```
1  *    GENERAL AUTOMATION, INC.  ALL RIGHTS RESERVED
2  ***********************************************
3  *
4  *    PROGRAM NAME  FPH-18
5  *
6  *    MODEL NUMBER  8F018
7  *
8  *    PURPOSE       FORTRAN PHASE-18
9  *
10 *    PROGRAMMER    DICK WALLMANN
11 *
12 *****************   REVISION LIST   **************
13 *
14 *    RV DATE     SCO    BY  REASON FOR CHANGE
15 *    -- -------- ------ --- ----------------------
16 *
17 *    01 11/16/70 NONE   RPH INITIAL RELEASE
18 *
19 ***********************************************
20 ***********************************************
21       HDNG    MPX FORTRAN ** EXPANDER II
22 ***********************************************
23 *STATUS-VERSION 1, MODIFICATION 0
24 *
25 *FUNCTION/OPERATION-
26 *    * REPLACES ARITHMETIC, STMNT FUNCTION, CALL,
27 *      AND IF STATEMENTS NOT INVOLVING SUBSCRIPTED
28 *      VARIABLES BY COMPILER-GENERATED CODING
29 *    * COMPLETES THE REPLACEMENT OF ARITHMETIC,
30 *      STMNT FUNCTION, CALL, AND IF STMNTS THAT DO
31 *      INVOLVE SUBSCRIPTED VARIABLES BY COMPILER-
32 *      GENERATED CODING,
33 *    * OPTIMIZES IF STMNT BRANCH INSTRUCTIONS,
34 *    * HANDLES MIXED MODE ARITHMETIC
35 *
36 *ENTRY POINTS-
37 *    * START-PHASE 18 IS READ INTO CORE BY PHASE
38 *            17 VIA ROLRX,  EXECUTION IS BEGUN AT
39 *            LOCATION LABLED START,
40 *
41 *INPUT-
42 *    * THE STATEMENT STRING
43 *    * THE SYMBOL TABLE
44 *    * THE FORTRAN COMMUNICATION AREA
45 *
46 *OUTPUT-
47 *    * THE STATEMENT STRING
48 *    * THE SYMBOL TABLE
49 *    * THE FORTRAN COMMUNICATION AREA
50 *
51 *EXTERNAL REFERENCES-
52 *    * SUBROUTINES
53 *      ROLRX,  ROLR
54 *
55 *EXITS-
56 *    * NORMAL-
57 *          PHASE 20 IS LOADED BY ROLRX
58 *          AND CONTROL PASSED TO IT
59 *    * ERRORS-
```

```
*      OVERLAP-
*          PHASE 20 IS LOADED BY ROLRX
*          AND CONTROL PASSED TO IT
*      SYNTAX-
*          NO SYNTAX ERRORS ARE DETECTED.
*
*TABLES/WORK AREAS-
*    * THE STATEMENT STRING
*    * THE SYMBOL TABLE
*    * THE FORTRAN COMMUNICATION AREA
*
*ATTRIBUTES-N/A
*
*NOTES-
*    THE SWITCHES USED IN PHASE 18 FOLLOW. IF
*    POSITIVE, THE SWITCH IS TRANSFER T, IF ZERO,
*    THE SWITCH IS NORMAL N, IF MINUS, THE SWITCH
*    IS NEGATIVE M.
*       * MODSW-MODE OF ARITHMETIC
*                N SWITCH NOT SET.
*                T FIXED MODE ARITHMETIC
*                M FLOATING MODE ARITHMETIC
*       * STNCH-STATEMENT NUMBER
*                N STMNT HAS NO STMNT NUMBER
*                T STMNT HAS STMNT NUMBER
*       * TEMP -OUTPUT SWITCH
*                T OUTPUT PROCESSED PARTS OF STMNT
*                  WHILE SEARCHING FOR NEXT NOT
*                  PROCESSED OPTR IN SUBROUTINE
*                  FINDN
**************************************************
      HDNG    MPX FORTRAN ** EXPANDER II
      ABS     REF CORE
*
*                          SYSTEM EQUATES
*
MEMRY EQU     77FFF CORE    MAXIMUM CORE SIZE
PHSIZ EQU     4*320         MAXIMUM PHASE SIZE
OVERL EQU     MEMRY-PHSIZ   PHASES 2-29 START
FCOM  EQU     OVERL-22      FORTRAN COMM. TABLE
PHNTB EQU     FCOM-56       PHASE TABLE
ROLRX EQU     PHNTB-50      INTERPHASE CALL
**************************************************
*
*                    FORTRAN COMMUNICATION AREA
*
      ORG     FCOM
SOFS  BSS     1     START OF STRING
EOFS  BSS     1     END OF STRING
SOFST BSS     1     START OF SYMBOL TABLE
SOFNS BSS     1     START OF NON-STATEMENT NUMBERS
SOFXT BSS     1     START OF SUBSCRIPT TEMPORARIES
SOFGT BSS     1     START OF GENERATED TEMPORARIES
EOFST BSS     1     END OF SYMBOL TABLE
COMON BSS     1     NEXT AVAILABLE COMMON
CSIZE BSS     1     SIZE OF COMMON
ERROR BSS     1     OVERLAP ERROR
FNAME BSS     1     PROGRAM NAME
      BSS     1      2ND WORD PROG NAME
SORF  BSS     1     SUBROUTINE -  OR FUNCTION
```

```
120  CCWD  BSS     1     CONTROL CARD WORD
121  *                  BIT 15 TRANSFER TRACE
122  *                  BIT 14 ARITHMETIC TRACE
123  *                  BIT 13 EXTENDED PRECISION
124  *                  BIT 12 LIST SYMBOL TABLE
125  *                  BIT 11 LIST SUBPROGRAM NAMES
126  *                  BIT 10 LIST SOURCE PROGRAM
127  *                  BIT  9 ONE WORD INTEGERS
128  IOCS  BSS     1     IOCS CONTROL CARD WORD
129  *
130  *        SEE PHASE ONE FOR BIT PATTERNS
131  *
132  DFCNT BSS     1          DEFINE FILE COUNT
133  *
134  LCOMN BSS     2          SIZE OF INSKEL COMMON
135  *
136  ICCER BSS     2          IOCS CONTROL CARD ERROR
137  *
138        BSS     2          SYSTEM LOADER USE
139  *
140  *                  END OF FORTRAN COMMUNICATION
141  *                  AREA
142        HDNG      MPX FORTRAN ** EXPANDER II
143        ORG       OVERL      ORIGIN TO FORT OVERLAY PT
144  START LD    L   ERROR      CHECK FOR OVERLAP ERROR
145        BSC   L   EXIT,Z   BRANCH IF YES
146        LD    L   EOFS       GET STRING ENDING ADDR
147        S     L   SOFS       COMPARE TO STARTING ADDR
148        BSC   L   CONT     CONTINUE PROGRAM INITIALIZE
149  EXIT  BSI   L   ROLRX      CALL IN PHASE 20
150        DC        20               NEXT PHASE NUMBER
151  P1011 LDX   I1  SOFIS    INITIALIZE STRING I/P PT
152        LDX   I2  SOFS     INITIALIZE STRING O/P PT
153        MDX    2  -1
154  *
155  *              EXTRACTS THE STMNT ID TYPE
156  *
157  P1021 MDX     1  1          MOVE STRING I/P PT
158        MDX     2  1          MOVE STRING O/P PT
159  P1022 LD      1  0          LOAD STMNT ID WORD
160        AND     3  HF803-Z    ELIMINATE NORM IN ID WORD
161        A       3  FOUR-Z     ADD ONE TO NORM
162        STO     2  0          PUT ON OUTPUT STRING
163        AND     3  HF800-Z    GET ID TYPE
164        STO     3  STTYP-Z    SAVE ID TYPE
165        STX    L2  OUTID      SAVE ADDR OF O/P ID
166        STX    L1  INID       SAVE ADDR OF I/P ID
167        MDX     1  1          MOVE I/P STRING PT
168        LD      1  -1         LOAD STMNT ID WORD
169        BSC   L    P1023,E    BRANCH IF HAVE STMNT NO,
170        MDX        P1031      TEST STMNT TYPE
171  *
172  P1023 LD      3  FOUR-Z     INCREMENT NORM
173        A     I    OUTID      BY ONE
174        STO   I    OUTID      PUT BACK ON OUTPUT STRING
175        MDX     1  1          MOVE I/P PT
176        MDX     2  1          MOVE O/P PT
177        LD      1  -1         LOAD STMNT NO,
178        STO     2  0          PUT ON OUTPUT STRING
179  *              CHECKS FOR A CALL, IF, ARITHMETIC, O
```

```
*                 STMNT FUNCTION STATEMENT
P1031 LD    3 0
      STO   3 MODSW-Z  ZERO MODE SWITCH
*
*                 TEST IF ARITHMETIC STMNT FUNCTION
      LD    3 STTYP-Z  LOAD STMNT ID TYPE
      S     3 HD000-Z  IS IT ARITHMETIC STMNT FUNC
      BSC  L  P1052, - BRANCH IF YES
      LD    3 STTYP-Z  LOAD STMNT ID TYPE
      S     3 READ-Z   SUBTRACT READ CONSTANT
      BSC     Z        SKIP IF READ
      S     3 FIND-Z   TEST FOR FIND STATEMENT
      BSC     Z        SKIP IF FIND OR READ
      S     3 WRITE-Z  TEST FOR WRITE
      BSC  L  HERE,Z   BR IF NOT I/O RD-WRITE-FIND
      LD    1 0        LOAD 2ND WD I/P STRING
      S     3 H0054-Z  IS IT EXPRESSION OPERATOR
      BSC  L  A7A72,Z  BRANCH IF NOT
      MDX   1 1        MOVE I/P STRING PT
      MDX     P2011    BR TO CONTINUE PROCESSING
A7A72 LD    1 0        LOAD WORD
      S     3 LIBFR-Z  IS IT LIBF READ
      BSC     Z        SKIP IF YES
      S     3 H0080-Z  IS IT LIBF WRITE
      BSC     Z        SKIP IF YES
      S     3 LIBFF-Z  IS IT LIBF FIND
      BSC  L  HERE,Z   BRANCH IF NOT
      MDX   1 -1       DECREMENT STRING I/P PT
      BSC  L  D5011    CHECK REST OF STMNT
*
HERE  LD    3 STTYP-Z  LOAD STMNT ID TYPE
      BSC       *      SKIP IF NOT ARITH STMNT
      MDX     P2011    BRANCH IF ARITHMETIC STMNT
      S     3 TCALL-Z  IS IT CALL STMNT
      BSC  L  P2011, - BRANCH IF YES
      S     3 H4800-Z  IS IT IF STMNT
      BSC  L  P2011, - BRANCH IF YES
*
*                 MOVES THE STMNT TO THE O/P STRING
*                 UNALTERED, DETERMINES IF THE LAST
*                 STMNT WAS AN END STMNT
P1041 LDX  I1 INID     RESET I/P PT
      LDX  I2 OUTID    RESET O/P PT
      LD    1 0        LOAD STMNT ID WORD
      AND   3 H07FC-Z  GET STMNT NORM
      STO   3 NORM-Z   SAVE NORM
P1042 LD    1 0        MOVE WORD TO
      STO   2 0        OUTPUT STRING
      MDX  L  NORM,-4  DECREMENT NORM BY ONE
      MDX     P1043    CONTINUE LOOP
*
*                 TEST IF LAST OUTPUT STMNT
*                 WAS END STATEMENT
      LD    3 STTYP-Z  LOAD STMNT ID TYPE
      S     3 TEND-Z   IS IT END STMNT
      BSC  L  P1021,Z  BRANCH IF NOT
*
P1044 STX  L2 EOFS     NEW END OF STRING ADDR
      BSC  L  EXIT     GO TO NEXT PHASE
*
```

```
240   *
241   P1043 MDX    1 1          INCREMENT I/P PT
242         MDX    2 1          INCREMENT O/P PT
243         MDX      P1042      CONTINUE LOOP
244   *
245   *                ARITHMETIC STMNT FUNCTION ENCOUNTERE
246   *                OUTPUT ARITHMETIC STMNT FUNC NAME
247   *                AND DUMMY VARIABLES
248   *
249   P1051 LD     1 0          LOAD WORD
250         BSI    3 OUTUN-Z    PUT ON OUTPUT STRING
251   P1052 MDX    1 1          MOVE I/P PT
252         LD     1 0          LOAD WORD
253         BSC  L P1051, Z     BRANCH IF NAME
254   *
255   *
256   *
257   *                MOVES THE PT PAST THE PROCESSED PART
258   *                OF THE STRING ENTRY, RETAINS THE
259   *                PROCESSED PART IN THE STRING
260   *                UNALTERED, PICKS UP AT THE NEXT
261   *                UNPROCESSED WORD
262   *
263   P2011 MDX    1 -2         MOVE PT
264         LD     3 ZERO-Z     SET OUTPUT SWITCH
265         BSI  L FINDN        FIND NEXT UNPROCESSED OPTR
266   *
267         BSC  L P1021, -     BRANCH IF SEMI-COLON
268   *
269   *                NOT IN CARD SYSTEM
270   *
271         S    L EXPRO        IS IT EXPRESSION-OPERATOR
272         BSC  L D5011, -     BRANCH IF YES
273         LD     1 0          LOAD WORD
274   *
275         S      3 UNARY-Z    IS IT UNARY MINUS
276         BSC  L P6011, -     BRANCH IF YES
277         S      3 CALLX-Z    IS IT CALL
278         BSC  L P5011, -     BRANCH IF YES
279         S      3 IF-Z       IS IT IF
280         BSC  L P3011, -     BRANCH IF YES
281   *
282   *                FIND NEXT NOT-PROCESSED WORD
283   *                STARTING AT PTR 2
284   *
285   P2013 STX    1 * 3        FIND NEXT UNPROCESSED WORD
286         BSI  L FINDN        DECREMENT PT
287         LDX  L1 *-*
288   *
289         BSC  L P2071, Z     BRANCH IF 3RD WORD NOT OPTR
290         LD     1 0          LOAD WORD AT PT
291         S      3 H000E-Z    IS IT ,ASSIGN,
292         BSC  L P2051,Z      BRANCH IF NOT
293         LD     1 1          LOAD NEXT WORD
294         BSI    3 SYMT-Z     GET SYM TBL ID WORD
295         SLA      11         IS IT GENERATED TEMPORARY
296         BSC      -          SKIP IF YES
297         MDX      P2031      BRANCH IF NO
298   *
299   *                MAKE MODE OF GT AGREE WITH MODESW
```

```
300          LD    1 1          LOAD SYM TBL ENTRY NO
301          BSI  L  GTMOD      GET MODE OF NAME
302          MDX     P2033      CLEAR MODE SWITCH
303   *
304   *                GENERATES THE CODE TO HANDLE MIXED-
305   *                MODE ARITHMETIC
306   P2031  LD    3 STNA1-Z    LOAD SYM TBL ID WORD
307          BSC     -          SKIP IF FIXED PT
308          MDX     P2032      BRANCH IF NOT
309          LD    3 MODSW-Z    TEST MODE SWITCH
310          BSC     -          SKIP IF NOT FIX
311          MDX     P2033      BRANCH IF FIX
312          LD    3 IFIX-Z     LOAD ,CALL IFIX,
313          BSI   3 OUTUN-Z    OUTPUT CALL
314          MDX     P2033      CLEAR MODE SWITCH
315   *
316   *
317   P2032  LD    3 MODSW-Z    TEST MODE SWITCH
318          BSC     Z          SKIP IF FIXED
319          MDX     P2033      BRANCH IF NOT FIXED
320          LD    3 FLT-Z      LOAD ,CALL FLOAT,
321          BSI   3 OUTUN-Z    OUTPUT CALL
322   *
323   *                AT THAT POINT MODESWITCH MUST NOT
324   *                BE ZERO  ZERO WOULD MEAN AN ASSIGN-
325   *                OPERATOR PRECEDED BY NOT FILLING ANY
326   *                ACCUMULATOR
327   *                MODESWITCH IS SET ZERO SINCE NEXT
328   *                ARITHMETIC IS INDEPENDENT OF MODE
329   P2033  LD    3 ZERO-Z     CLEAR
330          STO   3 MODSW-Z    MODE SWITCH
331   P2041  LD    1 0          LOAD WORD
332          BSI   3 OUTOP-Z    OUTPUT OPERATOR
333          LD    1 1          LOAD WORD
334          BSI   3 OUTNA-Z    OUTPUT NAME
335          MDX   1 2          MOVE PT
336          MDX     P2011      FIND NEXT UNPROCESSED WORD
337   *
338   *
339   *                TEST IF WORD AT POINTER IS
340   *                ,EXPONETIATE, OPERATOR
341   *                OR ,REVERSE EXPONENTIATE,
342   P2051  LD    1 0          LOAD WORD
343          S     3 EXP-Z      IS IT EXPONENTIATE OPTR
344          BSC  L  P2041, -   BRANCH IF YES
345          S     3 H003E-Z    IS IT REVERSE EXPONENTIATE
346          BSC  L  P2041, -   OPTR - BRANCH IF YES
347   *
348   *                TESTS TO ASCERTAIN PROPER MODE
349          LD    1 1          LOAD WORD
350          BSI   3 SYMT-Z     GET SYM TBL ID WORD
351          BSC     Z          SKIP IF FLOAT
352          MDX     P2052      BRANCH IF FIX
353          LD    3 MODSW-Z    TEST MODE SWITCH
354          BSC      Z         SKIP IF FIX
355          MDX     P2041      BRANCH IF FLOAT
356          LD    3 FLT-Z      LOAD ,CALL FLOAT,
357          BSI   3 OUTUN-Z    OUTPUT CALL
358          LD    3 HF800-Z    SET MODE SWITCH
359          STO   3 MODSW-Z    TO FLOAT
```

```
360          MDX      P2041     OUTPUT OPERATOR
361   *
362   P2052  LD     3 MODSW-Z   TEST MODE SWITCH
363          BSC      -         SKIP IF FLOAT
364          MDX      P2041     BRANCH IF FIXED
365   *
366   *
367   P2061  LD     3 FSTO-Z    LOAD ,CALL FSTO,
368          BSI    3 OUTOP-Z   OUTPUT CALL
369          BSI    3 GETGT-Z   GET GENERATED TEMPORARY
370          BSI  L   GTMOD     MAKE MODE AGREE
371          BSI    3 OUTNA-Z   OUTPUT NAME
372          LD     1 1         LOAD WORD
373          BSI    3 SCKLD-Z   OUTPUT ,LOAD,
374          BSI    3 GETGT-Z   GENERATED TEMPORARY
375          STO    1 1         REPLACES NAME
376          LD     3 FLT-Z     LOAD ,CALL FLT,
377          BSI    3 OUTUN-Z   OUTPUT CALL
378          LD     1 0         LOAD WORD
379          BSI    3 REVOP-Z   REVERSE OPERATOR
380          STO    1 0         STORE REVERSED OPERATOR
381          MDX      P2041     OUTPUT OPERATOR
382   *
383   P2071  LD     3 HF800-Z   SET MODE SWITCH
384          STO    3 MODSW-Z   TO FLOATING PT
385          LD   I   PFIND     LOAD NEXT UNPROCESSED WORD
386          BSI    3 SYMT-Z    GET SYM TBL ID WORD
387          BSC  L   P2081,-   BRANCH IF FLOATING PT
388          LD     1 1         LOAD WORD
389          BSI    3 SYMT-Z    GET SYM TBL ID WORD
390          BSC  L   P2091,-   BRANCH IF FLOATING PT
391          LD     3 ONE-Z     SET MODE SWITCH TO
392          STO    3 MODSW-Z   FIXED PT
393          LD     1 1         OUTPUT FOLLOWING *
394          BSI    3 SCKLD-Z   LD    PT 1
395          MDX      P2111     FIND NEXT UNPROCESSED OPTR
396   *
397   P2081  LD     1 1         LOAD WORD
398          BSI    3 SYMT-Z    GET SYM TBL ID WORD
399          BSC      Z         SKIP IF FLOATING PT
400          MDX      P2101     BRANCH IF FIXED PT
401   *
402   P2083  LD     1 1
403          BSI    3 SCKLD-Z   OUTPUT LOAD INSTRUCTION
404          MDX      P2111     FIND NEXT UNPROCESSED OPTR
405   *
406   P2091  LD     1 0         LOAD WORD
407          S      3 H000E-Z   IS IT ASSIGN
408          BSC      Z         SKIP IF YES
409          MDX      P2093     BRANCH IF NOT
410          LD     1 1         LOAD WORD
411          BSI    3 SCKLD-Z   OUTPUT LOAD INSTRUCTION
412          LD     3 IFIX-Z    CHANGE VAR TO FIXED PT
413          BSI    3 OUTUN-Z   OUTPUT ,CALL IFIX,
414          MDX      P2111     GET NEXT UNPROCESSED OPTR
415   *
416   P2093  A      3 H0004-Z   IS IT EXPRESSION OPERATOR
417          BSC      -         SKIP IF NOT
418          MDX      P2083     BRANCH IF YES
419          LD     1 0
```

```
420          BSI    3 REVOP-Z  REVERSE THE OPERATOR
421          STO    1 0
422          LD     1 2        LOAD WORD
423          S      3 H4800-Z  IS IT ,LDX I1,
424          BSC      Z        SKIP IF YES
425          S      3 H1300-Z  IS IT ,LDX  L1,
426          BSC  L   P2099,Z  BRANCH IF NOT
427          LD     1 1        LOAD VARIABLE
428          BSI    3 SCKLD-Z  OUTPUT LOAD INST AND VAR
429          LD     3 FSTO-Z   LOAD ,CALL FSTO,
430          BSI    3 OUTUN-Z  OUTPUT CALL
431          BSI    3 GETGT-Z  GET ADDR OF NEXT SYM TBL GT
432          BSI  L   GTMOD    MAKE MODE OF VAR AGREE
433          BSI    3 OUTNA-Z  OUTPUT VAR
434          SLA      16       CLEAR ACC
435          BSI  L   FINDN    FIND NEXT UNPROCESSED PART
436          BSI    3 SCKLD-Z  OUTPUT LOAD INST AND VAR
437          LD     3 FLT-Z    LOAD ,CALL FLOAT,
438          BSI    3 OUTUN-Z  OUTPUT CALL
439          LD     1 -4       LOAD WORD
440          BSI    3 OUTUN-Z  OUTPUT
441          BSI    3 GETGT-Z  GET ADDR OF NEXT SYM TBL GT
442          BSI    3 OUTNA-Z  OUTPUT NAME
443          MDX      P3010    GO GET NEXT UNPROCESSED PAR
444  P2099   LD     1 1        LOAD WORD AT PT   1
445          SRT      16       MOVE TO EXTENSION
446          LD   I   PFIND    LOAD WORD AT PT   2
447          STO    1 1        STORE AT PT   1
448          SLT      16       PUT WORD ORIGINALLY AT PT 1
449          STO  I   PFIND    IN PFIND  PT   2
450  P2101   LD     1 1        LOAD WORD
451          BSI    3 SCKLD-Z  OUTPUT LOAD INSTRUCTION
452          LD     3 FLT-Z    CHANGE WORD TO FLOATING PT
453          BSI    3 OUTUN-Z  OUTPUT ,CALL FLT,
454  *
455  *                FIND AND OUTPUT NEXT NONPROCESSED
456  *                OPERATOR, ALSO OUTPUT INTERSPERSED
457  *                PROCESSED DATA, IF ANY
458  *
459  P2111   STX    1 P2112 1  SAVE INPUT PT
460          LD     3 ZERO-Z   SET OUTPUT SWITCH
461          BSI  L   FINDN    FIND NEXT UNPROCESSED OPTR
462  P2112   LD   L   ***      LOAD OPERATOR
463          BSI    3 OUTOP-Z  OUTPUT OPERATOR
464          LD   I   PFIND    LOAD NAME
465          BSI    3 OUTNA-Z  OUTPUT NAME
466  P3010   MDX    1 1        MOVE I/P PT
467  *
468          BSC  L   P2011    GET NEXT UNPROCESSED PART
469  *
470  *                CHECKS FOR AN IF OPTR, GENERATES THE
471  *                TRACING CALLS, CHECKS FOR A STMNT
472  *                NO, ON THE NEXT STMNT, OPTIMIZES THE
473  *                 BSC  L  INST GENERATED TO COMBINE
474  *                CONDITIONS
475  P3011   LD     1 1        LOAD WORD
476          BSI    3 SYMT-Z   GET SYM TBL ID WORD
477          SLA      5        IS IT STMNT NO,
478          BSC  L   P301A, Z BRANCH IF YES
479  *
```

```
480          LD     1 1          LOAD WORD
481          BSI    3 SCKLD-Z    OUTPUT LOAD INSTRUCTION
482          LD     3 ONE-Z      SET MODE SWITCH
483          STO    3 MODSW-Z    TO FIXED POINT
484          LD     3 STNA1-Z    LOAD SYM TBL ID WORD
485          MDX    1 1          MOVE INPUT POINTER
486          BSC  L   P301A,Z    BRANCH IF INTEGER VARIABLE
487          LD     3 HF800-Z    SET MODE SWITCH
488          STO    3 MODSW-Z    TO FLOATING POINT
489   *
490   *
491   P301A  LD     3 ZERO-Z     ZERO TO
492          STO    3 STNCH-Z    CLEAR STMNT NO, CHECK
493          MDX    1 1          MOVE INPUT PT
494          LD     1 4          LOAD NEXT STMNT ID WORD
495          SLT      15         DOES STMNT HAVE STMNT NO,
496          BSC      -          SKIP IF NEXT STMT HAS NUMBE
497          MDX      P3012      BRANCH IF NOT
498          LD     1 5          LOAD STMNT NO,
499          STO    3 STNCH-Z    STORE STMNT NO,
500   P3012  LD     3 MODSW-Z    TEST MODE SWITCH
501          BSC      -          SKIP IF FLOATING POINT
502          MDX      P3021      BRANCH IF FIXED POINT
503          LD     3 LDFAC-Z    LOAD ,LD FAC 1,
504          BSI    3 OUTUN-Z    OUTPUT
505   P3021  LD   L   CCWD
506          SLA      15         IS IF TRACE REQUIRED
507          BSC  L   P3022,-    BRANCH IF NOT
508          LD     3 MODSW-Z    TEST MODE SWITCH
509          BSC  L   P3020,-    BRANCH IF FIXED PT
510          LD     3 FIF-Z      REPLACE ,LD FAC, BY
511          STO    2 0          ,CALL FIF,  FLOATING TRACE
512          MDX      P3022      CHECK BRANCH ADDRESSES
513   P3020  LD     3 FIIF-Z     LOAD INTEGER IF TRACE CALL
514          BSI    3 OUTUN-Z    OUTPUT CALL
515   P3022  LD     1 0          LOAD IF-NEGATIVE BRANCH ADD
516          S      3 STNCH-Z    IS IT NEXT STMNT
517          BSC      -          SKIP IF NOT
518          MDX      P3041      BRANCH IF YES
519          LD     1 0          IS IF-NEGATIVE BRANCH EQUAL
520          S      1 1          TO IF-ZERO BRANCH ADDRESS
521          BSC      Z          SKIP IF YES
522          MDX      P3031      BRANCH IF NOT
523          LD     3 0          LOAD ZERO
524          STO    1 1          STORE IN IF ZERO WORD
525          LD     3 BSCL1-Z    LOAD ,BSC L  , ,
526          MDX      P3033      OUTPUT INSTRUCTION
527   *
528   *
529   P3031  LD     1 0          IS IF-NEGATIVE BRANCH EQUAL
530          S      1 2          TO IF-POSITIVE BRANCH ADDR
531          BSC      Z          SKIP IF YES
532          MDX      P3032      BRANCH IF NOT
533          LD     3 0          LOAD ZERO
534          STO    1 2          STORE IN IF-POSITIVE WORD
535          LD     3 BSCL2-Z    LOAD ,BSC L  ,Z,
536          MDX      P3033      OUTPUT INSTRUCTION
537   P3032  LD     3 BSCL3-Z    LOAD ,BSC L  ,Z ,
538   P3033  BSI    3 OUTOP-Z    OUTPUT BSC INSTRUCTION
539          LD     1 0          LOAD BRANCH TO ADDRESS
```

```
540          BSI    3 OUTNA-Z   OUTPUT ADDRESS
541   P3041  LD     1 1         LOAD ZERO-BRANCH ADDRESS
542          BSC      -         SKIP IF NOT ZERO
543          MDX      P3051     BRANCH IF EQUAL TO NEG-ADDR
544          S      3 STNCH-Z   IS IT NEXT STMNT
545          BSC      -         SKIP IF NOT
546          MDX      P3051     BRANCH IF YES
547          LD     1 1         IS IF-ZERO BRANCH EQUAL TO
548          S      1 2         IF POSITIVE BRANCH ADDRESS
549          BSC      Z         SKIP IF YES
550          MDX      P3042     BRANCH IF NOT
551          LD     3 0         LOAD ZERO
552          STO    1 2         STORE IN IF-POSITIVE WORD
553          LD     3 BSCL4-Z   LOAD ,BSC L  ,-,
554          MDX      P3043     OUTPUT INSTRUCTION
555   *
556   P3042  LD     3 BSCL5-Z   LOAD ,BSC L  , -,
557   P3043  BSI    3 OUTOP-Z   OUTPUT BSC INSTRUCTION
558          LD     1 1         LOAD ZERO-BRANCH ADDRESS
559          BSI    3 OUTNA-Z   OUTPUT ADDRESS
560   P3051  LD     1 2         LOAD POSITIVE BRANCH ADDRES
561          BSC      -         SKIP IF NOT ZERO
562          MDX      P3053     BRANCH IF ZERO
563          S      3 STNCH-Z   IS IT NEXT STMNT
564          BSC      -         SKIP IF NOT
565          MDX      P3053     BRANCH IF YES
566          LD     3 BSCL6-Z   LOAD ,BSC L  ,-Z,
567          BSI    3 OUTOP-Z   OUTPUT INSTRUCTION
568          LD     1 2         LOAD POSITIVE BRANCH ADDR
569          BSI    3 OUTNA-Z   OUTPUT ADDRESS
570   P3053  MDX    1 3         MOVE I/P POINTER
571          BSC  L   P1021     GET NEXT STMNT
572   *
573   *                DETECTS A CALL OPTR, SETS THE MODE
574   *                SWITCH, IF IT IS AN IFIX OR FLOAT
575   *                CALL, CHANGES IT TO A ONE-WD CALL
576   P5011  LD     3 ONE-Z     SET MODE SWITCH
577          STO    3 MODSW-Z   TO FIXED POINT
578          LD     1 1         LOAD WORD
579          BSI    3 SYMT-Z    GET SYM TBL ID WORD
580          BSC      Z         SKIP IF FLOATING POINT
581          MDX      P5012     BRANCH IF FIXED PT
582          LD     3 HF800-Z   SET MODE SWITCH
583          STO    3 MODSW-Z   TO FLOATING POINT
584   *
585   *                TEST IF CALL IS ,IFIX, OR ,FLOAT,
586   *                IF SO, CHANGE TO ONE-WORD CALL
587   *
588   P5012  LD     1 1         LOAD WORD
589          BSI    3 SYMT-Z    GET SYM TBL ID WORD
590          LD     3 SYMT1 1-Z LOAD ADDR OF ID WORD
591          A      3 ONE-Z
592          STO      P501X 1   GET ADDRESS OF NAME IN ENTR
593   P501X  LD   L  *-*         LOAD FIRST WORD OF NAME
594          STO    3 TEMP-Z    STORE
595          MDX  L   P501X 1,1     GET ADDR OF 2ND WORD
596          LD   I   P501X 1   LOAD 2ND WORD
597          STO    3 TEMP-Z 1  STORE
598   *
599   *                TEST IF NAME IS ,FLOAT,
```

```
600           LDD    3 TEMP-Z    LOAD NAME
601           SD     3 NFLT-Z    IS IT ,FLOAT,
602           BSC  L   P5013,Z   BRANCH IF NOT
603           RTE      16        CHECK 2ND WORD
604           BSC  L   P5013,Z   BRANCH NOT ,FLOAT,
605     *
606     *               OUTPUT ,LOAD,, ONEWORD ,FLOAT,
607           LD     3 FLT-Z     LOAD ,CALL FLOAT,
608           MDX      P5015     OUTPUT CALL
609     *
610     *               TEST IF ,IFIX,
611     P5013 LDD    3 TEMP-Z    LOAD NAME
612           SD     3 NIFIX-Z   IS IT ,IFIX,
613           BSC  L   P5016,Z   BRANCH IF NOT
614           RTE      16        CHECK 2ND WORD
615           BSC  L   P5016,Z   BRANCH NOT ,IFIX,
616     *
617     *               OUTPUT ONE-WORD ,IFIX,
618           LD     3 IFIX-Z    LOAD ,CALL IFIX,
619     *
620     *               2-WD CALL ,IFIX, OR ,FLOAT, ENCOUNTR
621     P5015 STO    3 TEMP-Z    STORE CALL
622           LD     1 2         LOAD NEXT WORD
623           BSI    3 SYMT-Z    GET SYM TBL ID WORD
624           SLA      2         ISOLATE DIMENSION
625           SRA      14        INFORMATION
626           BSC  L   P5017, -  BRANCH NOT DIMENSIONED
627           LD     3 H6000-Z   LOAD NOP
628           BSI    3 OUTUN-Z   OUTPUT
629           LD     3 TEMP-Z    LOAD CALL
630           S      3 IFIX-Z    IS IT IFIX
631           BSC      Z         SKIP IF YES
632           LD     3 H8380-Z   CREATE PROPER
633           A      3 FLD-Z     LOAD INSTRUCTION
634           BSI    3 OUTUN-Z   OUTPUT INSTRUCTION
635           LD     3 H6000-Z   LOAD NOP
636           BSI    3 OUTUN-Z   OUTPUT
637           MDX      P5018     OUTPUT ONE-WORD CALL
638     P5017 LD     1 2         LOAD WORD
639           BSI    3 SCKLD-Z   OUTPUT LOAD INSTRUCTION
640     P5018 LD     3 TEMP-Z    LOAD ONE-WORD CALL
641           BSI    3 OUTUN-Z   OUTPUT CALL
642     *
643     *               MAKE ENTRY IN SYM TBL A PSEUDO ENTRY
644     *               TO AVOID LISTING SUBP NAME TWICE
645     *
646           LD     1 1         LOAD WORD
647           BSI    3 SYMT-Z    GET SYM TBL ID WORD
648           BSC  L   * 2,-     BRANCH IF FLOATING POINT
649           LD     3 H4220-Z   LOAD INTEGER STMNT NO, ID
650           MDX      * 1       PUT IN SYMBOL TABLE
651           LD     3 H0220-Z   LOAD REAL STMNT NO, ID
652           STO  I   SYMT1 1   PUT IN SYM TBL
653     *
654           MDX    1 3         MOVE I/P POINTER
655           BSC  L   P2011     GET NEXT UNPROCESSED WORD
656     *
657     *               TWO-WORD CALL OTHER THAN
658     *               ,FLOAT, OR ,IFIX,
659     *
```

```
660  P5016 LD     3 ONE-Z     LOAD OPTR FOR TWO WORD CALL
661        BSI    3 OUTUN-Z   OUTPUT CALL
662        LD     1 1         LOAD WORD
663        BSI    3 OUTUN-Z   OUTPUT
664        MDX    1 2         MOVE PT TO NEXT WORD
665        LD     1 0         LOAD WORD
666        BSC  L P2011,-     BRANCH IF OPERATOR
667  *
668  *               CHECKS FOR THE EXTERNAL SPECIFICATIO
669  *               GENERATES THE PROPER CALL
670  *               OUTPUTS THE COMPLETE CALL AND
671  *               ARGUMENTS
672  P5023 LD     1 0         LOAD WORD
673        BSI    3 SYMT-Z    GET SYM TBL ID WORD
674        RTE      1
675        BSC  L P5024, Z BRANCH IF CONSTANT
676        SLA      11
677        BSC  L P5024,-  BRANCH IF NOT EXTERNAL
678  *
679  *               NAME IS ,EXTERNAL,
680  *               OUTPUT ,CALL, OPERATOR   /0001
681  *
682        LD     3 ONE-Z     LOAD CALL OPERATOR
683        BSI    3 OUTUN-Z   OUTPUT OPERATOR
684  *
685  *
686  *               OUTPUT NAME AT POINTER
687  *
688  P5024 LD     1 0         LOAD NAME
689        BSI    3 OUTUN-Z   OUTPUT NAME
690  *
691  *               TEST IF END OF ARGUMENT LIST
692  *                NEXT WORD AN OPERATOR
693  *
694        MDX    1 1         MOVE I/P PT
695        LD     1 0         LOAD WORD
696        BSC  L P2011,-     BRANCH IF OPERATOR
697        MDX      P5023     BRANCH IF NOT
698  *
699  *
700  *               GENERATES THE CALLS OR INSTRUCTIONS
701  *               TO HANDLE THE UNARY MINUS
702  *
703  P6011 LD     1 1         LOAD WORD
704        BSC      -         SKIP IF NOT OPERATOR
705        MDX      P6024     BRANCH IF OPERATOR
706        MDX    1 1         MOVE I/P POINTER
707        BSI    3 SYMT-Z    GET SYM TBL ID WORD
708        BSC      -         SKIP IF INTEGER NAME
709        MDX      P6021     BRANCH IF REAL NAME
710        LD     3 ONE-Z     SET MODE SWITCH
711        STO    3 MODSW-Z   TO FIXED POINT
712        LD     3 SLA16-Z   LOAD ,SLA  16,
713        BSI    3 OUTUN-Z   OUTPUT INSTRUCTION
714        LD     3 SL-Z      LOAD ,S L,
715        BSI    3 OUTOP-Z   OUTPUT INSTRUCTION
716        LD     1 0         LOAD NAME
717        BSI    3 OUTNA-Z   OUTPUT
718  P6012 MDX    1 1         MOVE I/P PT
719        BSC  L P2011       GET NEXT UNPROCESSED OPER
```

```
720  P6021 LD     3 HF800-Z  SET MODE SWITCH
721        STO    3 MODSW-Z  TO FLOATING POINT
722        LD     3 FLD-Z    LOAD ,FLD,
723        BSI    3 OUTOP-Z  OUTPUT
724        LD     1 0        LOAD NAME
725        BSI    3 OUTNA-Z  OUTPUT
726  P6022 LD       H1C80    LOAD ,CALL RSIGN,
727        BSI    3 OUTUN-Z  OUTPUT
728        MDX      P6012    PROCESS NEXT WORD
729  H1C80 DC       /1C80    ,CALL RSIGN,
730  P6024 LD     3 MODSW-Z  TEST MODE SWITCH
731        BSC               SKIP IF FIXED POINT
732        MDX      P6022    BRANCH IF NOT
733        LD     3 STOFA-Z  LOAD ,STO FAC 1,
734        BSI    3 OUTUN-Z  OUTPUT
735        LD     3 SLA16-Z  LOAD ,SLA  16,
736        BSI    3 OUTUN-Z  OUTPUT
737        LD     3 SFAC-Z   LOAD ,S    FAC 1,
738        BSI    3 OUTUN-Z  OUTPUT
739        MDX      P6012    GET NEXT WORD
740  *
741  *               GENERATES THE CODE FOR A  LD  INST
742  *               IN THE DESIGNATED MODE
743  SCKLD DC         0        ENTRY POINT
744        STO        SCKL3    SAVE ACCUMULATOR
745        BSI    3 SYMT-Z   GET SYM TBL ID WORD
746        BSC  L   SCKL1,*    BRANCH IF REAL VARIABLE
747        LD         SCKL3    LOAD INTEGER
748        OR     3 LDL-Z    COMBINE WITH ,LD L,
749        MDX        SCKL2    OUTPUT INSTRUCTION
750  SCKL1 LD     3 FLD-Z    LOAD ,FLD,
751        BSI    3 OUTUN-Z  OUTPUT
752        LD         SCKL3    LOAD REAL VARIABLE
753  SCKL2 BSI    3 OUTUN-Z  OUTPUT
754        BSC  I   SCKLD    RETURN
755  *
756  *               SUBROUTINE
757  *               INCREMENT NORM IN OUTPUT STATEMENT
758  *               AND INCREMENT OUTPUT POINTER
759  *
760  INCR  DC         0        ENTRY POINT
761  NADR  MDX  L     *-*,4    INCREMENT NORM IN O/P STRIN
762        MDX    2 1          INCREMENT O/P PT
763        BSC  I   INCR     RETURN
764  OUTID EQU        NADR 1
765  *
766  *
767        BSS  E   0
768  TEMP  DC         0        TEMPURARY
769        DC         0        STORAGE
770        BSS  E   0
771  STNA1 DC         0        STORAGE FOR
772        DC         0        PROGRAM NAME
773  TEMP2 DC         0        TEMPURARY STORAGE
774  SCKL3 DC         0        TEMPURARY STORAGE
775  MODSW DC         0        MODE SWITCH
776  STTYP DC         0        STMNT ID TYPE
777  NORM  DC         0        STMNT NORM
778  SOFIS DC         0        START OF I/P STRING - 1
779  STNCH DC         0        STORAGE FOR STMNT NO,
```

```
780  INID   DC       0          INPUT STMNT ID WORD ADDR
781  SSOST  DC       0          STORED START OF SYM TBL
782  GTMO3  DC       0          TEMPORARY
783  *
784  *
785         ORG      TEMP
786  CONT   LDX  L3 ZERO        LOAD INDEX REGISTER 3
787         A     3 ONE-Z       ADD 1 TO GET SIZE OF STRING
788         STO      MOVCT      SAVE SIZE
789         LDX  I1 EOFST       INITIALIZE LOOP O/P PT
790         MDX   1 -10
791         LDX  I2 EOFS        INITIALIZE LOOP I/P PT
792  *
793  *                TEST IF OVERLAP ERROR
794         STX   1 TEMP        SAVE END OF SYM TBL ADDR
795         LD   L  EOFS        LOAD END OF STRING ADDR
796         S        TEMP       SUBTRACT END OF SYM TBL ADD
797         BSC  L  MOVST, Z    BRANCH NO OVERLAP ERROR
798         MDX  L  ERROR,1     SET OVERLAP ERROR INDICATOR
799         BSC  L  EXIT        GO TO NEXT PHASE
800  *
801  *                MOVES THE STRING NEXT TO THE SYM TBL
802  MOVST  LD    2  0           MOVE WORD NEXT
803         STO   1 0           TO SYMBOL TABLE
804         MDX   2 -1          MOVE POINTERS
805         MDX   1 -1
806         MDX  L  MOVCT,-1    SKIP IF FINISHED MOVE
807         MDX      MOVST      CONTINUE LOOP
808  *
809  *
810  *
811  *
812         STX  L1 SOFIS       STORE ADDR OF I/P STRING *
813         LD   L  SOFST       INSERT STORED START
814         STO   3 SSOST-Z     OF SYM TBL ADDR
815         BSC  L  P1011       BRANCH TO MAIN PROGRAM
816  *
817  MOVCT  EQU      TEMP 1     COUNTER
818  *
819  *
820  FAXB   DC       /0000      SYM TBL ADDR OF 2-WORD CALL
821  *                            INITIAL VALUE   ZERO
822  FAXBX  DC       /0000      SYM TBL ADDR OF 2-WORD CALL
823  *                            INITIAL VALUE   ZERO
824  *
825  H000C  DC       /000C      MULTIPLY OPERATOR
826  FDVR   DC       /0780      ,CALL FDVR,
827  DFAC   DC       /1180      ,D  FAC 1,
828  FSBR   DC       /0680      ,CALL FSBR,
829  H0220  DC       /0220      FORMAT STMNT NO, ID
830  FIXI   DC       /0980      ,CALL FIXI,
831  FAXI   DC       /0880      ,CALL FAXI,
832  STOL   DC       /9000      ,STO L,
833  H9000  EQU      STOL
834  ISTOX  DC       /0D80       ,CALL ISTOX,
835  FARIT  DC       /0A80      ,CALL FARIT,
836  IARIT  DC       /0B80      ,CALL IARIT,
837  DL     DC       /A800      ,D L,
838  SRT16  DC       /5800      ,SRT  16,
839  *
```

```
840  *                    THE FOLLOWING 7 CONSTANTS
841  *                    MUST FOLLOW EACH OTHER AND RETAIN
842  *                    THEIR GIVEN ORDER
843  *
844  AL      DC           /9800      'A L'
845  SL      DC           /A000      'S L'
846  FADD    DC           /0080      'CALL FADD'
847  FSUB    DC           /0180      'CALL FSUB'
848  FDIV    DC           /0380      'CALL FDIV'
849          DC           0          NOT USED BUT NEEDED FOR TBL
850  FMPY    DC           /0280      'CALL FMPY'
851  *
852  *
853  H0036   DC           /0036      'REVERSE SUB'
854  HF803   DC           /F803      MASK TO ELIMINATE NORM
855  SFAC    DC           /1100      'S  FAC 1'
856  SLA16   DC           /5900      'SLA 16'
857  STOFA   DC           /1080      'STO  FAC 1'
858  BSCL6   DC           /5030      'BSC L ,Z-'
859  FIIF    DC           /0E80      'CALL FIIF'   FIX IF TRACE
860  FIF     DC           /0F00      'CALL FIF'    FLT IF TRACE
861  LDFAC   DC           /1000      'LD  FAC 1'
862  FSTO    DC           /0580      'CALL FSTO'
863  IFIX    DC           /0C80      'CALL IFIX'
864  LDL     DC           /8800      'LD L'
865  H8800   EQU          LDL
866  FLD     DC           /0480      'CALL FLD'
867  FLT     DC           /0D00      'CALL FLOAT'
868  *
869  *                    FOLLOWING TWO ENTRIES ARE USED
870  *                    AS DOUBLE-LENGTH WORDS
871  *
872          BSS  E       0
873  NFLT    DC           /8C9A      NAME
874          DC           /E063      'FLOAT'
875  NIFIX   DC           /9231      NAME
876          DC           /99C0      'IFIX'
877  *
878  H3000   DC           /3000      MASK FOR DIMENSION TEST
879  ZERO    DC           0          ZERO                    00
880  Z       EQU          ZERO
881  ONE     DC           1          ONE
882  TWO     DC           2          TWO
883  THREE   DC           3          THREE
884  FOUR    DC           /0004      FOUR
885  H0004   EQU          FOUR
886  HF800   DC           /F800      ID TYPE MASK
887  TCALL   DC           /3000      CALL ID TYPE
888  H4800   DC           /4800      'LDX I1'
889  TEND    DC           /1000      END ID TYPE
890  H000E   DC           /000E      ASSIGN OPERATOR
891  H800C   DC           /800C      VARIABLE TEST MASK
892  *                    BITS FOR CONST,GENTEMP,SPECGENT
893  H07FC   DC           /07FC      MASK TO GET NORM
894  BSCL1   DC           /5008      'BSC L , '
895  BSCL2   DC           /5020      'BSC L ,Z'
896  BSCL3   DC           /5028      'BSC L ,Z '
897  BSCL4   DC           /5010      'BSC L ,-'
898  BSCL5   DC           /5018      'BSC L , -'
899  H07FF   DC           /07FF      MASK TO GET SYM TBL ADDR
```

```
900   H8000 DC       /8000     SIGN BIT MASK
901   EXP   DC       /000A     EXPONENTE OPERATOR
902   HD000 DC       /D000     ARITHMETIC STMNT FUNC ID
903   H0080 DC       /0080     SUBPROGRAM ID WORD
904   H0028 DC       /0028     REFERENCED GENERATED TEMP I
905   H0054 DC       /0054     EXPRESSION OPERATOR
906   H5980 DC       /5980     ,SLT 16,
907   HB000 DC       /B000     ,M L,
908   H4000 DC       /4000     END STMNT ID WORD
909   LIBFR DC       /2180     ,LIBF READ,
910   LIBFF DC       /0480     ,LIBF WRITE,
911   H4028 DC       /4028     INTEGER, DEFINED VAR, GT ID
912   H1900 DC       /1900     ,CALL SUBSCR,
913   H6000 DC       /6000     CONSTANT FOR DIMENSION TEST
914   H8380 DC       /8380     CONSTANT
915   MDXL1 DC       /6100-/1900   ,MDX  L1,
916   LDXL  DC       /5B00-/6100   ,LDX  L,
917   LDXI  DC       /4800-/5B00   ,LDX  I,
918   STXL1 DC       /6200-/4800   ,STX  L1,
919   UNARY DC       /0020     UNARY MINUS
920   CALLX EQU      H000E
921   IF    DC       /0014-/002E  IF OPERATOR
922   H4220 DC       /4220     INTEGER STMNT NO,
923   H1300 DC       /1300     ,LDX L1,
924   READ  EQU      H9000
925   WRITE EQU      SL
926   FIND  EQU      SRT16
927   *
928   *              GETS THE SYM TBL ID WORD OF THE
929   *              VARIABLE NAME,
930   SYMT  DC       0         ENTRY POINT
931         AND    3 H07FF-Z   GET NO, OF ENTRY RELATIVE T
932         S      3 ONE-Z     START OF SYMBOL TABLE
933         STO      SYMT1 1   SAVE NUMBER
934         LD   L   SOFST     GET ADDRESS OF ENTRY BY
935         S        SYMT1 1   SUBTRACTING 3 TIMES ENTRY
936         S        SYMT1 1   NO, FROM START OF STRING
937         S        SYMT1 1   ADDRESS
938         STO      SYMT1 1   STORE ADDR OF ID WORD
939   SYMT1 LD   L   *-*       LOAD SYM TBL ID WORD
940         RTE      31        ROTATE WORD
941         STD    3 STNA1-Z   STORE ROTATED WORD
942         BSC  I   SYMT      RETURN
943   *
944   *              GETS THE NAME OF THE NEXT NEW
945   *              SYMBOL TABLE ENTRY
946   *
947   STENT DC       0         ENTRY POINT
948         MDX  L   EOFST,-3  MOVE END OF SYM TBL ADDR
949         LD   L   SOFST     LOAD START OF SYM TBL ADDR
950         S    L   EOFST     GET NEW SIZE OF SYM TBL
951         SRT      16        POSITION FOR DIVIDE
952         D      3 THREE-Z   GET NO, OF TABLE ENTRY
953         OR     3 H8000-Z   PUT IN SIGN BIT FOR NAME
954         BSC  I   STENT     RETURN
955   *
956   *              CHANGES THE SUBTRACT, DIVIDE, AND
957   *              EXPONENTIATE OPTRS INTO REVERSE
958   *              OPTRS WHERE NEEDED
959   *
```

```
REVOP DC      0          ENTRY POINT
      STO   3 TEMP2-Z    SAVE OPERATOR
      S     3 H003E-Z    HAS OPERATOR BEEN REVERSED
      BSC  L  REV2,-Z    BRANCH IF YES
      A       H0038      IS OPERATOR MINUS
      BSC     -          SKIP IF NOT
      MDX     REV1       BRANCH IF YES
      S     3 TWO-Z      IS OPERATOR DIVIDE
      BSC     -          SKIP IF NOT
      MDX     REV1       BRANCH IF YES
      S     3 TWO-Z      IS OPERATOR EXPONENT
      BSC     Z          SKIP IF YES
      MDX     REV3       BRANCH IF NOT
*
REV1  LD    3 TEMP2-Z    LOAD OPERATOR
      A       H003E      ADD REVERSE CONSTANT
REV2  STO   3 TEMP2-Z    STORE REVERSED OPTR
REV3  LD    3 TEMP2-Z    LOAD OPERATOR
      BSC  I  REVOP      RETURN
H003E DC      /003E      REVERSING CONSTANT
H0038 DC      /0038      MINUS TEST CONSTANT
*
*               COMPUTES THE GT SYM TBL ADDR
GETGT DC      0          ENTRY POINT
      LD    3 NXTGT-Z    IS NEXT GT COMPUTED
      BSC  I  GETGT,Z    RETURN IF YES
*
*
CMPGT BSI   3 STENT-Z    GET SYM TBL ADDR FOR NEXT G
      STO   3 NXTGT-Z    STORE ADDR
*
*               MAKE THE ENTRY A GENER TEMP ENTRY
*               IN SYM T
*
      BSI   3 SYMT-Z     GET ADDRESS FOR NEXT ENTRY
      LD    3 H0028-Z    LOAD SYM TBL ID WORD
      STO  I  SYMT1 1    PUT IN SYM TBL
      MDX     GETGT 1    RETURN
NXTGT DC      0          NEXT AVAILABLE GT
*
*
*****************************************************
*               FOLLOWING SUBROUTINE PUTS THE CON-
*               TENTS OF A-REGISTER INTO OUTPUT
*               STRING AND PERFORMS HOUSEKEEPING OF
*               OUTPUT STRING POINTER,  OUTPUT
*               POINTER IS INDEX REGISTER 2   SUB-
*               ROUTINE HAS THREE ENTRY POINTS, USE
*               OF FIRST MOVES THE CONTENTS OF A-RE-
*               GISTER UNALTERED AS ONE WORD INTO
*               OUTPUT STRING, USING OTHER ENTRIES:
*                ,NAME, AND ,OPERATOR,-ENTRIES  MAY
*               RESULT IN PACKED OUTPUT  OPERATOR
*               AND NAME IN THE SAME WORD , SOME
*               OPERATORS ARE CONVERTED INTO SEVERAL
*               WORDS OF OUTPUT, THE SUBROUTINE ALSO
*               PERFORMS PARTS OF THE CONTROLS  THAT
*               ARE REQUIRED FOR MIXED MODE ARITH-
*               METIC AND MAY GENERATE CODES REPRE-
*               SENTING ,CALL FLT, OR ,CALL IFIX,
```

```
1020  ********************************************************
1021  ********************************************************
1022  *              ADDS A WORD TO THE STMNT STRING FROM
1023  *              THE ACCUMULATOR
1024  ********************************************************
1025  OUTUN DC       0          ENTRY POINT
1026  OUT12 STO    2 1          PUT WORD ON OUTPUT STRING
1027        BSI    3 INCR-Z     INCREMENT O/P PT AND NORM
1028  OUT13 BSC   I  OUTUN      RETURN
1029  ********************************************************
1030  *              OUTPUT AND COUNT
1031  *              ENTRY   OPERATOR
1032  ********************************************************
1033  *
1034  *              SAME AS UNALTERED ARGUMENT ENTRY
1035  *
1036  OUTOP EQU      OUTUN
1037  *
1038  CWORD DC       0          CODEWORD                 117
1039  NAME  DC       0          STORED NAME              119
1040  *
1041  ********************************************************
1042  *              ADDS A NAME TO THE STMNT STRING
1043  *              PACKED INTO ONE WORD WITH AN OPTR
1044  ********************************************************
1045  OUTNA DC       0          ENTRY POINT
1046  OUT31 STO      NAME       STORE OUTPUT WORD
1047        LD       OUTNA      LOAD RETURN ADDRESS
1048        STO      OUTUN      STORE IN OUTUN
1049        LD     2 0          LOAD LAST O/P WORD
1050        STO      CWORD      SAVE WORD
1051  *
1052  *              TEST IF END OF ARITH STMNT FUNC
1053  *               IDENTIFIED BY OPTR   ,ASSIGN,
1054  *              AND NAME    ,ARITH STMNT FUNC,
1055  *
1056        S      3 H000E-Z    IS IT ASSIGN OPERATOR
1057        BSC  L   OUT33,Z    BRANCH IF NOT
1058        LD       NAME       LOAD WORD TO BE OUTPUT
1059        BSI    3 SYMT-Z     GET SYM TBL ID WORD
1060        SLA      6          IS IT STMNT FUNCTION
1061        BSC  L   OUT33,-    BRANCH IF NOT
1062  *
1063  *              IS IT END OF AN ARITH STMNT FUNC
1064  *              STATEMENT
1065  *              DELETE LAST OUTPUT OPERATOR
1066  *
1067        LD     I NADR 1     LOAD STMNT ID WORD
1068        S      3 FOUR-Z     DECREMENT NORM
1069        STO    I NADR 1     PUT BACK ON STRING
1070        MDX    2 -1         DECREMENT O/P PT
1071        BSC    I OUTNA      RETURN
1072  *
1073  OUT33 LD       CWORD      LOAD LAST O/P WORD
1074        SRA      8
1075        BSC      -          SKIP IF NO REDEFINITION
1076        MDX      OUT51      BRANCH TO REDEFINE CODE WOR
1077  OUT41 LD       CWORD      LOAD CODE WORD
1078        BSC      Z          SKIP IF NOT PACKED O/P
1079        MDX      OUT44      BRANCH TO PACKED OUTPUT
```

```
1080  *
1081  *              NONPACKED
1082  *              ENTRY IF A-REG   CODEWORD VALUE
1083  *
1084  OUT43 STO    2 0          STORE CODE WORD
1085        LD       NAME       LOAD NAME
1086        MDX      OUT12      OUTPUT NAME
1087  *              PACKED
1088  *              A-REG IS ASSUMED   CODEWORD VALUE
1089  OUT44 OR       NAME       PACK CODE WORD AND NAME
1090        STO    2 0          PUT ON O/P STRING
1091        MDX      OUT13      RETURN
1092  *
1093  OUT51 LD       NAME       GET SYM TBL ID
1094        BSI    3 SYMT-Z     WORD FOR FUTURE USE
1095        LD       CWORD      LOAD CODE WORD
1096        S      3 H000E-Z    IS IT ASSIGN OPERATOR
1097        BSC  L   OUT91, -   BRANCH IF YES
1098        BSC      -Z         SKIP IF LESS THAN ASSIGN
1099        MDX      OUT61      BRANCH IF NOT
1100        LD       CWORD      LOAD CODE WORD
1101        SRA      1          COMPUTE ADDR TO
1102        A        ADRC1      GET OPERATOR
1103        STO      OUT53 1    STORE ADDR
1104        S        ADRC2      IS IT EXPONENT OPERATOR
1105        BSC  L   OUTC4, -   BRANCH IF NOT
1106        LD     3 STNA1-Z    LOAD SUM TBL ID WORD OF NAM
1107        BSC  L   OUT55,-    BRANCH IF FLOATING POINT
1108  *
1109  *              NAME IS INTEGER NAME
1110        LD       CWORD      LOAD OPERATOR
1111        S      3 H000C-Z    IS IT MULTIPLY OPERATOR
1112        BSC  L   OUT54, -   BRANCH IF YES
1113        A      3 H0004-Z    IS IT DIVIDE OPERATOR
1114        BSC  L   OUT85, -   BRANCH IF YES
1115  OUT53 LD   L   *-*        LOAD OPERATOR
1116        MDX      OUT44      PACK AND OUTPUT
1117  *
1118  *              ENTRY FOR INTEGER MULTIPLY
1119  OUT54 LD     3 NAME-Z     LOAD NAME
1120        OR     3 HB000-Z    PACK WITH MULTIPLY OPTR
1121        STO    2 0          PUT ON STRING
1122        LD     3 H5980-Z    LOAD ,SLT  16,
1123        MDX      OUT12      OUTPUT INSTRUCTION
1124  *
1125  *              ENTRY FOR OBTAINING ,CALL FADD,,
1126  *              ,CALL FSUB,, ,CALL FDIV,
1127  *
1128  OUT55 MDX  L   OUT53 1,FADD-AL   MODIFY ADDRESS
1129        LD   I   OUT53 1    LOAD OPERATOR
1130        MDX      OUT43      OUTPUT OPERATOR, NAME
1131  *
1132  ADRC1 DC       AL-2       ADDRESS CONSTANT
1133  ADRC2 DC       AL 3       ADDRESS CONSTANT
1134  *
1135  OUT61 S      3 H0036-Z    IS IT REVERSE SUBTRACT
1136        BSC  L   OUTF1, -   BRANCH IF YES
1137        S      3 TWO-Z      IS IT REVERSE DIVIDE
1138        BSC  L   OUTG1, -   BRANCH
1139        MDX      OUTB1      MUST BE REVERSE EXPONENTIAT
```

```
1140  *
1141  OUT85 LD     3 SRT16-Z   LOAD ,SRT 16,
1142        STO    2 0         PUT ON O/P STRING
1143        BSI    3 INCR-Z    INCREMENT STMNT NORM, O/P I P
1144        LD     3 DL-Z      LOAD ,D L,
1145        MDX      OUT44     OUTPUT OPERATOR, NAME
1146  *
1147  *               ASSIGN
1148  OUT91 LD    L  CCWD
1149        SLA      14        IS ARITHMETIC TRACE NEEDED
1150        BSC      -         SKIP IF YES
1151        MDX      OUT95     BRANCH IF NOT
1152        LDD    3 STNA1-Z   LOAD SYM TBL ID WORD
1153        RTE      1         PUT ENTIRE WORD IN ACC
1154        AND    3 H800C-Z   IS IT VARIABLE NAME
1155        BSC      Z         SKIP IF YES
1156        MDX      OUT95     BRANCH IF NOT
1157        LD     3 STNA1-Z   LOAD SYM TBL ID WORD
1158        BSC      -         SKIP IF INTEGER NAME
1159        MDX      OUT93     BRANCH IF NOT
1160        LD     3 IARIT-Z   LOAD ,CALL IARIT,
1161        MDX      OUT43     OUTPUT CALL, NAME,
1162  OUT93 LD     3 FARIT-Z   LOAD ,CALL FARIT,
1163        MDX      OUT43     OUTPUT CALL, NAME
1164  *
1165  OUT95 LD     3 STNA1-Z   LOAD SYM TBL ID WORD
1166        BSC      -         SKIP IF INTEGER NAME
1167        MDX      OUTA1     BRANCH IF NOT
1168        SLA      3
1169        BSC    L OUT96,C    BR IF INDEXED INTEGER STOR
1170        BSC    L OUT96,+Z   BR IF INDEXED INTEGER STOR
1171        LD     3 STOL-Z    LOAD ,STO L,
1172        MDX      OUT44     OUTPUT OPTR, NAME
1173  OUT96 LD     3 ISTOX-Z    CALL ISTOX SUBROUTINE
1174        MDX      OUT43      LOOP BACK
1175  OUTA1 LD     3 FSTO-Z    LOAD ,CALL FSTO,
1176        MDX      OUT43     OUTPUT OPTR, WORD
1177  *
1178  *               REV, EXPON,
1179  OUTB1 LD     3 MODSW-Z   TEST MODE SWITCH
1180        BSC    L OUTB2,Z   BRANCH IF FLOATING POINT
1181  *
1182  *               MODE   FIX
1183        BSI    3 GETGT-Z   GET GENERATED TEMPORARY
1184        OR     3 H9000-Z   COMBINE WITH ,STO L,
1185        MDX      OUTB3     OUTPUT
1186  *
1187  *               MODE   FLOAT
1188  OUTB2 LD     3 FSTO-Z    LOAD ,CALL FSTO,
1189        STO    2 0         OUTPUT CALL
1190        BSI    3 INCR-Z    MOVE PT, INCREMENT NORM
1191        BSI    3 GETGT-Z   GET GENERATED TEMPORARY
1192  OUTB3 STO    2 0         PUT ON O/P STRING
1193  *
1194  *               MAKE GT-MODE AGREE WITH MODESWITCH
1195        BSI    L GTMOD
1196  *
1197        BSI    3 INCR-Z    MOVE O/P PT, INCREMENT NORM
1198  *
1199  *
```

```
1200          LD     3 ONE-Z      SET MODE SWITCH
1201          STO    3 MODSW-Z    TO FIXED POINT
1202          LD     3 NAME-Z     OUTPUT
1203          OR     3 H8800-Z    ,LD L  NAME,
1204          STO    2 0          PUT ON OUTPUT STRING
1205          BSI    3 SYMT-Z     WAS NAME FIXED PT
1206          BSC   L  OUTC1,Z    BRANCH IF YES
1207    *
1208    *               CHANGE MODESWITCH AND
1209    *               REPLACE OUTPUT BY FLOATING ARITHM
1210    *
1211          LD     3 HF800-Z    SET MODE SWITCH
1212          STO    3 MODSW-Z    TO FLOATING POINT
1213          LD     3 FLD-Z      LOAD ,CALL FLD,
1214          STO    2 0          PUT ON O/P STRING
1215          BSI    3 INCR-Z     MOVE O/P PT, INCREMENT NORM
1216          LD     3 NAME-Z     LOAD NAME
1217          STO    2 0          PUT ON O/P STRING
1218    OUTC1 BSI    3 INCR-Z     MOVE O/P PT, INCREMENT NORM
1219    OUTC3 BSI    3 GETGT-Z    REPLACE STORED NAME BY
1220          STO    3 NAME-Z     GENERATED TEMPORARY
1221    *
1222    *               OPERTR, EXP
1223    OUTC4 LD     3 MODSW-Z    TEST MODE SWITCH
1224          BSC                 SKIP IF FIXED PT
1225          MDX      OUTD5      BRANCH IF NOT
1226          LD     3 NAME-Z     LOAD NAME
1227          BSI    3 SYMT-Z     GET SYM TBL ID WORD
1228          BSC      -          SKIP IF INTEGER NAME
1229          MDX      OUTD1      BRANCH IF NOT
1230          LD     3 FIXI-Z     LOAD ,CALL FIXI,
1231    OUTC5 BSC   L  OUT43      OUTPUT CALL, NAME
1232    *
1233    OUTD1 LD     3 FLT-Z      LOAD ,CALL FLT,
1234          STO    2 0          PUT ON O/P STRING
1235          BSI    3 INCR-Z     MOVE O/P PT, INCREMENT NORM
1236          LD     3 HF800-Z    SET MOD SWITCH
1237          STO    3 MODSW-Z    TO FLOATING PT
1238    *
1239    OUTD2 LD     3 TWO-Z      MAKE CODE WORD IN O/P STRIN
1240          STO    2 0          INTO A 2-WORD CALL OPTR
1241    *
1242    *               TEST IF NAME DIMENSIONED
1243          LD     3 NAME-Z     LOAD NAME
1244          BSI    3 SYMT-Z     GET SYM TBL ID WORD
1245          AND    3 H3000-Z    IS IT DIMENSIONED
1246          BSC   L  OUTD3, -   BRANCH IF NOT
1247    *
1248    *               OUTPUT 2-WD CALL FAXBX
1249          LD     3 FAXBX-Z    LOAD ,CALL FAXBX,
1250          BSI   L  CAXBX, -   BRANCH TO MAKE SYM TBL ENTR
1251    *                         IF NOT THERE ALREADY
1252          MDX      OUTD4      MOVE PT
1253    *
1254    *               OUTPUT 2-WD CALL ,FAXB,
1255    OUTD3 LD     3 FAXB-Z     LOAD ,CALL FABX,
1256          BSI   L  CAXB, -    BRANCT TO MAKE SYM TBL ENTR
1257    *                         IF NOT THERE ALREADY
1258    *
1259    OUTD4 STO    2 1          OUTPUT WORD
```

```
1260          BSI    3 INCR-Z    MOVE O/P PT, INCREMENT NORM
1261   *
1262          LD     3 NAME-Z    LOAD NAME
1263   OUTDX  BSC    L OUT12     OUTPUT NAME
1264   *
1265   OUTD5  LD     3 NAME-Z    LOAD NAME
1266          BSI    3 SYMT-Z    GET SYM TBL ID WORD
1267          BSC      -         SKIP IF INTEGER
1268          MDX      OUTD2     BRANCH IF NOT
1269   *
1270          LD     3 FAXI-Z    LOAD ,CALL FAXI,
1271          MDX      OUTC5     OUTPUT CALL
1272   *
1273   *                 REVERSE SUB
1274   OUTF1  LD     3 STNA1-Z   LOAD SYM TBL ID WORD
1275          BSC      -         SKIP IF INTEGER
1276          MDX      OUTF4     BRANCH IF NOT
1277          LD     3 STOFA-Z   LOAD ,STO FAC 1,
1278          STO    2 0         PUT ON O/P STRING
1279          LD     3 NAME-Z    LOAD NAME
1280          AND    3 H07FF-Z   GET SYM TBL ENTRY NO
1281          A      3 LDL-Z     COMBINE WITH ,LD L,
1282          STO    2 1         PUT ON O/P STRING
1283          BSI    3 INCR-Z    MOVE O/P PT, INCREMENT NORM
1284          LD     3 SFAC-Z    LOAD ,S FAC 1,
1285          MDX      OUTDX     OUTPUT
1286   *
1287   OUTF4  LD     3 FSBR-Z    LOAD,CALL FSBR,
1288          MDX      OUTC5     OUTPUT
1289   *
1290   *                 REVERSE DIVIDE
1291   OUTG1  LD     3 STNA1-Z   LOAD SYM TBL ID WORD
1292          BSC      -         SKIP IF INTEGER
1293          MDX      OUTG4     BRANCH IF NOT
1294          LD     3 STOFA-Z   LOAD ,STO FAC 1,
1295          STO    2 0         OUTPUT
1296          LD     3 NAME-Z    LOAD NAME
1297          OR     3 LDL-Z     COMBINE WITH ,LD L,
1298          STO    2 1         PUT ON O/P STRING
1299          LD     3 SRT16-Z   LOAD ,SRT 16,
1300          STO    2 2         PUT ON O/P STRING
1301          BSI    3 INCR-Z    INCREMENT NORM, MOVE O/P PT
1302          BSI    3 INCR-Z    INCREMENT NORM, MOVE O/P PT
1303          LD     3 DFAC-Z    LOAD ,D  FAC 1,
1304          MDX      OUTDX     OUTPUT
1305   *
1306   OUTG4  LD     3 FDVR-Z    LOAD ,CALL FDVR,
1307          MDX      OUTC5     OUTPUT
1308   *
1309   *                 MAKES THE MODE OF THE GT AGREE WITH
1310   *                 THE CURRENT MODE OF THE MODE SW
1311   *
1312   GTMOD  DC               0          ENTRY POINT
1313          STO    3 GTMO3-Z   SAVE WORD
1314          BSI    3 SYMT-Z    GET SYM TBL ID WORD
1315          LD     3 MODSW-Z   TEST MODE
1316          BSC    L GTMO1,    BRANCH IF FLOATING PT
1317          LD     3 H4028-Z   SET ID WORD TO FIXED PT
1318          MDX      GTMO2     PUT IN TABLE
1319   GTMO1  LD     3 H0028-Z   SET ID WORD TO FLOATING PT
```

```
1320  GTM02 STO  I  SYMT1 1   STORE IN TABLE
1321        LD    3 GTM03-Z  RESTORE ACCUMULATOR
1322        BSC  I  GTMOD     RETURN
1323  *
1324  *
1325  *               MAKES THE SYM TBL ENTRY FOR THE
1326  *               SUBROUTINE NAMES FAXB AND EAXB SINCE
1327  *               THEY ARE 2-WORD CALLS
1328  *
1329  CAXB  DC        0         ENTRY POINT
1330  *
1331  *               GET NAM  OF NEXT SYMT ENTRY
1332  *               AND STORE
1333  *
1334        BSI   3 STENT-Z   GET NEXT SYM TBL ENTRY NO,
1335        STO   3 FAXB-Z    SAVE ENTRY NO,
1336        STO     CAXBX     SAVE ENTRY NO,
1337  *
1338        LDD     NAME1     GET EBC-NAME ,FABX,
1339  CAXB1 STD   3 TEMP-Z    STORE NAME
1340  *
1341  *               MAKE SYMBOL TABLE ID-WD ENTRY
1342        STX  L1 CAXB3 1   SAVE I/P PT
1343        LDX  I1 EOFST     LOAD XR1 WITH END OF STRING
1344        LD    3 H0080-Z   MAKE ID WORD A SUBPROGRAM
1345        STO   1 3         PUT IN SYM TBL
1346  *
1347  *               IF EXTENDED PREC, CHANGE
1348  *               NAME
1349  *
1350        LD   L  CCWD
1351        SLA     13        IS THERE STANDARD PRECISION
1352        BSC  L  CAXB2,-   BRANCH IF YES
1353  *
1354  *               CHANGE NAME TO BEGIN WITH ,E,
1355        LD      NAME3     CHANGE ,FAXB,
1356        STO   3 TEMP-Z    TO ,EAXB,
1357  *
1358  CAXB2 LDD   3 TEMP-Z    LOAD NAME
1359        STO   1 4         PUT 1ST WORD IN SYM TBL
1360        RTE     16
1361        STO   1 5         PUT 2ND WORD IN SYM TBL
1362  *
1363  CAXB3 LDX  L1 *-*       RESTURE I/P PT
1364        LD      CAXBX     LOAD SUBPROGRAM NAME
1365        BSC  I  CAXB      RETURN
1366  *
1367  *
1368  *               MAKES THE SYM TBL ENTRY FOR THE
1369  *               SUBROUTINE NAMES FAXBX AND EAXBX
1370  *               SINCE THEY ARE 2-WD CALLS
1371  *
1372  CAXBX DC        0         ENTRY POINT
1373        LD      CAXBX     MOVE RETURN ADDR SO CAN USE
1374        STO     CAXB      ,CALL FABX, SUBROUTINE
1375        BSI   3 STENT-Z   GET NEXT SYM TBL ENTRY NO,
1376        STO     CAXBX     STORE
1377        STO   3 FAXBX-Z   STORE
1378        LDD     NAME2     LOAD EBC-NAME ,FAXBX,
1379        MDX     CAXB1     PUT IN SYM TBL
```

```
1380  *
1381  *                CONSTANTS
1382          BSS  E  0
1383  NAME1   DC       /8C0C      EBC=NAME
1384          DC       /F080      ,FAXB,
1385  NAME2   DC       /8C0C      EBC=NAME
1386          DC       /F0A7      ,FAXBX,
1387  NAME3   DC       /8A0C      EBC=NAME ,EAX,
1388  *
1389  *
1390  *                LOCATES THE NEXT OPTR NOT YET
1391  *                PROCESSED
1392  *
1393  FINDN   DC       0          ENTRY POINT
1394          STO   3  TEMP-Z     STORE O/P SWITCH
1395          MDX   1  2          INITIALIZE POINTER
1396          STX   1  PFIND      SAVE I/P PT
1397  FINDA   LD    I  PFIND      LOAD WORD IN STMNT
1398          S     3  H1900-Z    IS WORD ,CALL SUBSCR,
1399          BSC   L  MVSUB,-    BRANCH IF YES
1400          S     3  MDXL1-Z    IS IT ,MDX L1,
1401          BSC   L  MV3,-      BRANCH IF YES
1402          S     3  LDXL-Z     IS IT ,LDX L1,
1403          BSC   L  MV2,-      BRANCH IF YES
1404          S     3  LDXI-Z     IS IT ,LDX I1,
1405          BSC   L  MV2,-      BRANCH IF YES
1406          S     3  STXL1-Z    IS IT ,STX L1,
1407          BSC   L  MV2,-      BRANCH IF YES
1408          LD    I  PFIND      RESTORE WORD
1409          BSC   I  FINDN      RETURN
1410  *
1411  MV3     BSI      MOVE1      MOVE WORD TO O/P STRING
1412  MV2     BSI      MOVE1      MOVE WORD TO O/P STRING
1413  MV1     BSI      MOVE1      MOVE WORD TO O/P STRING
1414          MDX      FINDA      RETURN
1415  MVSUB   BSI      MOVE1      MOVE WORD TO O/P STRING
1416          BSI      MOVE1      MOVE WORD TO O/P STRING
1417  MVS1    BSI      MOVE1      MOVE WORD TO O/P STRING
1418          BSI      MOVE1      MOVE WORD TO O/P STRING
1419          LD    I  PFIND      LOAD WORD
1420          BSC   L  MVS1,-     BRANCH IF NOT TAGGED
1421          MDX      MV1        MOVE WORD, RETURN
1422  *
1423  MOVE1   DC       0          ENTRY POINT
1424          LD    3  TEMP-Z     TEST SWITCH
1425          BSC   L  MOVEX,Z    BRANCH NOT OUTPUT
1426          LD    I  PFIND      LOAD WORD
1427          BSI   3  OUTUN-Z    PUT ON O/P STRING
1428          MDX   1  1          MOVE I/P PT
1429  MOVEX   MDX   L  PFIND,1    MOVE SUBROUTINE PT
1430          BSC   I  MOVE1      RETURN
1431  *
1432  PFIND   DC       0          POINTER IN SUBROUTINE
1433  *
1434  *
1435  *                NOT IN CARD SYSTEM
1436  *
1437  *
1438  *                OUTPUTS DISK READ/WRITE STMNTS
1439  *
```

```
1440  D5011 LD    I  INID      LOAD STMNT ID WORD
1441        AND   3  H07FC-Z   GET NORM
1442        SRA      2         RIGHT JUSTIFY
1443        A     3  INID-Z    GET ADDR OF NEXT STMNT
1444        S     3  ONE-Z     SUBTRACT ONE
1445        STO      NXTID     SAVE ADDRESS
1446        LD    1  3         LOAD WORD
1447        BSI   3  SYMT-Z    GET SYM TBL ID WORD
1448        BSC   L  D5021,-   BRANCH IF NOT INTEGER
1449  *
1450  *              OUTPUT REST OF READ/WRITE
1451  *              STATEMENT
1452  *
1453  D5013 MDX   1  1         MOVE I/P PT
1454        LD    1  0         LOAD WORD
1455        BSI   3  OUTUN-Z   PUT ON O/P STRING
1456        STX   1  TSX1      STORE I/P PT
1457        LD       TSX1      TEST FOR END
1458        S        NXTID     OF STMNT
1459        BSC   L  P1021,-   BRANCH IF END OF STMNT
1460        BSC   L  D5013     CONTINUE O/P
1461  *
1462  *              ERROR ENCOUNTERED
1463  *              EXPRESSION NOT INTEGER
1464  *              IN READ/WRITE STMT
1465  *
1466  D5021 LDX   I2 OUTID     RESET O/P PT
1467        LDX   I1 INID      RESET I/P PT
1468        LD       HA008     LOAD ERROR STMNT ID WORD
1469        STO   2  0         PUT ON O/P STRING
1470        LD    1  0         LOAD STMNT ID WORD
1471        SLA      15        DOES STMNT HAVE STMNT NO,
1472        BSC   L  D5023,-   BRANCH IF NOT
1473        LD       HA00D     REPLACE ERROR ID WORD WITH
1474        STO   2  0         ERROR WORD WITH STMNT NO,
1475        LD    1  1         LOAD STMNT NO,
1476        STO   2  1         PUT ON O/P STRING
1477        MDX   2  1         MOVE O/P PT
1478  D5023 LD       ERNO      LOAD ERROR NO,
1479        STO   2  1         PUT ON O/P STRING
1480        MDX   2  1         MOVE O/P PT
1481        LDX   I1 NXTID     MOVE I/P PT TO NEXT STMNT
1482  *
1483        BSC   L  P1021     PROCESS NEXT STMNT
1484  *
1485  EXPRO DC       /0054     EXPRESSION OPERATOR
1486  HA008 DC       /A008     ERROR STMNT ID WORD
1487  HA00D DC       /A00D     ERROR ID WITH STMNT NO,
1488  ERNO  DC       73        ERROR NO, 73
1489  NXTID DC       0         ADDR OF NEXT I/P STMNT - 1
1490  TSX1  DC       0         INPUT PTR SAVE
1491  *
1492  *
1493  *
1494  *
1495        BSS      OVERL-**+320*4  PHASE-18 PATCH AREA
1496        END      START
```