  0
  036F  1800    1155         @LDM =HD3PRL
P 0370  0462
  0371  CAE0    1156         SVC  0
  0372  0700    1157         LDA  =0          RESET LINE #
  0373  8128    1158  L160   STA  LINENO
  0374  C557    1159         RMO  L,P
```

```
900A OFF LINE ASSEMBLER                                          SHEET   36
CR7 DISASSEMBLER
                    1160#          PEJ
                    1161  *
                    1162  *        --  SET PREDICATE FOR INDIRECT ADDRR.
                    1163  *
  0#375             1164  INDIRT BSS  #
    #375  DB76      1165           SMBO 6,SVECTR
P   #376  ##83
    #377  C557      1166           RMO  L,P
```

```
                    1167          PEJ
                    1168  *
                    1169  *       --   WRITE LP SPOOL TO LINE PRINTER IF 'H' COMMAND
                    1170  *
  0378              1171  WLP     BSS  0
  0378  0B3F        1172          TMB0 15,LORH
P 0379  00A3
  037A  C557        1173          RMO  L,P            'L'
  037B  1000        1174          @LDM =PRBC          'H' - CLOSE LP SPOOL
P 037C  0388
  037D  CAE8        1175          SVC  8
  037E  C557        1176          RMO  L,P
                    1177  *
                    1178  *       OPEN LP SPOOL TO LUN 3
  037F  0003        1179  PRBA0   DATA 3              LUN
  0380  0003        1180          DATA >0003          OP-CODE
  0381  CCD0        1181          DATA 'LP'           DEV MNEM.
  0382  0010        1182          DATA >0010          CATEGORY 1
  0383  C6D5        1183          DATA 'FUP/G '       LISTING IDENT.
  0384  D0AF
  0385  C7A0
  0386  0000        1184          DATA 0,0
  0387  0000
                    1185  *
                    1186  *       CLOSE LP SPOOL - OUTPUT LISTING TO LP
  0388  0003        1187  PRBC    DATA 3              LUN
  0389  0004        1188          DATA >0004          OP-CODE
  038A  0000        1189          DATA 0,0
  038B  0000
```

```
                    1190#          PEJ
                    1191  *
                    1192  *       --    OUTPUT 'ILLEGAL OPCODE'
                    1193  *
  #38C              1194  ILLCDE BSS  #
  #38C  B#21        1195          DLD  ASTERK       PLACE '****' IN PALCE OF INST.
  #38D  A4##        1196          @DST OPBUF+21
P #38E  ##69
  #38F  B#2#        1197          DLD  ILLEGL       PLACE 'ILLEGAL OPCODE'
  #39#  A4##        1198          @DST OPBUF+24
P #391  ##6C
  #392  B#1F        1199          DLD  ILLEGL+2
  #393  A4##        12##          @DST OPBUF+26
P #394  ##6E
  #395  B#1E        12#1          DLD  ILLEGL+4
  #396  A4##        12#2          @DST OPBUF+28
P #397  ##7#
  #398  ##1D        12#3          LDA  ILLEGL+6
  #399  84##        12#4          @STA OPBUF+3#
P #39A  #572
  #39B  DB2F        12#5          TMBZ 15,LORH
P #39C  ##A3
  #39D  78#3        12#6          BRU  ILL#3        OUTPUT TO LP SPOOL
  #39E  18##        12#7          @LDM =OPCPRB      OUTPUT LINE BUFFER TO CONSOLE
P #39F  #2CD
  #3A#  78#2        12#8          BRU  ILL#5
  #3A1  18##        12#9  ILL#3   @LDM =LPSPRB
P #3A2  #2C6
  #3A3  CAE#        121#  ILL#5   SVC  #
  #3A4  B#12        1211          DLD  BLANK        ERASE IT IN O/P BUFFER
  #3A5  A4##        1212          @DST OPBUF+24
P #3A6  ##6C
  #3A7  A4##        1213          @DST OPBUF+26
P #3A8  ##6E
  #3A9  A4##        1214          @DST OPBUF+28
P #3AA  ##7#
  #3AB  84##        1215          @STA OPBUF+3#
P #3AC  ##72
  #3AD  C557        1216          RMO  L,P
                    1217  *
  #3AE  ADAD        1218  ASTERK DATA '----'
  #3AF  ADAD
  #3B#  C9CC        1219  ILLEGL DATA 'ILLEGAL OPCODE'
  #3B1  CCC5
  #3B2  C7C1
  #3B3  CCA#
  #3B4  CFD#
  #3B5  C3CF
  #3B6  C4C5
  #3B7  A#A#        122#  BLANK  DATA '    '       BLANKS TO EARSE 'ILLEGAL CODE'
  #3B8  A#A#
```

```
                1221          PEJ
                1222  *
                1223  *       BUMP ADDRESS POINTER - RESET TO START OF HBUF1
                1224  *       BRING NEXT 38 WORDS TO NEXT HBUF IF NECESSARY
                1225  *
  03B9  0125    1226  BUMPA   LDA  AGBUF0
  03BA  C300    1227          RIN  A,A
  03BB  6800    1228          @CPA =HBUF1+38     OVER TO BUFFER 2 ?
        0028
E 03BC  0026
  03BD  CD20    1229          SEQ
  03BE  7813    123#          BRU  L17#
  03BF  8125    1231          STA  AGBUF0        YES - BRING NEXT 38 WORDS INTO HBUF2
                1232  *
                1233  *           BRING NEXT 34 WORDS INTO BUFFER 2
                1234  *
  03C0  0129    1235          LDA  PAGENO
  03C1  9F4C    1236          MPY  =38*2
  03C2  C510    1237          RMO  E,A
  03C3  2123    1238          ADD  GRLOC1        BYTE ADDR FOR NEXT PAGE
  03C4  802A    1239          STA  GI2+1         ...IN 'GI'
  03C5  1800    124#          @LDM =RPRB2
P 03C6  03E8
  03C7  CAE0    1241          SVC  0
  03C8  5129    1242          IMO  PAGENO        BUMP PAGE NO.
  03C9  B400    1243          @DLD BHBUF2        DESTN BYTE ADDR
        002A
E 03CA  0000
  03CB  C503    1244          RMO  A,M           ...IN (M)
  03CC  C514    1245          RMO  E,S           ... & (S)
  03CD  B400    1246          @DLD BHBUF         SRCE BYTE ADDR IN (A),(E)
P 03CE  041D
  03CF  174C    1247          LDX  =38*2         CHAR COUNT
  03D0  DF00    1248          MVC                MOVE CHARACTERS
  03D1  7815    1249          BRU  L21#
                125#  *
  03D2  6800    1251  L17#    @CPA =HBUF2+38
        0029
E 03D3  0026
  03D4  CD20    1252          SEQ
  03D5  780F    1253          BRU  L18#          NO
  03D6  0000    1254          @LDA =HBUF1        RESET TO HBUF1
        0028
E 03D7  0000
  03D8  8125    1255          STA  AGBUF0        YES
                1256  *
                1257  *           BRING NEXT 38 WORDS INTO HBUF1
                1258  *
  03D9  0129    1259          LDA  PAGENO
  03DA  9F4C    126#          MPY  =76
  03DB  C510    1261          RMO  E,A
  03DC  2123    1262          ADD  GRLOC1        BYTE STARTING ADDR FOR NEXT PAGE
  03DD  8400    1263          @STA GI+1
        0027
E 03DE  0001
  03DF  1800    1264          @LDM =RPRB
```

```
          0026
E 03E0    0000
* 03E1    CAE0    1265            SVC   0
  03E2    5129    1266            IMO   PAGENO       INCREMENT PAGE NO.
  03E3    C557    1267            RMO   L,P
  03E4    7802    1268            BRU   L210
                  1269  *
  03E5    8400    1270  L180      @STA  AGBUF@
P 03E6    00A2
  03E7    C557    1271  L210      RMO   L,P
                  1272  *
                  1273  *
  03E8    0002    1274  RPRB2     DATA  2           GR7
  03E9    0001    1275            DATA  1           READ OBJ
  03EA    0006    1276            DATA  6           CHAR COUNT FOR 'GI'
P 03EB    03EE    1277            DATA  GI2
  03EC    0058    1278            DATA  12+38*2     CHAR COUNT
P 03ED    03F1    1279            DATA  RI2
  03EE    4749    1280  GI2       DATA  >4749
  03EF            1281            BSS   1           START BYTE ADDR
  03F0    0026    1282            DATA  38          WORD COUNT
  03F1    5249    1283  RI2       DATA  >5249
  03F2            1284            BSS   1
  03F3            1285            BSS   1
  03F4    0000    1286            DATA  0
  03F5            1287  VL2       BSS   2
  03F7            1288  HBUF      BSS   38
P 041D    0000    1289  BHBUF     BYTE  HBUF        BYTE ADDR OF HBUF
  041E    07EE
```

```
                    1290            PEJ
                    1291  *
  0041F  0203       1292  LPRB      DATA >0203
  00420  0002       1293            DATA 2
  00421  0000       1294            DATA 0
P 00422  00426      1295            DATA NPAGE
  00423  0000       1296            DATA 0
  00424  0000       1297            DATA 0
  00425  0001       1298            DATA 1
  00426  8C8D       1299  NPAGE     DATA >8C8D
                    1300  *
  00427  0203       1301  HD1PRL    DATA >0203          LP SPOOL
  00428  0002       1302            DATA 2
  00429  0000       1303            DATA 0
P 0042A  00432      1304            DATA HD1
  0042B  0000       1305            DATA 0
  0042C  0000       1306            DATA 0
  0042D  0026       1307            DATA 38
  0042E  0000       1308  HD1PRB    DATA 0              0 : CONSOLE
  0042F  0002       1309            DATA 2              WRITE
  00430  0026       1310            DATA 38             CHAR COUNT
P 00431  00432      1311            DATA $+1            BUFFER ADDR
  00432  A0A0       1312  HD1       DATA '   DEC   HEX    OCT     HEX    OCT',LFCR
  00433  A0C4
  00434  C5C3
  00435  A0A0
  00436  A0C8
  00437  C5D8
  00438  A0A0
  00439  A0A0
  0043A  CFC3
  0043B  D4A0
  0043C  A0A0
  0043D  A0A0
  0043E  C8C5
  0043F  D8A0
  00440  A0A0
  00441  A0CF
  00442  C3D4
  00443  8A8D
                    1313  *
  00444  0203       1314  HD2PRL    DATA >0203
  00445  0002       1315            DATA 2
  00446  0000       1316            DATA 0
P 00447  0044F      1317            DATA HD2
  00448  0000       1318            DATA 0
  00449  0000       1319            DATA 0
  0044A  0028       1320            DATA 40
  0044B  0000       1321  HD2PRB    DATA 0              0 : CONSOLE
  0044C  0002       1322            DATA 2              WRITE
  0044D  0028       1323            DATA 40             CHAR COUNT
P 0044E  00450      1324            DATA $+1
  0044F  A0A0       1325  HD2       DATA '   ADDR  ADDR   ADDR    VALUE  VALUE',LFCR
  00450  A0C1
  00451  C4C4
  00452  D2A0
```

```
R7 DISASSEMBLER
   0453  A0C1
   0454  C4C4
   0455  D2A0
   0456  A0A0
   0457  C1C4
   0458  C4D2
   0459  A0A0
   045A  A0A0
   045B  D6C1
   045C  CCD5
   045D  C5A0
   045E  A0D6
   045F  C1CC
   0460  D5C5
   0461  BA8D
                1326  *
   0462  0203   1327  HD3PRL  DATA >0203
   0463  0002   1328         DATA 2
   0464  0000   1329         DATA 0
P  0465  0460   1330         DATA HD3
   0466  0000   1331         DATA 0
   0467  0000   1332         DATA 0
   0468  0028   1333         DATA 40
   0469  0000   1334  HD3PRB  DATA 0              0 : CONSOLE
   046A  0002   1335         DATA 2              WRITE
   046B  0028   1336         DATA 40             CHAR COUNT
P  046C  046D   1337         DATA $+1
   046D  A0A0   1338  HD3    DATA '    ----   ----    ----     -----   -----',LFCR
   046E  A0AD
   046F  ADAD
   0470  ADA0
   0471  A0AD
   0472  ADAD
   0473  ADA0
   0474  A0A0
   0475  ADAD
   0476  ADAD
   0477  A0A0
   0478  A0A0
   0479  ADAD
   047A  ADAD
   047B  ADA0
   047C  A0AD
   047D  ADAD
   047E  ADAD
   047F  BA8D
                1339  *
                1340         END
```

	A	####		A5	##19		A5.15R	#1E4		A5.162	#2#F		A5.164	#218
	A5.16R	#224		A5.19R	#26B		A51.2	##45	R	A51RET	#1A1		A52.#	#1A6
	A52.2	#1B1		A5RETN	##5#		A6	####		ABLENT	####		ACLS1#	#147
	ACLS11	#14E		ACLS12	#152		ACLS13	#157		ACLS14	#15F		ACLS15	#165
	ACLS16	#16A		ACLS17	#171		ACLS18	#176		ACLS19	#177		ACLS2#	#18#
	ACLS21	#189		ACLST#	#1#6		ACLST1	#1#A		ACLST2	#1#C		ACLST3	#111
	ACLST4	#118		ACLST5	#11F		ACLST6	#126		ACLST7	#12F		ACLST8	#136
	ACLST9	#13D	R	ACLST@	##82		ADDRSS	#26C		ADJPTR	#313		ADJRET	#31F
	ADVRET	#352		AFORLN	#239		AGBUF@	##A2		AIN2HX	#1C#		APTR##	##F#
R	APTR#@	##B8		ARL2HX	#1E7		ASCII	#226	R	ASSTR1	###4	R	ASSTR2	###5
R	ASSTR3	###6	R	ASSTR4	###7	R	ASSTR5	###8	R	ASSTR6	###9	R	ASSTR7	###A
	ASTERK	#3AE		B	###6		B###3	##15		B####7	##1B		B##38	##2#
	B##3F	##19		B##7F	##1D		B#1FF	##17		B#7FF	##16		B#FFF	##1A
	B3F##	##1B		B7F##	##1C		B7FFF	##1F		BBBB	#552		BFFFF	##1E
	BHBUF	#41D		BHBUF2	##2A		BI2DAS	###2		BI2HAS	###1		BI2OAS	###3
	BLANK	#3B7		BR	###1	R	BREG	##89		BUMPA	#3B9		CHAR	#2F2
	CHKALL	#353	R	CLSMA@	##B4		CLSMAT	##A8		DBBB	#331		DDBB	#1A2
R	DISAS#	##49		DWADV	#333		DWADVL	#349		E	###1		FDPOS	#21D
	FORLN	#23E		FORLN#	#242		FPTR	#2F1		FSTL	##A7		GBUF@@	#26D
	GBUFF	##12	R	GBUFF@	##13		GETCHA	#2D2		GI	##27		GI2	#3EE
	GRFORM	##23		GRINST	##22		GRLOC1	##A#		GRMASK	##21		GRMNE	##24
	HBUF	#3F7		HBUF1	##28		HBUF2	##29		HD1	#432		HD1PRB	#42E
	HD1PRL	#427		HD2	#44F		HD2PRB	#44B		HD2PRL	#444		HD3	#46D
	HD3PRB	#469		HD3PRL	#462	R	IBUFF	##9E	R	IBUFF@	##9D		IGBUF@	##A1
	ILL#3	#3A1		ILL#5	#3A3		ILLCDE	#38C		ILLEGL	#3B#		IN2HX#	#1C9
	INCWP	#35C		INDBIT	#231		INDIRT	#375		INIT	#18C		INT2HX	#1C5
	IPVAL2	###B	R	IPVAL3	###C	R	IPVAL4	###D	R	IPVAL5	###E	R	IPVAL6	###F
R	IPVAL7	##1#		L	###5		L##12	#2A8		L##15	#2BA		L##17	#2C2
R	L##5	##1A		L##6	##32		L##7	##38		L##LP	#3#F		L#15#	#32D
	L#155	#32F		L1##	#31D		L16#	#373		L17#	#3D2		L18#	#3E5
	L21#	#3E7		LFCR	8A8D		LINENO	##A5		LORH	##A3		LPERP	##26
	LPRB	#41F		LPSPRB	#2C6		LRA@	#222	R	LREG	##8A		M	###3
	MNE2LN	#271		MNEM1	##8C		MNEM1@	##99		MNEM2	##8E		MNEM3	##92
	MNEM4	##96		MNEM@	##8B		MNEM@@	#1DF	R	MNEMS	##98		MNEPTR	##98
	MNERET	#2D1	R	MPRMA@	##87		MPRMAT	##DA	R	NCLMA@	##85		NCLMAT	##AE
	NEG@	#221		NEGSGN	#22#		NMASK	#21E		NOINS	##2B		NOWDP	##A4
	NOWDP@	#248		NOWORD	##11		NPAGE	#426		OP&DVA	#32#		OPBUF	##54
	OPCPRB	#2CD		OPE@FL	#228		OPEFLD	##14		OVER	#35#		P	###7
R	P#	####	R	P1	###1	R	P1#	###A	R	P11	###B	R	P12	###C
R	P13	###D	R	P14	###E	R	P15	###F	R	P2	###2	R	P3	###3
	P4	###4	R	P5	###5	R	P6	###6	R	P7	###7	R	P8	###8
R	P9	###9	R	PACT	###8	R	PACTOF	###1		PAGENO	##A6	R	PALT	##7F
	PCKS##	#29#		PCKS#2	#296		PCKS#5	#2EA		PCKS#8	#2EE		PCKS1#	#2A3
R	PCKSTR	#273	R	PCLUST	##7E		PRBAO	#37F		PRBC	#388		PROGNO	##7D
	PUTCHA	#2DD		REOPCO	#1A4		RI2	#3F1		RL2HX	#1EA		RL2HX#	#1EC
	RPRB	##26		RPRB2	#3E8		RSBIT	#21F		S	###4	R	SABOIN	#1BF
R	SABZ#	#225		SETP7	#26E		SHBUF1	#351		SVC	CAE#		SVECTR	##83
	SWADV	#33C		SWADVL	#34B		SWAPB	#2F3		TABO#	#223		TBTR	#2F#
	VALUE	#1E5	R	VL2	#3F5		WLP	#378		WRLWCB	#2F7		WTC	##25
R	WTC@	##51		X	###2	R	XB	###3	R	YNPMA@	##86		YNPMAT	##C4
	ZSUPP	#1E6												

NO ERRORS NO WARNINGS
V4#15S 19/#3/84 16:#5:18

```
V9#04S    19/03/84    16:05:18

  9900A OFF LINE ASSEMBLER                                    SHEET    1
  DISASSEMBLER - PART 2
                1  *          DISASSEMBLER - PART 2 (ACTIONS)
                2  *          TI990A
                3  *
                4  *          K. VO
                5  *
                6  *          ISSUE 0
                7  *          REVISION 0
                8  *
                9          IDT  V9#
               10          HED  DISASSEMBLER - PART 2
               11  *
               12  *
               13  *          THIS MODULE CONTAINS A NUMBER OF ACTIONS OF
               14  *          DISASSEMBLER PROGRAM.
               15          DEF  OPEFLD,B###3,B07FF,B01FF,B3F##,B##3F,B0FFF,B###7
               16          DEF  B7F##,B##7F,BFFFF,B7FFF,B##38
               17          DEF  GRINST,GRMASK,GRFORM,GRMNE,NOINS
               18  *
               19          REF  ABLENT,IPVAL#
```

```
                    20          PEJ
                    21  *
                    22  *
        0000        23  A       EQU  0
        0001        24  E       EQU  1
        0002        25  X       EQU  2
        0003        26  M       EQU  3
        0004        27  S       EQU  4
        0005        28  L       EQU  5
        0006        29  B       EQU  6
        0007        30  P       EQU  7
                    31  *
        0001        32  BR      EQU  1
        0003        33  XB      EQU  3
        CAE0        34  SVC     OPD  >CAE0,3
        0A0D        35  LFCR    EQU  >0A0D
                    36  *
                    37  *
        0000        38  P0      EQU  0
        0001        39  P1      EQU  1
        0002        40  P2      EQU  2
        0003        41  P3      EQU  3
        0004        42  P4      EQU  4
        0005        43  P5      EQU  5
        0006        44  P6      EQU  6
        0007        45  P7      EQU  7
        0008        46  P8      EQU  8
        0009        47  P9      EQU  9
        000A        48  P10     EQU  10
        000B        49  P11     EQU  11
        000C        50  P12     EQU  12
        000D        51  P13     EQU  13
        000E        52  P14     EQU  14
        000F        53  P15     EQU  15
```

```
                    54          PEJ
                    55  *
                    56  *              DATA SECTION
                    57  *              ************
                    58  *
  0000              59  OPEFLD BSS  1              OPERAND FIELD
P 0001  0002        60  MNEM0  DATA MNEM1          POINTER TO MNE STRG
  0002              61  MNEM1  BSS  2              1ST MNE. STRING (OP-CODE)
  0004              62  MNEM2  BSS  1              2ND  "   "   "
  0005              63  MNEM3  BSS  1              3RD  "   "   "
  0006              64  MNEM4  BSS  1              4TH  "   "   "
  0007              65  MNEMS  BSS  0
                    66  *
  0007              67  IBUFF0 BSS  1              POINTER TO INST. BUFFER
  0008              68  IBUFF  BSS  2              BUFFER OF ASSEMBLED INSTRUCTION
  000A              69  NOWRDS BSS  1              # OF WORDS FOR INSTRUCTION (1 OR 2)
  000B  FFD6        70  NOINS  DATA NOINST         # OF INSTRUCTIONS IN REPERTOIRE
  000C              71  GRLOC1 BSS  1              MEMORY LOCATION 1
```

```
                    72          PEJ
                    73  *
                    74  *           GRAPHIC 7 INSTRUCTION
                    75  *
  0000              76  GRINB   BSS  0
  0000  0000        77          DATA >0000        HALT
  0002  0200        78          DATA >0200        JUMP
  0004  0240        79          DATA >0240        JRMP
  0006  0280        80          DATA >0280        JMPZ
  0008  02C0        81          DATA >02C0        JPRZ
  000A  0400        82          DATA >0400        JMPM
  000C  0440        83          DATA >0440        CALL
  000E  0480        84          DATA >0480        CALR
  0010  04C0        85          DATA >04C0        RTRN
  0012  0600        86          DATA >0600        JZPR
  0014  0800        87          DATA >0800        LINK
  0016  0840        88          DATA >0840        SAVD
  0018  0880        89          DATA >0880        RESV
  001A  08C0        90          DATA >08C0        ADDI
  001C  0A00        91          DATA >0A00        JMPR
  001E  0C00        92          DATA >0C00        LDRI
  0020  0C40        93          DATA >0C40        LDDI
  0022  0C80        94          DATA >0C80        LDSP
  0024  0E00        95          DATA >0E00        WATE
  0026  1000        96          DATA >1000        LDDZ
  0028  1800        97          DATA >1800        LDDP
  002A  2000        98          DATA >2000        LDXA
  002C  2800        99          DATA >2800        LDXR
  002E  3000       100          DATA >3000        DRXA
  0030  3800       101          DATA >3800        DRXR
  0032  4000       102          DATA >4000        DRYA
  0034  4800       103          DATA >4800        DRYR
  0036  5000       104          DATA >5000        MVXA
  0038  5800       105          DATA >5800        MVXR
  003A  6000       106          DATA >6000        MVYA
  003C  6800       107          DATA >6800        MVYR
  003E  7000       108          DATA >7000        LDKX
  0040  7800       109          DATA >7800        LDKY
  0042  8000       110          DATA >8000        DRSR
  0044  8400       111          DATA >8400        MVSR
  0046  C000       112          DATA >C000        PPLR
  0048  C400       113          DATA >C400        LDTI
  004A  9F80       114          DATA >9F80        CHAR
  004C  8080       115          DATA >8080        TEXT
  004E  00E4       116          DATA >00E4        INIT (GR8)
  0050  00DC       117          DATA >00DC        CLRM ( * )
  0052  00E8       118          DATA >00E8        MODE ( * )
  0054             119  GRINST BSS  0
        002A       120  NOINST EQU  GRINB-GRINST # OF INSTRUCTIONS
        0054       121  NOINS2 EQU  NOINST*2
                   122  *
                   123  *           GR7 INSTRUCTION MASKS
                   124  *
  0054  FFC0       125          DATA >FFC0        1  HALT
  0056  FFC0       126          DATA >FFC0
  0058  FFC0       127          DATA >FFC0
```

```
  003A  FFC0   128          DATA >FFC0
  003B  FFC0   129          DATA >FFC0          5  JPRZ
  003C  FFC0   130          DATA >FFC0
  003D  FFC0   131          DATA >FFC0
  003E  FFC0   132          DATA >FFC0
  003F  FFC0   133          DATA >FFC0
  0040  FFC0   134          DATA >FFC0          10 IZPR
  0041  FFC0   135          DATA >FFC0
  0042  FFFC   136          DATA >FFFC
  0043  FFFC   137          DATA >FFFC
  0044  FFFC   138          DATA >FFFC
  0045  FE00   139          DATA >FE00          15 JMPR
  0046  FFC0   140          DATA >FFC0
  0047  FFFC   141          DATA >FFFC
  0048  FFC0   142          DATA >FFC0
  0049  FFC0   143          DATA >FFC0
  004A  F800   144          DATA >F800          20 LDDZ
  004B  F800   145          DATA >F800
  004C  F800   146          DATA >F800
  004D  F800   147          DATA >F800
  004E  F800   148          DATA >F800
  004F  F800   149          DATA >F800          25 DRXR
  0050  F800   150          DATA >F800
  0051  F800   151          DATA >F800
  0052  F800   152          DATA >F800
  0053  F800   153          DATA >F800
  0054  F800   154          DATA >F800          30 MVYA
  0055  F800   155          DATA >F800
  0056  F800   156          DATA >F800
  0057  F800   157          DATA >F800
  0058  C0C0   158          DATA >C0C0
  0059  C0C0   159          DATA >C0C0          35 MVSR
  005A  C0C0   160          DATA >C0C0
  005B  FFC0   161          DATA >FFC0
  005C  FF80   162          DATA >FF80          38 CHAR
  005D  8080   163          DATA >8080          39 TEXT
  005E  FFC0   164          DATA >FFC0          40 INIT (GR8)
  005F  FFC0   165          DATA >FFC0             CLRM ( * )
  0060  FFC0   166          DATA >FFC0             MODE ( * )
  0061         167  GRMASK BSS  0
                    168  *
                    169  *          GR 7 INSTRUCTION FORMATS
                    170  *
  0061  0000   171          DATA 0              HALT
  0062  000A   172          DATA 10
  0063  000D   173          DATA 13
  0064  000A   174          DATA 10
  0065  000D   175          DATA 13
  0066  000A   176          DATA 10
  0067  0009   177          DATA 9
  0068  000D   178          DATA 13
  0069  0000   179          DATA 0
  006A  0000   180          DATA 0              IZPR
  006B  000A   181          DATA 10
  006C  0001   182          DATA 1
  006D  0001   183          DATA 1
```

```
##6E  ###E   184          DATA 14
##6F  ###3   185          DATA 3
##7#  ###B   186          DATA 11
##71  ###C   187          DATA 12
##72  ###9   188          DATA 9
##73  ####   189          DATA #
##74  ###2   19#          DATA 2              LDDZ
##75  ###2   191          DATA 2
##76  ###5   192          DATA 5
##77  ###5   193          DATA 5
##78  ###5   194          DATA 5
##79  ###5   195          DATA 5
##7A  ###5   196          DATA 5
##7B  ###5   197          DATA 5
##7C  ###5   198          DATA 5
##7D  ###5   199          DATA 5
##7E  ###5   2##          DATA 5              MVYA
##7F  ###5   2#1          DATA 5
##8#  ###2   2#2          DATA 2
##81  ###2   2#3          DATA 2
##82  ###4   2#4          DATA 4
##83  ###4   2#5          DATA 4
##84  ###4   2#6          DATA 4
##85  ###6   2#7          DATA 6
##86  ###8   2#8          DATA 8              CHAR
##87  ##7   2#9          DATA 7              TEXT
##88  ####   21#          DATA #              INIT (GR8)
##89  ####   211          DATA #              CLRM ( * )
##8A  ###6   212          DATA 6              MODE ( * )
##8B         213  GRFORM BSS  #
                214  *
                215  *          GR7 INSTRUCTION MNEMONICS
                216  *
##8B  C8C1   217          DATA 'HALT'         1
##8C  CCD4
##8D  CAD5   218          DATA 'JUMP'
##8E  CDD#
##8F  CAD2   219          DATA 'JRMP'
##9#  CDD#
##91  CACD   22#          DATA 'JMPZ'
##92  D#DA
##93  CAD#   221          DATA 'JPRZ'         5
##94  D2DA
##95  CACD   222          DATA 'JMPM'
##96  D#CD
##97  C3C1   223          DATA 'CALL'
##98  CCCC
##99  C3C1   224          DATA 'CALR'
##9A  CCD2
##9B  D2D4   225          DATA 'RTRN'
##9C  D2CE
##9D  C9DA   226          DATA 'IZPR'         1#
##9E  D#D2
##9F  CCC9   227          DATA 'LINK'
##A#  CECB
##A1  D3C1   228          DATA 'SAVD'
```

```
  ØØA2  D6C4
  ØØA3  D2C5   229          DATA 'RESD'
  ØØA4  D3C4
  ØØA5  C1C4   23Ø          DATA 'ADDI'
  ØØA6  C4C9
  ØØA7  CACD   231          DATA 'JMPR'       15
  ØØA8  DØD2
  ØØA9  CCC4   232          DATA 'LDRI'
  ØØAA  D2C9
  ØØAB  CCC4   233          DATA 'LDDI'
  ØØAC  C4C9
  ØØAD  CCC4   234          DATA 'LDSP'
  ØØAE  D3DØ
  ØØAF  D7C1   235          DATA 'WATE'
  ØØBØ  D4C5
  ØØB1  CCC4   236          DATA 'LDDZ'       2Ø
  ØØB2  C4DA
  ØØB3  CCC4   237          DATA 'LDDP'
  ØØB4  C4DØ
  ØØB5  CCC4   238          DATA 'LDXA'
  ØØB6  D8C1
  ØØB7  CCC4   239          DATA 'LDXR'
  ØØB8  D8D2
  ØØB9  C4D2   24Ø          DATA 'DRXA'
  ØØBA  D8C1
  ØØBB  C4D2   241          DATA 'DRXR'       25
  ØØBC  D8D2
  ØØBD  C4D2   242          DATA 'DRYA'
  ØØBE  D9C1
  ØØBF  C4D2   243          DATA 'DRYR'
  ØØCØ  D9D2
  ØØC1  CDD6   244          DATA 'MVXA'
  ØØC2  D8C1
  ØØC3  CDD6   245          DATA 'MVXR'
  ØØC4  D8D2
  ØØC5  CDD6   246          DATA 'MVYA'       3Ø
  ØØC6  D9C1
  ØØC7  CDD6   247          DATA 'MVYR'
  ØØC8  D9D2
  ØØC9  CCC4   248          DATA 'LDKX'
  ØØCA  C8D8
  ØØCB  C4D2   249          DATA 'DRKY'
  ØØCC  C8D9
  ØØCD  C4D2   25Ø          DATA 'DRSR'
  ØØCE  D3D2
  ØØCF  CDD6   251          DATA 'MVSR'       35
  ØØDØ  D3D2
  ØØD1  DØDØ   252          DATA 'PPLR'
  ØØD2  CCD2
  ØØD3  CCC4   253          DATA 'LDTI'
  ØØD4  D4C9
  ØØD5  C3C8   254          DATA 'CHAR'       38
  ØØD6  C1D2
  ØØD7  D4C5   255          DATA 'TEXT'       39
  ØØD8  D8D4
  ØØD9  C9CE   256          DATA 'INIT'       4Ø (GR8)
```

```
9900A OFF LINE ASSEMBLER                                        SHEET    8
DISASSEMBLER - PART 2
  0#DA   C9D4
  0#DB   C3CC    257          DATA 'CLRM'      41 ( * )
  0#DC   D2CD
  0#DD   CDCF    258          DATA 'MODE'      42       ( * )
  0#DE   C4C5
  0#DF           259  GRMNE    BSS  #
```

```
                    260       PEJ
                    261  *
                    262  *
                    263  *       --   PICK UP REG OPERAND BITS 14-15
                    264  *
  0#0DF             265  B0003  BSS  #
  0#0DF  0000       266       @LDA =>0003
  0#0E0  0003
  0#0E1  041E       267       STA  *OPEFL@      OPERAND FIELD
  0#0E2  C557       268       RMO  L,P
                    269  *
                    270  *
                    271  *       A9.4 BITS 5-15
                    272  *
  0#0E3             273  B07FF  BSS  #
  0#0E3  0000       274       @LDA =>07FF
  0#0E4  07FF
  0#0E5  041A       275       STA  *OPEFL@
  0#0E6  C567       276       RMO  L,P
                    277  *
                    278  *
                    279  *       --   BITS 7-15
                    280  *
  0#0E7  0000       281  B01FF  @LDA =>01FF
  0#0E8  01FF
  0#0E9  0416       282       STA  *OPEFL@
  0#0EA  C557       283       RMO  L,P
                    284  *
                    285  *
                    286  *       --   BITS 2-7
                    287  *
  0#0EB  0000       288  03F00  @LDA =>3P00
  0#0EC  3F00
  0#0ED  0412       289       STA  *OPEFL@
  0#0EE  C557       290       RMO  L,P
                    291  *
                    292  *
                    293  *       --   BITS 10-15
                    294  *
  0#0EF  0000       295  B003F  @LDA =>003F
  0#0F0  003F
  0#0F1  0400       296       @STA OPEFLD
P 0#0F2  0000
  0#0F3  C557       297       RMO  L,P
                    298  *
                    299  *
                    300  *       --   BITS BITS 4-15
                    301  *
  0#0F4  0000       302  B0FFF  @LDA =>0FFF
  0#0F5  0FFF
  0#0F6  0409       303       STA  *OPEFL@
  0#0F7  C557       304       RMO  L,P
                    305  *
                    306  *
                    307  *       --   BITS 13-15
                    308  *
```

```
  00F8  0000   309  B0007  @LDA =>0007
  00F9  0007
  00FA  8405   310        STA  *OPEFL@
  00FB  C557   311        RMO  L,P
               312  *
               313  *
               314  *       --   BITS 1-7
               315  *
  00FC  0000   316  B7F00  @LDA =>7F00
  00FD  7F00
  00FE  8401   317        STA  *OPEFL@
  00FF  C557   318        RMO  L,P
P 0100  0000   319  OPEFL@ DATA OPEFLD
               320  *
               321  *
               322  *       --   BITS 9-15
               323  *
  0101  0000   324  B007F  @LDA =>007F
  0102  007F
  0103  84FC   325        STA  *OPEFL@
  0104  C557   326        RMO  L,P
               327  *
               328  *
               329  *       --   BITS 0-15
               330  *
  0105  0000   331  BFFFF  @LDA =>FFFF
  0106  FFFF
  0107  84F8   332        STA  *OPEFL@
  0108  C557   333        RMO  L,P
               334  *
               335  *
               336  *       --   BITS 1-15
               337  *
  0109  0000   338  B7FFF  @LDA =>7FFF
  010A  7FFF
  010B  84F4   339        STA  *OPEFL@
  010C  C557   340        RMO  L,P
               341  *
               342  *
               343  *       --   BITS 10-12
               344  *
  010D  0000   345  B0038  @LDA =>0038
  010E  0038
  010F  84F0   346        STA  *OPEFL@
  0110  C557   347        RMO  L,P
               348  *
               349  *            MAX POSITIVE VALUES
  0111  0001   350  MAXPOS DATA >1          2 BITS FIELD
  0112  0003   351        DATA >3                    3
  0113  0007   352        DATA >7                    4
  0114  000F   353        DATA >F                    5
  0115  001F   354        DATA >1F                   6
  0116  003F   355        DATA >3F                   7
  0117  007F   356        DATA >7F                   8
  0118  00FF   357        DATA >FF                   9
  0119  01FF   358        DATA >1FF                 10
```

```
  #11A   #3FF    359          DATA >3FF                  11
  #11B   #7FF    360          DATA >7FF                  12
  #11C   #FFF    361          DATA >FFF                  13
  #11D   1FFF    362          DATA >1FFF                 14
  #11E   3FFF    363          DATA >3FFF                 15
  #11F   7FFF    364          DATA >7FFF                 16
                 365  *
                 366  *            MIN NEGATIVE VALUES
  #120   FFFF    367  MAXNEG  DATA ->1           2 BITS FIELD
  #121   FFFD    368          DATA ->3                    3
  #122   FFF9    369          DATA ->7                    4
  #123   FFF1    370          DATA ->F                    5
  #124   FFE1    371          DATA ->1F                   6
  #125   FFC1    372          DATA ->3F                   7
  #126   FF81    373          DATA ->8F                   8
  #127   FF01    374          DATA ->FF                   9
  #128   FE01    375          DATA ->1FF                  10
  #129   FC01    376          DATA ->3FF                  11
  #12A   F801    377          DATA ->7FF                  12
  #12B   F001    378          DATA ->FFF                  13
  #12C   E001    379          DATA ->1FFF                 14
  #12D   C001    380          DATA ->3FFF                 15
  #12E   8001    381          DATA ->7FFF                 16
                 382  *
  #12F           383  FLDPOS  BSS  1          FIELD POSITION, # OF BITS FROM RIGHT
P #130   0000    384  IBUF00  DATA IBUFF
                 385  *
                 386          END
```

```
 R A       0000    R ABLENT  0000    R B       0006      B000F  00DF      B0007  00F8
   B0038   010D      B003F   00EF      B07F8   0101      B01FF  00E7      B07FF  00E3
   B0FFF   00F4      B3F00   00EB      B7F00   00FC      B7FFF  0109      BFFFF  0105
 R BR      0001    R E       0001    R FLDPOS  012F      GRFORM 0008      GRINB  000D
   GRINST  0037    R GRLOC1  000C      GRMASK  0061      GRMNE  00DF    R IBUF00 013F
   IBUFF   0008    R IBUFF0  0007    R IPVAL0  0001      L      0005    R LFCR   0A0D
 R M       0003    R MAXNEG  012F    R MAXPOS  0111      MNEM1  0002    R MNEM2  0004
 R MNEM3   0005    R MNEM4   0006    R MNEM0   0001    R MNEMS  0007      NOINS  0008
 R NOINS2  FFAC      NOINST  FFD6    R NOWRDS  000A      OPEFL0 0100      OPEFLD 0000
   P       0007    R P0      0000    R P1      0001    R P10    000A    R P11    000B
 R P12     000C    R P13     000D    R P14     000E    R P15    000F    R P2     0002
 R P3      0003    R P4      0004    R P5      0005    R P6     0006    R P7     0007
 R P8      0008    R P9      0009    R S       0004    R SVC    CAE0    R X      0002
 R XB      0003
```

NO ERRORS NO WARNINGS
V90.04S 19/03/84 16:06:08

```
VD##BS    #7/#3/84    16:#3:32

  98#A ON LINE ASSEMBLER                                    SHEET    1
  ROUTINES
                    1  *          SUBROUTINES
                    2  *          TI98#A
                    3  *
                    4  *          K. VO
                    5  *
                    6  *          ISSUE #
                    7  *          REVISION #
                    8  *
                    9        IDT  VA#
                   1#        HED  ROUTINES
                   11        DEF  BI2DAS,BI2HAS,BI2OAS,WTC,HAS2BI,CC
                   12  *
```

```
                    13          PEJ
                    14  *
                    15  *
0000                16  A       EQU  0
0001                17  E       EQU  1
0002                18  X       EQU  2
0003                19  M       EQU  3
0004                20  S       EQU  4
0005                21  L       EQU  5
0006                22  B       EQU  6
0007                23  P       EQU  7
                    24  *
0001                25  BR      EQU  1
0003                26  XB      EQU  3
CAE0                27  SVC     OPD  >CAE0,3
8A8D                28  LFCR    EQU  >8A8D
```

ROUTINES

```
                    29          PEJ
                    3Ø  *
                    31  *
                    32  *       CONVERT FROM BINARY TO DEC ASCII
                    33  *            LEADING ZERO(S) SUPPRESSED
                    34  *
                    35  *            ENTRY  : (A)=BIN VALUE
                    36  *
                    37  *            RETURN : (X) IST MS DIGI
                    38  *                 ,   (A),(E)=4 LS ASCII DIGITS
                    39  *
                    4Ø  *
 ØØØØ  17FB     41  BI2DAS LDX  =-5
 ØØØ1  ØFØØ     42         LDE  =Ø
 ØØØ2  18ØØ     43         ØLDM =>ØØBØ
 ØØØ3  ØØBØ
                    44  *
 ØØØ4  6229     45  COMP   CPL  DCONST,X     CONVERSION UNIT
 ØØØ5  CDCØ     46         SLE               UNIT .LE. VALUE ?
 ØØØ6  78Ø3     47         BRU  NXUNIT       NO
 ØØØ7  2A26     48         SUB  DCONST,X     YES - SUBSTRACT
 ØØØ8  C311     49         RIN  E,E                AND COUNT
 ØØØ9  78FA     5Ø         BRU  COMP         KEEP GOING
                    51  *
 ØØØA  CØB1     52  NXUNIT RAD  M,E           ASCII DIGIT
 ØØØB  8A1D     53         STE  ASBUF,X
 ØØØC  ØFØØ     54         LDE  =Ø           RESET (E)
 ØØØD  4ØF6     55         BIX  COMP         CONTINUE CONVERSION TO 5 DIGITS
                    56  *                    SUPPRESS LEADING ZERO(S)
 ØØØE  17FB     57         LDX  =-5
 ØØØF  ØØØØ     58         ØLDA =>ØØBØ
 ØØ1Ø  ØØBØ
 ØØ11  6A17     59  SUPZER CPA  ASBUF,X
 ØØ12  CD2Ø     6Ø         SEQ               ZERO ?
 ØØ13  78Ø6     61         BRU  PACK         NO - LEADING NON-ZERO - STOP SUPPR
 ØØ14  CCA2     62         SNO  X            YES - LAST DIGIT ?
 ØØ15  78Ø4     63         BRU  PACK         YES
 ØØ16  Ø8ØØ     64         ØLDE =>ØØAØ       NO - SUPPRESS ZERO
 ØØ17  ØØAØ
 ØØ18  8A1Ø     65         STE  ASBUF,X
 ØØ19  4ØF7     66         BIX  SUPZER
                    67  *
 ØØ1A  1ØØ9     68  PACK   LDX  ASBUF-5      (X)=MS DIGIT
 ØØ1B  ØØØB     69         LDA  ASBUF-2      2 LS DIGITS
 ØØ1C  C8C8     7Ø         LLA  8
 ØØ1D  2ØØA     71         ADD  ASBUF-1
 ØØ1E  C5Ø1     72         RMO  A,E
 ØØ1F  ØØØ5     73         LDA  ASBUF-4      2 MIDDLE DIGITS
 ØØ2Ø  C8C8     74         LLA  8
 ØØ21  2ØØ4     75         ADD  ASBUF-3
 ØØ22  C557     76         RMO  L,P          RETURN
 ØØ23           77  ASBUFR BSS  Ø
 ØØ23           78         BSS  6            6 ASCII DIGITS
 ØØ29           79  ASBUF  BSS  Ø
                    8Ø  *
 ØØ29  271Ø     81         DATA 1ØØØØ
```

```
ROUTINES
  ##2A  #3E8     82          DATA 1###
  ##2B  ##64     83          DATA 1##
  ##2C  ###A     84          DATA 1#
  ##2D  ###1     85          DATA 1
  ##2E           86  DCONST  BSS  #
  ##2E           87  REGB    BSS  1
  ##2F           88  WSRETN  BSS  1
```

```
                    89          PEJ
                    90  *
                    91  *       CONVERT FROM BINARY TO HEX ASCII
                    92  *
                    93  *            ENTRY  : (A)=BIN VALUE
                    94  *                     (E)=-1 LEADING ZERO(S) SUPPRESSED
                    95  *
                    96  *            RETURN : (A)=2 MS ASCII DIGITS
                    97  *                     (E)=2 LS ACIII DIGITS
                    98  *                     (M),(X) DESTROYED
                    99  *
 0030               100 BI2HAS BSS  0
 0030  9833         101        STE  ZSUPPR       SAVE ZERO SUPPRESSION
 0031  17FC         102        LDX  =-4
 0032  0F00         103        LDE  =0
 0033  1800         104        @LDM =>0080
 0034  0080
                    105 *
 0035  622E         106 HCOMP  CPL  HCONST,X     CONVERSION UNIT
 0036  CDC0         107        SLE               UNIT .LE. VALUE ?
 0037  7803         108        BRU  BI00         NO
 0038  2A2B         109        SUB  HCONST,X     YES - SUBSTRACT
 0039  C311         110        RIM  E,E               AND COUNT
 003A  78FA         111        BRU  HCOMP        KEEP GOING
                    112 *
 003B  C790         113 BI00   REX  E,A
 003C  1800         114        @LDM =>0080       ASSUME <= 9
 003D  0080
 003E  6F0A         115        CPA  =>A          DIG OR CHAR ?
 003F  CDC0         116        SLE
 0040  7803         117        BRU  BI05         DIGIT
 0041  2F0A         118        SUB  =>A          CHAR
 0042  1800         119        @LDM =>00C1
 0043  00C1
 0044  C0B0         120 BI05   RAD  M,A          HEX ASCII
 0045  C790         121        REX  E,A
 0046  8AE2         122        STE  ASBUF,X
 0047  0F00         123        LDE  =0           RESET (E)
 0048  40EC         124        BIX  HCOMP        CONTINUE CONVERSION TO 4 DIGITS
                    125 *
                    126 *            SUPPRESS LEADING ZERO(S)
                    127 *
 0049  001A         128        LDA  ZSUPPR       ZERO SUPPRESSION ?
 004A  CC20         129        SOO  A
 004B  780C         130        BRU  BI08         NO
 004C  17FC         131        LDX  =-4          YES
 004D  0000         132        @LDA =>0080
 004E  0080
 004F  6AD9         133 SUPPZE CPA  ASBUF,X      ZERO ?
 0050  CD20         134        SEQ
 0051  7806         135        BRU  BI08         NO - STOP SUPPRESSION
 0052  CCA2         136        SNO  X            YES - LAST DIGIT ?
 0053  7804         137        BRU  BI08         YES - LAST ZERO IS NOT SUPPRESSED
 0054  00A0         138        @LDE =>00A0       NO - SUPPRESS
 0055  00A0
 0056  BAD2         139        STE  ASBUF,X
```

```
0057  40F7   140          BIX   SUPPZE
               141  *
0058  00CE   142  BI00   LDA   ASBUF-2      2 LS DIGITS
0059  C8C8   143          LLA   8
005A  20CD   144          ADD   ASBUF-1
005B  C501   145          RMO   A,E
005C  00C8   146          LDA   ASBUF-4      2 MS DIGITS
005D  C8C8   147          LLA   8
005E  20C7   148          ADD   ASBUF-3
005F  C557   149          RMO   L,P          RETURN
               150  *
0060  1000   151          DATA  >1000
0061  0100   152          DATA  >0100
0062  0010   153          DATA  >0010
0063  0001   154          DATA  >0001
0064         155  HCONST  BSS   0
0064         156  ZSUPPR  BSS   1            -1 : ZERO SUPPR. REQUESTED
```

```
                    157          PEJ
                    158  *
                    159  *       CONVERT FROM BINARY TO OCT ASCII
                    160  *
                    161  *            ENTRY  : (A)=BIN VALUE
                    162  *
                    163  *            RETURN : (X)=2 MS ASCII DIGITS
                    164  *                     (A),(E)=4 LS ASCII DIGITS
                    165  *
                    166  *
 0065  17FA         167  BI2OAS LDX   =-6
 0066  0F00         168          LDE   =0
 0067  1800         169          @LDM  =>00B0
 0068  00B0
                    170  *
 0069  621B         171  OCOMP  CPL   OCONST,X     CONVERSION UNIT
 006A  CDC0         172          SLE              UNIT .LE. VALUE ?
 006B  7803         173          BRU   BIO00      NO
 006C  2A18         174          SUB   OCONST,X   YES - SUBSTRACT
 006D  C311         175          RIN   E,E              AND COUNT
 006E  78FA         176          BRU   OCOMP      KEEP GOING
                    177  *
 006F  C0B1         178  BIO00  RAD   M,E        ASCII DIGIT
 0070  8AB8         179          STE   ASBUF,X
 0071  0F00         180          LDE   =0         RESET (E)
 0072  40F6         181          BIX   OCOMP      CONTINUE CONVERSION TO 5 DIGITS
 0073  00AF         182          LDA   ASBUF-6    2 MS DIGITS
 0074  C8C8         183          LLA   8
 0075  20AE         184          ADD   ASBUF-5
 0076  C502         185          RMO   A,X
 0077  00AF         186          LDA   ASBUF-2    2 LS DIGITS
 0078  C8C8         187          LLA   8
 0079  20AE         188          ADD   ASBUF-1
 007A  C501         189          RMO   A,E
 007B  00A9         190          LDA   ASBUF-4    2 MIDDLE DIGITS
 007C  C8C8         191          LLA   8
 007D  20A8         192          ADD   ASBUF-3
 007E  C557         193          RMO   L',P       RETURN
                    194  *
 007F  8000         195          DATA  >8000
 0080  1000         196          DATA  01000
 0081  0200         197          DATA  00100
 0082  0040         198          DATA  00010
 0083  0008         199          DATA  00001
 0084  0001         200          DATA  00000
 0085               201  OCONST BSS   0
```

```
                    2#2           PEJ
                    2#3   *
                    2#4   *            CONVERT FROM HEX ASCII TO BINARY VALUE
                    2#5   *
                    2#6   *            ENTRY  : (A) POINT TO ASCII STRING
                    2#7   *                     (X)= # OF ASCIIS
                    2#8   *
                    2#9   *            RETURN : (A) = CONVERTED BINARY VALUE
                    21#   *                     (X) = -1 ON UNVALID HEX CHARACTER
                    211   *
 ##85               212   HAS2BI BSS   #
 ##85  8#32         213          STA   TEMPTR        SAVE ASCII POINTER
 ##86  #7##         214          LDA   =#            RESET ASCII COUNTER
 ##87  8#31         215          STA   COUNT
 ##88  8#2E         216          STA   RESULT        CLEAR RESULT
 ##89  C122         217          RCO   X,X           - # OF ASCII
 ##8A  C322         218          RIN   X,X
 ##8B  ##2D         219   HAS##  LDA   COUNT
 ##8C  C841         22#          LRA   1             FORM WORD OFFSET
 ##8D  2#2A         221          ADD   TEMPTR
 ##8E  6##1         222          STA   $+2
 ##8F  #4##         223          LDA   *$+1          GET WORD CONTAINING THE ASCII
 ##9#               224          BSS   1
 ##91  DB3F         225          TMBO  15,COUNT      IF IN LH BYTE
P##92  ##B9
 ##93  C848         226          LRA   8             THEN SHIFT IT TO RH BYTE
 ##94  38##         227          @AND  =>##FF        KEEP THE CURRENT ASCII
 ##95  ##FF
                    228   *
                    229   *            CHECK FOR LEGAL HEX ASCII
                    23#   *
 ##96  68##         231          @CPA  =>##8#        '#' <= ASCII <= '9' ?
 ##97  ##8#
 ##98  CDC#         232          SLE
 ##99  78#7         233          BRU   HAS#1         NO
 ##9A  68##         234          @CPA  =>##B9
 ##9B  ##B9
 ##9C  CD8#         235          SGE
 ##9D  78#3         236          BRU   HAS#1         NO
 ##9E  28##         237          @SUB  =>##8#        YES - GET BINARY VALUE
 ##9F  ##8#
 ##A#  78#8         238          BRU   HAS#2
                    239   *
 ##A1  68##         24#   HAS#1  @CPA  =>##C1        'A' <= ASCII <= 'F' ?
 ##A2  ##C1
 ##A3  CDC#         241          SLE
 ##A4  781#         242          BRU   HAS#4         NO
 ##A5  68##         243          @CPA  =>##C6
 ##A6  ##C6
 ##A7  CD8#         244          SGE
 ##A8  78#C         245          BRU   HAS#4         NO
 ##A9  28##         246          @SUB  =>##C#        YES - GET BINARY VALUE
 ##AA  ##C#
 ##AB  27#9         247          ADD   =9
 ##AC  9AB7         248   HAS#2  MPY   HCONST,X
 ##AD  C8CF         249          LLA   15            TO PLACE RESULT BETWEEN
```

```
ROUTINES
  00AE  C090  250          RAD  E,A        ...0-FFFF INTO (A)
  00AF  2007  251          ADD  RESULT
  00B0  8006  252          STA  RESULT
  00B1  5007  253          IMO  COUNT
  00B2  40D8  254          BIX  HAS00
  00B3  0003  255          LDA  RESULT
  00B4  C557  256          RMO  L,P
              257  *
  00B5  17FF  258  HAS04   LDX  =-1        SET ERROR ON RETURN
  00B6  C557  259          RMO  L,P
              260  *
  00B7        261  RESULT  BSS  1          RESULT OF CONVERSION
  00B8        262  TEMPTR  BSS  1          TEMP POINTER
  00B9        263  COUNT   BSS  1          COUNT OF ASCII
```

```
                    264         PEJ
                    265  *
                    266  *      SUBROUTINE TO COUNT NUMBER OF WORDS IN LINE
                    267  *      A LINE IS TERMINATED ( AND INCLUDING) BY CR
                    268  *      IF CR IS LOCATED IN LH BYTE, A BLANK IS INSERTED BEFORE IT
                    269  *
                    270  *      ENTRY  : (R0)= START ADDR OF BUFFER
                    271  *      RETURN : (A)= # OF WORDS (INCLUDING CR)
                    272  *
  0000              273  CC     BSS  0
  0000  C550        274         RMO  0,A
  0002  8020        275         STA  CCRET
                    276  *
  0004  07FF        277         LDA  =-1          RESET CHAR COUNT
  0006  8019        278         STA  CCNT
                    279  *
  0008  5018        280  CC5    IMO  CCNT
  000A  0017        281         LDA  CCNT
  000C  7018        282         BRL  GETCH        GET CHAR
  000E  3800        283         @AND =>007F
  0010  007F
  0012  6800        284         @CPA =>000D       CR ?
  0014  000D
  0016  CD20        285         SEQ
  0018  78F7        286         BRU  CC5          NO - CHECK NEXT CHAR
                    287  *                        YES
  001A  DB2F        288  CC7    TMBZ 15,CCNT      IS CR IS LH BYTE ?
P 001C  00D7
  001E  7808        289         BRU  CC10         NO
  0020  000C        290         LDA  CCNT         YES - THEN PLACE BLANK BEFORE CR
  0022  0800        291         @LDE =>A00D
  0024  A00D
  0026  C041        292         LRA  1            WD OFFSET FROM START OF BUFFER
  0028  C080        293         RAD  0,A          FORM POINTER
  002A  8008        294         STA  TTPTR
  002C  8C07        295         STE  *TTPTR       PLACE BLANK + CR
  002E  5005        296         IMO  CCNT         TO COUNT THE BLANK
  0030  0004        297  CC10   LDA  CCNT         CHAR COUNT
  0032  2701        298         ADD  =1           ADJUST
  0034  C041        299         LRA  1            WORD COUNT
  0036  7C00        300         BRU  *CCRET       ..ON RETURN
                    301  *
  0038              302  CCRET  BSS  1            RETURN
  003A              303  CCNT   BSS  1            CHAR COUNT
                    304  *
  003C              305  TTPTR  BSS  1            TEMP POINTER
```

```
                    306         PEJ
                    307 *
                    308 *       GET A CHARACTER FROM A STRING
                    309 *
                    310 *       ENTRY  : (M)= START ADDR. OF STRING
                    311 *                (A)= BYTE OFFSET WITHIN STRING
                    312 *
                    313 *       RETURN : (A)= THE CHARACTER, RH-JUSTIFIED
                    314 *           ALL OTHER REGISTERS PRESERVED
                    315 *
  0000              316 GETCH   BSS  0
  0000 8006         317         STA  BOFF           SAVE BYTE OFFSET
  0002 C041         318         LRA  1              CONVERT TO WORD OFFSET
  0004 C000         319         RAD  M,A            FORM POINTER TO CHARACTER
  0006 8008         320         STA  TPTR
  0008 0407         321         LDA  *TPTR          GET THE CHARACTER
  000A DB3F         322         TMBO 15,BOFF        ...IN RH BYTE
P 000C 000E
  000E C048         323         LRA  8
  0010 3000         324         SAND =>00FF
  0012 00FF
  0014 C557         325         RMO  L,P            RETURN
                    326 *
  0016              327 BOFF    BSS  1              BYTE OFFSET
  0017              328 TPTR    BSS  1              TEMPORARY POINTER
```

```
                    329          PEJ
                    33#  *
                    331  *       SUBROUTINE TO WRITE A MESSAGE TO CONSOLE
                    332  *
                    333  *       ENTRY: (M): MESSAGE INDEX - ALL REGS PRESERVED
                    334  *
  ##E6              335  WTC     BSS  #
  ##E6  D8E#        336          @SRF REGF2          SAVE REG FILE
P ##E7  ##FB
  ##E8  C7B2        337          REX  M,X
  ##E9  C564        338          RMO  B,S            SAVE (B)
                    339  *
  ##EA  #218        34#          LDA  M#,X           (A)=MESSAGE STRING ADDRESS
  ##EB  C5#6        341          RMO  A,B             (B) POINTS TO COUNT + MESSAGE
  ##EC  #1##        342          LDA  #,BR           (A)=CHARACTER COUNT
  ##ED  C361        343          RIN  B,E             (E)=START OF MESSAGE
  ##EE  A##A        344          DST  PWTC+2
                    345  *
  ##EF  C532        346          RMO  M,X
                    347  *
  ##F#  18##        348          @LDM =PWTC          CALL SUPERVISOR TO WRITE
P ##F1  ##F7
  ##F2  CAE#        349          SVC  #
  ##F3  C546        35#          RMO  S,B             RESTORE (B)
  ##F4  D8A#        351          @LRF REGF2           RESTORE REG FILE
P ##F5  ##FB
  ##F6  C557        352          RMO  L,P
                    353  *
                    354  *       P.R.B
                    355  *
  ##F7  ####        356  PWTC    DATA #
  ##F8  ###2        357          DATA 2
  ##F9              358          BSS  1
  ##FA              359          BSS  1
                    36#  *
  ##FB              361  REGF2   BSS  8               REGISTER FILE
P #1#3  #114        362  M#      DATA M#,M1,#,#,#,#,#,#,#,#,M1#,M11,M12,#,#,M15,M16
P #1#4  #116
  #1#5  ####
  #1#6  ####
  #1#7  ####
  #1#8  ####
  #1#9  ####
  #1#A  ####
  #1#B  ####
  #1#C  ####
P #1#D  #119
P #1#E  #12#
P #1#F  #125
  #11#  ####
  #111  ####
P #112  #131
P #113  #13F
                    363  *
  #114  ###2        364  M#      DATA 2
  #115  8A8D        365          DATA LFCR
```

```
ROUTINES
  Ø116  ØØØ4   366  M1      DATA 4
  Ø117  AØAØ   367          DATA '  ',LFCR
  Ø118  8A8D
                368  *
  Ø119  ØØØC   369  M1Ø     DATA 12
  Ø11A  C6D5   37Ø          DATA 'FUP/G VØ',LFCR,LFCR
  Ø11B  DØAF
  Ø11C  C7AØ
  Ø11D  D6B8
  Ø11E  8A8D
  Ø11F  8A8D
  Ø12Ø  ØØØ8   371  M11     DATA 8
  Ø121  AAC6   372          DATA '*FUP/G: '
  Ø122  D5DØ
  Ø123  AFC7
  Ø124  8AAØ
  Ø125  ØØ16   373  M12     DATA 22
  Ø126  AØAØ   374          DATA '  **INVALID COMMAND ',LFCR
  Ø127  AAAA
  Ø128  C9CE
  Ø129  D6C1
  Ø12A  CCC9
  Ø12B  C4AØ
  Ø12C  C3CF
  Ø12D  CDCD
  Ø12E  C1CE
  Ø12F  C4AØ
  Ø13Ø  8A8D
  Ø131  ØØ1A   375  M15     DATA 26
  Ø132  AØAØ   376          DATA '  **INVALID ARGUMENT(S) ',LFCR
  Ø133  AAAA
  Ø134  C9CE
  Ø135  D6C1
  Ø136  CCC9
  Ø137  C4AØ
  Ø138  C1D2
  Ø139  C7D5
  Ø13A  CDC5
  Ø13B  CED4
  Ø13C  A8D3
  Ø13D  A9AØ
  Ø13E  8A8D
  Ø13F  ØØ12   377  M16     DATA 18
  Ø14Ø  C6D5   378          DATA 'FUP/G TERMINATES',LFCR
  Ø141  DØAF
  Ø142  C7AØ
  Ø143  D4C5
  Ø144  D2CD
  Ø145  C9CE
  Ø146  C1D4
  Ø147  C5D3
  Ø148  8A8D
                379          END
```

ROUTINES

A	0000	ASBUF	0029	R	ASBUFR	0023	B	0006	BI00	003B
BI05	0044	BI08	0058		BI2DAS	0000	BI2HAS	0030	BI2OAS	0065
BIO00	006F	BOFF	00E4		BR	0001	CC	00BA	CC10	00D2
CC5	00BE	R CC7	00C7		CCNT	00D7	CCRET	00D6	COMP	0004
COUNT	00B9	DCONST	002E		E	0001	GETCH	00D9	HAS00	008B
HAS01	00A1	HAS02	00AC		HAS04	00B5	HAS2BI	0085	HCOMP	0035
HCONST	0064	L	0005		LFCR	0A0D	M	0003	M0	0114
M1	0116	M10	0119		M11	0120	M12	0125	M15	0131
M16	013F	M0	0103		NXUNIT	000A	OCOMP	0069	OCONST	0085
P	0007	PACK	001A		PWTC	00F7	R REGB	002E	REGF2	00FB
RESULT	00B7	S	0004		SUPPZE	004F	SUPZER	0011	SVC	CAE0
TEMPTR	00B8	TPTR	00E5		TTPTR	00D8	R WSRETN	002F	WTC	00E6
X	0002	R XB	0003		ZSUPPR	0064				

NO ERRORS NO WARNINGS

VD0080S 07/03/84 16:04:29