```
1         HDNG    MPX FORTRAN ** EXPANDER I
2   *   GENERAL AUTOMATION, INC.  ALL RIGHTS RESERVED
3   ************************************************
4   *
5   *   PROGRAM NAME  FPH-17
6   *
7   *   MODEL NUMBER  8F017
8   *
9   *   PURPOSE       FORTRAN PHASE-17
10  *
11  *   PROGRAMMER    DICK WALLMANN
12  *
13  *****************   REVISION LIST   **************
14  *
15  *   RV DATE     SCO   BY  REASON FOR CHANGE
16  *   -- -------- ----- --- ------------------------
17  *
18  *   01 11/16/70 NONE  RPH INITIAL RELEASE
19  *
20  ************************************************
21  ************************************************
22  ************************************************
23  *STATUS-VERSION 1, MODIFICATION 0
24  *
25  *FUNCTION/OPERATION-
26  *    * REPLACES READ, FIND, WRITE, GO TO, AND
27  *      RETURN STATEMENTS WITH COMPILER-GENERATED
28  *      CODING
29  *    * SETS UP IMPLIED DO LOOPS WITHIN READ AND
30  *      WRITE STATEMENT
31  *    * REPLACES THOSE PARTS OF ARITHMETIC, IF,
32  *      CALL AND STMNT FUNCTION STATEMENTS THAT
33  *      INVOLVE SUBSCRIPTING OF VARIABLES WITH
34  *      COMPILER-GENERATED CODING,
35  *    * CHECKS SUBPROGRAMS FOR A RETURN STATEMENT.
36  *
37  *ENTRY POINTS-
38  *    * START-PHASE 17 IS READ INTO CORE BY PHASE
39  *            16 AND EXECUTION BEGUN AT LOCATION
40  *            LABLED START.
41  *
42  *INPUT-
43  *    * THE STATEMENT STRING
44  *    * THE SYMBOL TABLE
45  *    * THE FORTRAN COMMUNICATION AREA
46  *
47  *OUTPUT-
48  *    * THE STATEMENT STRING
49  *    * THE SYMBOL TABLE
50  *    * THE FORTRAN COMMUNICATION AREA
51  *
52  *EXTERNAL REFERENCES-
53  *    * SUBROUTINES
54  *      ROLRX
55  *    *OTHER FORTRAN PHASES
56  *      NONE
57  *
58  *EXITS-
59  *    * NORMAL-
```

```
60  *          PHASE 18 IS CALLED VIA ROLRX AND
61  *          CONTROL IS PASSED TO IT.
62  *    * ERRORS-
63  *      OVERLAP-
64  *          PROCESSING IS HALTED, THE ERROR WORD IS
65  *          SET IN FCOM, PHASE 18 IS CALLED VIA
66  *          ROLRX, CONTROL IS PASSED TO IT.
67  *      SYNTAX-
68  *          THE ERRONEOUS STATEMENT IS REPLACED
69  *          WITH AN ERROR NUMBER, PROCESSING
70  *          CONTINUES, PHASE 18 IS CALLED VIA ROLRX
71  *          AND CONTROL IS PASSED TO IT.
72  *          THE SYNTAX ERROR DETECTED IN PHASE
73  *          NUMBRR 17 IS 69.
74  *
75  *TABLES/WORK AREAS-
76  *    * THE STATEMENT STRING
77  *    * THE SYMBOL TABLE
78  *    * THE FORTRAN COMMUNICATION AREA
79  *
80  *ATTRIBUTES-N/A
81  *
82  *NOTES-
83  *    THE SWITCHES USED IN PHASE 14 FOLLOW, IF NON-
84  *    ZERO, THE SWITCH IS TRANSFER T, IF ZERO, THE
85  *    SWITCH IS NORMAL N.
86  *       * RETCT-RETURN STATEMENTS
87  *                N NO RETURN STMNTS ENCOUNTERED
88  *       * STXSW-
89  *                T TWO WORD CALL HAS INDEXED
90  *                  ARGUMENTS, SECOND SCAN NECESSARY
91  *                  TO COMPUTE DISPLACEMENT VALUE IN
92  *                  STX L1 INSTRUCTION
93  *       * USWIT-UNFORMATTED I/O
94  *                T STMNT IS NOT FORMATTED, I.E.,
95  *                  DOES NOT HAVE UPTR AT I/O OPTR 2.
96  *       * DSWIT-DISK SWITCH
97  *                T DISK I/O OPTR
98  ****************************************************
99        HDNG    MPX FORTRAN ** EXPANDER I
100       ABS     REF CORE
101 *
102 *     SYSTEM AND FORTRAN EQUATES
103 *
104 MEMRY EQU     ~~77FFF~~ CORE      MAXIMUM CORE SIZE
105 PHSIZ EQU     4*320               MAXIMUM PHASE SIZE
106 OVERL EQU     MEMRY-PHSIZ         PHASES 2-29 START
107 FCOM  EQU     OVERL-22            FORTRAN COMM, TABLE
108 PHNTB EQU     FCOM-56             PHASE TABLE
109 ROLRX EQU     PHNTB-50            INTERPHASE CALL
110 *
111 *     FORTRAN COMMUNICATION AREA
112 *
113 ****************************************************
114 *
115 *                    FORTRAN COMMUNICATION AREA
116       ORG     FCOM
117 *
118 SOFS  BSS     1     START OF STRING
119 EOFS  BSS     1     END OF STRING
```

```
120  SOFST BSS      1     START OF SYMBOL TABLE
121  SOFNS BSS      1     START OF NON-STATEMENT NUMBERS
122  SOFXT BSS      1     START OF SUBSCRIPT TEMPORARIES
123  SOFGT BSS      1     START OF GENERATED TEMPORARIES
124  EOFST BSS      1     END OF SYMBOL TABLE
125  COMON BSS      1     NEXT AVAILABLE COMMON
126  CSIZE BSS      1     SIZE OF COMMON
127  ERROR BSS      1     OVERLAP ERROR
128  FNAME BSS      1     PROGRAM NAME
129        BSS      1      2ND WORD PROG NAME
130  SORF  BSS      1     SUBROUTINE -  OR FUNCTION
131  CCWD  BSS      1     CONTROL CARD WORD
132  *                   BIT 15 TRANSFER TRACE
133  *                   BIT 14 ARITHMETIC TRACE
134  *                   BIT 13 EXTENDED PRECISION
135  *                   BIT 12 LIST SYMBOL TABLE
136  *                   BIT 11 LIST SUBPROGRAM NAMES
137  *                   BIT 10 LIST SOURCE PROGRAM
138  *                   BIT  9 ONE WORD INTEGERS
139  IOCS  BSS      1     IOCS CONTROL CARD WORD
140  *
141  *         SEE PHASE ONE FOR BIT PATTERNS
142  *
143  DFCNT BSS      1           DEFINE FILE COUNT
144  *
145  LCOMN BSS      2           SIZE OF INSKEL COMMON
146  *
147  ICCER BSS      2           IOCS CONTROL CARD ERROR
148  *
149        BSS      2           SYSTEM LOADER USE
150  *
151  *                     END OF FORTRAN COMMUNICATION
152  *                     AREA
153        HDNG       MPX FORTRAN ** EXPANDER I
154        ORG        OVERL       ORIGIN TO FORT OVERLAY PT
155  START LD         ERROR       CHECK FOR OVERLAP
156        BSC  L     EXIT,Z      EXIT PHASE IF OVERLAP
157        LD         EOFS        LD END OF STRING ADDRESS
158        S          SOFS      SUBTRACT START OF STRING
159        LDX  L3 ZERO         SET XR3
160        A       3  ONE-Z     SIZE OF STRING
161        STO        MOVCT     STORE STRING SIZE
162        LDX  I1 EOFST        INITIALIZE OUTPUT PT
163        MDX     1  -10
164        LDX  I2 EOFS         INITALIZE INPUT PT
165  *
166  *                TEST IF OVERLAP ERROR
167        STX  L1 TEMP1        IS THERE OVERLAP
168        LD         EOFS      BETWEEN STRING
169        S       3  TEMP1-Z   AND SYMBOL TABLE
170        BSC  L     MOVST, Z  BRANCH IF NO
171  OVLAP MDX  L     ERROR,1   SET OVERLAP ERROR
172        BSC  L     EXIT      GO TO NEXT PHASE
173  MOVCT DC         *-*        LENGTH OF STRING
174  *
175  MOVST LD      2  0         MOVE STRING ITEM
176        STO     1  0         NEXT TO SYMBOL TABLE
177        MDX     2  -1        MOVE POINTER
178        MDX     1  -1
179        MDX  L     MOVCT,-1  DECREMENT STRING SIZE CT
```

```
180          MDX      MOVST     CONTINUE LOOP
181   *
182          STX   L1 SOFIS     NEW START OF STRING ADDR -1
183          LD    L  SOFST     PUT START OF SYM TBL ADDR
184          STO    3 SSOST-Z   IN SSOST
185   *
186   Q1011  LDX   I1 SOFIS     INITIALIZE I/P STRING PT
187          LDX   I2 SOFS      INITALIZE O/P STRING PT
188          MDX    2 -1
189   *
190   *
191   *                INITIALIZES TO SCAN THE STMNT
192   Q1021  MDX    1 1         MOVE I/P STRING PT
193          MDX    2 1         MOVE O/P STRING PT
194   *
195          LD     3 ZERO-Z
196          STO    3 STXSW-Z
197          STO    3 CURR-Z    INITIALIZE SWITCHES
198          LD     3 H8000-Z
199          STO    3 CURD4-Z
200   *
201   *                EXTRACTS THE STMNT TYPE
202   Q1022  LD     1 0         LOAD STMNT ID WORD
203          AND    3 HF803-Z   ELIMINATE NORM
204          A      3 FOUR-Z    ADD ONE TO NORM
205          STO    2 0         PUT ON OUTPUT STRING
206          AND    3 HF800-Z   GET STMNT ID TYPE
207          STO    3 STTYP-Z   SAVE ID TYPE
208          STX   L2 OUTID     STORE ADDR OF O/P ID
209          STX   L1 INID      STORE ADDR OF I/P ID
210          MDX    1 1         MOVE I/P PT
211          LD     1 -1        LOAD STMNT ID WORD
212          BSC   L  Q1023,E   BRANCH IF HAVE STMNT NO,
213          MDX      Q1031     IDENTIFY STMNT TYPE
214   *
215   Q1023  BSI    3 MOVAW-Z   O/P STMNT NO,, MOVE PTS
216   *
217   *                TESTS FOR ARITHMETIC, STMNT FUNCTION
218   *                CALL, IF, GO TO, READ, FIND, WRITE,
219   *                AND RETURN STMNTS,
220   Q1031  STX    1 Q1033 1   SAVE I/P STRING PT
221          LDX   L1 BRTAB-2   INITIALIZE TABLE PT
222   Q1032  MDX    1 2         INCREMENT TABLE PT
223          LD     1 0         LOAD TABLE WORD
224          BSC   L  Q1041, -  BRANCH IF END OF TABLE
225          S      3 STTYP-Z   DOES TABLE TYPE MATCH
226          S      3 ONE-Z     STATEMENT TYPE
227          BSC   L  Q1032,Z   BRANCH IF NOT
228          LD     1 1         LOAD BRANCH ADDRESS
229          STO      Q1034 1   INSERT BRANCH ADDRESS
230   Q1033  LDX   L1 *-*       RESTORE I/P STRING PT
231   Q1034  BSC   L  *-*       BRANCH TO PROPER ID SUBR
232   *
233   *                BRANCH TABLE
234   *                FOR STATEMENT TYPE
235   *
236   *
237   *                /0001 IS ADDED TO TYPE IN ORDER TO
238   *                ENABLE NORMAL SEARCH FOR
239   *                TYPE   ZERO  ARITHMETIC STMNT
```

```
240  BRTAB DC      /0001     ARITH
241        DC      Q1061      BRANCH ADDR
242        DC      /0001     ARITH STMT FUNCTION
243        DC      Q1051      BRANCH ADDR
244        DC      /3001     CALL
245        DC      Q1061      BRANCH ADDR
246        DC      /7801     IF
247        DC      Q1061      BRANCH ADDR
248        DC      /7001     GOTO
249        DC      P2011      BRANCH ADDR
250        DC      /9001     READ
251        DC      P1052      BRANCH ADDR
252        DC      /8801     WRITE
253        DC      P1051      BRANCH ADDR
254        DC      /8001     RETURN
255        DC      P2031      BRANCH ADDR
256        DC      /E801     FIND
257        DC      P1053      BRANCH ADDR
258        DC      0         END-OF-TABLE-VALUE
259  *
260  *
261  *
262  *
263  P1053 LD     3 FIND-Z    LOAD FIND CALL
264        MDX      D1011     CHECK I/O OPERATOR
265  *
266  *
267  P1051 LD     3 FWRT-Z    LOAD WRITE CALL
268        MDX      D1011     CHECK FOR I/O OPERATOR
269  *
270  *                 ENTRY PT FOR READ-STMT
271  *
272  P1052 LD     3 FRED-Z    LOAD READ CALL
273  *
274  *                 DETECTS THE DISK I/O OPTR, OUTPUTS
275  *                 THE APPROPRIATE OPTRS AND ARGUMENTS
276  D1011 STO    3 TEMPY-Z   STORE CALL
277  *
278        LD     3 ZERO-Z    CLEAR
279        STO    3 DSWIT-Z   DISK SWITCH AND
280        STO    3 USWIT-Z   UNFORMATTED I/O SWITCH
281  *
282        LD     1 0         LOAD WORD
283        S      3 IOOPR-Z   IS IT I/O OPERATOR
284        BSC  L  D1015, -   BRANCH IF YES
285        MDX  L  DSWIT,1    SET DISK SWITCH
286        LD     1 0         LOAD WORD
287        S      3 DIOOP-Z   IS IT DISK I/O OPERATOR
288        BSC  L  Q2008, -   BRANCH IF YES
289        LD     3 EXPRO-Z   LOAD EXPRESSION OPERATOR
290        BSI    3 OUTP-Z    OUTPUT ON STRING
291  *
292        BSC  L  Q2011      IDENTIFY NEXT OPERATOR
293  *
294  *                 DISK I/O OPTR ENCOUNTERED AFTER
295  *                 AN ARITHMETIC EXPRESSION
296  *
297  D1014 LD     3 EXPRO-Z   LOAD EXPRESSION OPERATOR
298        BSI    3 OUTP-Z    OUTPUT ON STRING
299  *
```

```
300  *
301  D1015 LD    1 2         IS THIS UNFORMATTED I/O
302        BSC  L D1021,Z    BRANCH IF NOT
303        STX  L USWIT      SET UNFORMATTED I/O SWITCH
304  *
305  D1021 LD    3 TEMPY-Z   OUTPUT CALL WITH DISK
306        BSI   3 DTEST-Z   AND UNFORMATTED I/O TEST
307  *
308        LD    1 1         LOAD AND O/P ADDR OF
309        BSI   3 OUTP-Z    LOGICAL UNIT NUMBER
310        LD    3 USWIT-Z   IS STMNT UNFORMATTED
311        BSC  L D1024, -   BRANCH IF NO
312        MDX   1 2         INCREMENT PT
313        MDX     D1026     CHECK FOR FIND STMNT
314  D1024 LD    1 2         LOAD AND OUTPUT
315        BSI   3 OUTP-Z    ADDRESS OF FORMAT
316        MDX   1 3         MOVE PT
317  D1026 LD    1 0         LOAD WORD
318        BSC  L P3011,Z    BRANCH IF NOT SEMI-COLON
319  *
320  *               GENERATES A CALL SCOMP WHEN A
321  *               WRITE STMNT IS DETECTED
322  D1031 EQU       *
323  P1061 EQU       D1031
324        LD    3 STTYP-Z   LOAD STMNT ID TYPE
325        S     3 FNDID-Z    IS IT A FIND STATEMENT
326        BSC  L Q1021,+-    BRANCH IF YES
327  P1062 LD    3 FCOMP-Z   OUTPUT ,CALL SCOMP,
328        BSI   3 DTEST-Z   WITH DISK TEST
329        BSC  L Q1021      GET NEXT STMNT
330  *
331  *
332  *               GO TO STMNT
333  *               PRODUCES A  BSC  L  INSTRUCTION FROM
334  *               A SIMPLE GO TO STMNT, IF TRACING  IS
335  *               REQUIRED, OUTPUTS THE CALL TO THE
336  *               TRACE ROUTINE
337  *
338  P2011 LD    1 1         IS THIS UNCONDITIONAL GO TO
339        BSC     Z         SKIP IF YES
340        MDX     P2012     COMPUTED GO TO BRANCH
341        LD    3 HF800-Z   OUTPUT ,BSC L,
342        BSI   3 OUTP-Z    PUT ON STRING
343        LD    1 0         LOAD BRANCH TO ADDR
344        AND   3 H07FF-Z   COMBINE INSTRUCTION AND
345        A     2 0         ADD ADDRESS TO GET
346        STO   2 0         *****  BSC  L  ADDRESS
347        MDX   1 1         MOVE I/P PT
348        BSC  L Q1021      GO TO NEXT STMNT
349  *
350  *               COMPUTED GOTO
351  *               PRODUCES A  BSC  I1  INSTR FROM A
352  *               COMPUTED GO TO OPTR, IF TRACING IS
353  *               REQUIRED, OUTPUTS THE CALL TO THE
354  *               TRACE ROUTINE
355  P2012 LD    1 1         LOAD INTEGER VARIABLE
356        AND   3 H07FF-Z   COMBINE WITH ,LDX  I1,
357        A     3 LDXI1-Z   TO GET
358        BSI   3 OUTP-Z    *****  MDX  I1 INTEGER VAR
359  *
```

```
360  *                MOVE FIRST STMNT NUMBER NAME ON
361  *                STRING SO THAT ALL STMNT NUMBERS
362  *                WILL BE CONSECUTIVE
363  *
364          LD     1 0          MOVE FIRST STMNT NO.
365          STO    1 1          ON STRING
366  *
367  *                TEST IF TRACE REQUIRED
368          LD     L CCWD       LOAD CONTROL CARD WORD
369          SLA      15         IS TRACE REQUIRED
370          BSC      -          SKIP IF YES
371          MDX      P2021      BRANCH IF NOT
372  *
373          LD     3 FGOTO-Z    OUTPUT ,CALL SGOTO,
374          BSI    3 OUTP-Z     ON OUTPUT STRING
375          MDX      P202A      OUTPUT STMNT NOS,
376  *
377  P2021   LD     3 COMGO-Z    OUTPUT ,CALL COMGO,
378          BSI    3 OUTP-Z     GO DO IT
379  *
380  *                GET TWO WORD CALL DUMMY
381  *
382  P202A   MDX    1 1          MOVE I/P POINTER
383          LD     3 H5F00-Z    GET DUMMY WORD
384          BSI    3 OUTP-Z     GO PUT IT ON STRING
385          STX    2 P2023+1    SAVE XR2 COUNT
386  *
387  *                OUTPUT STATEMENT NUMBERS IN
388  *                COMPUTED GO TO
389  *
390  P2022   LD     1 0          LOAD WORD
391          BSC    L Q1021, -   BRANCH IF SEMICOLON
392          BSI    3 OUTP-Z     OUTPUT STMNT NO,
393          MDX    1 1          MOVE I/P STRING PT
394  P2023   MDX    L *-*,1      ADD ONE TO COUNT
395          MDX      P2022      CONTINUE LOOP
396  *
397  *                GENERATES THE RETURN LINKAGE FROM A
398  *                SUBPROGRAM, ALLOWS THE ARGUMENTS TO
399  *                BE PASSED
400  P2031   LD     L SORF       IS THIS MAINLINE PROGRAM
401          BSC    L ERRR, -    BRANCH IF YES
402          BSC    L P2034,     BRANCH IF SUBROUTINE
403          LD     L FNAME      LOAD PROGRAM NAME
404          BSI    3 GETID-Z    GET SYMBOL TABLE ID WORD
405          SLA      1
406          BSC      -          SKIP IF INTEGER FUNCTION
407          MDX      P2032      BRANCH IF NOT
408          LD     L FNAME      LOAD PROGRAM NAME
409          OR     3 LDL-Z      COMBINE WITH ,LD   L, TO GE
410          MDX      P2033      *****  LD   L  PROGRAM NAME
411  P2032   LD     3 FLD-Z      LOAD ,CALL FLD,
412          BSI    3 OUTP-Z     PUT CALL ON OUTPUT STRING
413          LD     L FNAME      LOAD FUNCTION NAME
414  P2033   BSI    3 OUTP-Z     PUT ON OUTPUT STRING
415  P2034   MDX    L RETCT,1    COUNT RETURN STMNTS
416          LD     3 H5080-Z    LOAD ,BSC I,
417          BSI    3 OUTP-Z     PUT ON OUTPUT STRING
418          LD     3 H5F00-Z    OUTPUT BLANK WHICH WILL BE
419  *                           REPLACED BY ENTRY-PT ADDR
```

```
420  *                              *****  BSC  I  ENTRY PT
421         BSI    3 OUTP-Z       PUT ON OUTPUT STRING
422         BSC  L  Q1021         GO TO NEXT STMNT
423  *
424  *                 CHANGE RETURN STMNT INTO
425  *                 ERROR STATEMENT
426  *
427  ERRR   LD     I  OUTID       LOAD RETURN STMNT ID TYPE
428         A         DIFF        CHANGE TO ERROR STMNT ID
429         STO    I  OUTID       PUT BACK ON O/P STRING
430         LD     3  C69-Z       PUT ERROR NO, 69
431         BSI    3  OUTP-Z      ON OUTPUT STRING
432         BSC  L  Q1021         GO TO NEXT STMNT
433  DIFF   DC        /A000-/8000   ERROR STMNT ID TYPE
434  *
435  *                 IOOP
436  P3011  LD     1  0           LOAD WORD
437         S      3  LOP-Z       IS IT LIST OPERATOR
438         BSC       -           SKIP IF NOT
439         MDX       P3031       BRANCH IF LIST OPERATOR
440  *
441         LD     1  0           LOAD WORD
442         S      3  H0018-Z     IS IT LITERAL SUBSCRIPT
443         BSC  L  P3025, -      BRANCH IF YES
444         BSC  L  P4011,        BRANCH IF LESS THAN  0 OPTR
445         S      3  H0008-Z     IS IT GREATER THAN  3 OPTR
446         BSC  L  P4011,-Z      BRANCH IF YES
447  P3023  BSI  L  SBSN          PROCESS SUBSCRIPT OPERATORS
448         MDX       P3011       CHECK NEXT WORD
449  P3025  BSI  L  SBSLT         PROCESS LITERAL SUBSCRIPT
450         MDX       P3011       CHECK NEXT WORD
451  *
452  *                 IDENTIFIES THE LIST VARIABLE TYPE
453  P3031  LD     1  1           LOAD WORD
454         BSI    3  GETID-Z     GET SYM TBL ID
455         AND    3  H1800-Z     IS IT DIMENSIONED
456         BSC       -           SKIP IF YES
457         MDX       P3043       BRANCH IF NOT
458         LD     1  1           LOAD WORD
459         AND    3  H7800-Z     GET PSEUDO INDEX FOR SGT
460         BSC                   SKIP IF HAVE INDEX
461         MDX       P3051       BRANCH IF NOT
462         BSI    3  SCHP1-Z     LOAD LITERAL SUBSC OFFSET
463  *
464  *                 GENERATES THE CALLS FOR THE
465  *                 DIMENSIONED LIST VARIABLES WITH THE
466  *                 ASSOCIATED SGT
467  *
468  P3041  LD     1  1           LOAD WORD
469         BSI    3  GETID-Z     GET SYM TBL ID WORD
470         SLA       1           IS IT REAL VARIABLE
471         BSC  L  P304A,-       BRANCH IF YES
472  *
473         LD     3  FIOIX-Z     LOAD ,LIBF SIOIX,, OUTPUT
474         MDX       P304B       CALL, BASE ADDR OF DIM VAR
475  *
476  P304A  LD     3  FIOFX-Z     LOAD ,LIBF SIOFX,, OUTPUT
477  P304B  BSI    3  OUTIO-Z     CALL, BASE ADDR OF DIM VAR
478  *
479  P3042  MDX    1  2           MOVE I/P PT
```

```
480          MDX      P3011     CHECK NEXT WORD
481   *
482   *
483   *                GENERATES THE CALL FOR NON-
484   *                DIMENSIONED LIST VARIABLES WITH THE
485   *                ASSOCIATED SGT
486   *
487   P3043  LD     1 1          LOAD WORD
488          BSI    3 GETID-Z    GET SYM TBL ID WORD
489          SLA      1          IS IT REAL VARIABLE
490          BSC  L   P3044,-    BRANCH IF YES
491          LD     3 FIOI-Z     LOAD ,LIBF SIOI,
492          MDX      P3045      OUTPUT CALL, ADDR OF VAR
493   *
494   P3044  LD     3 FIOF-Z     LOAD ,LIBF SIOF,
495   P3045  BSI    3 OUTIO-Z    OUTPUT CALL, ADDR OF VAR
496          MDX      P3042      GET NEXT WORD
497   *
498   *
499   *                GENERATES THE CALL FOR A
500   *                DIMENSIONED LIST VARIABLE WITHOUT
501   *                THE ASSOCIATED SGT
502   P3051  LD     1 1          LOAD WORD
503          BSI    3 GETID-Z    GET SYM TBL ID WORD
504          SLA      1          IS IT REAL VARIABLE
505          BSC  L   P3052,-    BRANCH IF YES
506          LD     3 FIOAI-Z    LOAD ,LIBF SIOAI,, O/P CALL
507          MDX      P3053      ADDR OF 1ST ELEMENT IN ARRA
508   *
509   P3052  LD     3 FIOAF-Z    LOAD ,LIBF SIOAF,, O/P CALL
510   P3053  BSI    3 OUTIO-Z    ADDR OF 1ST ELEMENT IN ARRA
511   *
512   *                OUTPUT ARRAY SIZE
513   *                 ,TAGGED, TO FACILITATE HANDLING
514   *
515          MDX  L   SYMT1 1,-3  DECR SYM TBL ADDR
516          LD   I   SYMT1 1     LOAD ARRAY SIZE
517          OR     3 H8000-Z    TAG SIZE WORD
518          BSI    3 OUTP-Z     OUTPUT ARRAY SIZE
519          MDX      P3042      BRANCH ADDR
520   *
521   *
522   *                GENERATES THE DO INITIALIZE CODE
523   *                 DO-OPERATOR ENCOUNTERED
524   *
525   P4011  LD     1 0          LOAD WORD
526          BSC  L   P1061, -    BRANCH IF SEMI-COLON
527          S      3 DOAOP-Z    IS IT DO OPERATOR
528          BSC      Z          SKIP IF YES
529          MDX      P4023      BRANCH IF NOT
530          LD     3 ZERO-Z     CLEAR
531          STO    3 CURR-Z     PSX-IN-USE SWITCH
532          LD     1 2          LOAD INITIAL DO VALUE
533          AND    3 H07FF-Z    COMBINE WITH ,LD L,
534          A      3 LDL-Z      TO GET
535          BSI    3 OUTP-Z     *****  LD  L  INITIAL VALUE
536          LD     1 1          LOAD INDEX
537          AND    3 H07FF-Z    COMBINE WITH ,STO L,
538          A      3 STOL-Z     TO GET
539          BSI    3 OUTP-Z     *****  STO  L  INDEX
```

```
540          LD     3 GENLO-Z   OUTPUT GENERATED
541          BSI    3 OUTP-Z    LABEL OPERATOR
542          MDX  L   GENLO,1   INCREMENT STORED GENERAL
543  *                           LABEL OPTR BY ONE
544  P4021   MDX    1 3         MOVE I/P PT
545          BSC  L   P3011     CHECK NEXT ELEMENT
546  *
547  *                 GENERATES DO TEST CODE
548  P4023   LD     1 0
549          S      3 DOTOP-Z   IS WORD DO TEST OPTR
550          BSC  L   P1061,Z   BRANCH IF NOT
551  *
552  *                 DOTEST
553          MDX  L   GENLO,-1  DECREMENT GENERAL LABEL
554  *                           OPERATOR NUMBER BY 1
555          LD     1 3         IS INCREMENT GT ONE
556          BSC      -         SKIP IF YES
557          MDX      P4032     BRANCH IF INCREMENT IS ONE
558          BSI    3 OULDL-Z   *****  LD   L  INDEX
559          LD     1 3         LOAD INCREMENT
560          AND    3 H07FF-Z   COMBINE WITH ,A L,
561          A      3 AL-Z      TO GET
562          BSI    3 OUTP-Z    *****  A    L  INCREMENT
563          LD     1 1         LOAD INDEX
564          AND    3 H07FF-Z   COMBINE WITH ,STO L,
565          A      3 STOL-Z    TO GET
566          BSI    3 OUTP-Z    *****  STO  L  INDEX
567  P4031   BSI      P4035     *****  S    L  LIMIT VALJE
568  *                           *****  BSC  L  GEN LAB, Z
569          MDX    1 1         MOVE I/P PT
570          MDX      P4021     CHECK NEXT WORD
571  *
572  *
573  P4032   LD     3 MDXL1-Z   LOAD ,MDX L  *-*,1,
574          BSI    3 OUTP-Z    OUTPUT
575          LD     1 1         LOAD INDEX
576          BSI    3 OUTP-Z    *****  MDX  L  INDEX,1
577          BSI    3 OULDL-Z   *****  LD   L  INDEX
578          BSI      P4035     *****  S    L  LIMIT VALJE
579          MDX      P4021     *****  BSC  L  GEN LAB, Z
580  *                           CHECK NEXT WORD
581  *
582  *                 SUBROUTINE TO OUTPUT
583  *                       S    L  LIMIT VALUE
584  *                       BSC  L  GEN LAB, Z
585  *
586  P4035   DC       0          ENTRY POINT
587          LD     1 2         LOAD LIMIT VALUE
588          AND    3 H07FF-Z   COMBINE WITH ,S L,
589          A      3 SL-Z      TO GET
590          BSI    3 OUTP-Z    *****  S    L  LIMIT VALJE
591          LD     3 GENLO-Z   LOAD RETURN INSTRUCTION
592          BSI    3 OUTP-Z    *****  BSC  L  GEN LAB, Z
593          BSC  I   P4035     RETURN
594  *
595  *                 INSERTS THE FUNCTION NAME IN NAME
596  *                 CODE INTO FNAME IN THE FORTRAN
597  *                 COMMUNICATIONS AREA
598  TMTE    SLA      0         NOP
599  *
```

```
600   *              PROCESS ,FNAME, IN
601   *              COMMUNICATION AREA
602   *
603   *              IF MAINLINE PROGRAM, NO ACTION TAKEN
604   *
605           LD   L  SORF       IS THIS MAINLINE PROGRAM
606           BSC  L  EXIT, -    BRANCH IF YES
607   *
608   *              IT IS SUBROUTINE OR FUNCTION
609   *              REPLACE SYM TBL REF IN FNAME BY
610   *              NAME  WITH CHARACTERS PACKED IN
611   *              OUTPUT MODE
612   *
613           LD   L  FNAME      LOAD PROGRAM NAME
614           BSC  L  EXIT,-     BRANCH IF ALREADY HAVE NAME
615           BSI   3 GETID-Z    GET SYM TBL ID WORD
616           LDX  I2 SYMT1 1    LOAD SYM TBL ID WORD ADDR
617           LD    2 2          LOAD 2ND WORD OF NAME
618           SRT     15         PUT IN EXTENSION
619           LD    2 1          LOAD FIRST WORD OF NAME
620           SLA     1          REMOVE ALPHA-INDICATOR BIT
621           SRA     1          RESTORE WORD
622           SRT     1          RIGHT JUSTIFY NAME
623           STO  L  FNAME      STORE FIRST WORD OF NAME
624           RTE     16         PUT 2ND WORD OF NAME IN ACC
625           STO  L  FNAME 1    STORE 2ND WORD
626   *
627   *              MAKE SYM TBL ID WORD ,VARIABLE,
628           LD    2 0          LOAD SYM TBL ID WORD
629           AND   3 H4020-Z    CLEAR ALL BUT REAL/INTEGER
630   *                          AND DEFINED BITS IN ID WORD
631           STO   2 0          STORE ID WORD IN SYM TBL
632   *
633   *              IF SUBR, MAKE SYM TBL ENTRY LOOK
634   *              LIKE REFERENCED GENERATED LABEL
635   *              INSTEAD   THIS PSEUDO-ENTRY WILL NOT
636   *              CAUSE CONFUSION IN FOLLOWING
637   *              PHASES BUT WILL BE PASSED BY
638   *              WITHOUT ACTION
639   *
640           LD   L  SORF       LOAD SORF WORD
641           BSC  L  EXIT,-     BRANCH IF FUNCTION
642   *
643           LD    3 H0220-Z    LOAD REFERENCED STMNT NO, I
644           STO   2 0          PUT IN SYM TBL
645           LD    3 ZERO-Z     LOAD ZERO
646           STO   2 1          PUT IN
647           STO   2 2          AS NAME
648           BSC  L  EXIT       GO TO NEXT PHASE
649   *
650   *              RETAINS THE CURRENT STMNT IN THE
651   *              STRING UNALTERED
652   Q1041 LDX   I1 INID        RESET I/P STRING PT
653           LDX  I2 OUTID      RESET O/P STRING PT
654           LD    1 0          LOAD STMNT ID WORD
655           AND   3 H07FC-Z    EXTRACT
656           STO   3 NORM-Z     NORM
657   Q1042 LD    1 0            AND STORE
658           STO   2 0          MOVE ONE WORD
659           MDX  L  NORM,-4    TO O/P STRING
```

```
660          MDX      Q1043     DECREMENT NORM BY ONE
661   *                           TEST IF LAST OUTPUT STMT
662   *                           WAS END STMNT
663          LD     3 STTYP-Z   CONTINUE LOOP
664          S      3 TEND-Z    LOAD STMNT ID TYPE
665          BSC  L   Q1021,Z   IS IT END STMNT
666   *                           BRANCH IF NOT
667   Q1044  STX  L2 EOFS       NEW END OF STRING ADDRESS
668   *
669   *                 TEST THAT SUBPROGRAM CONTAINS AT
670   *                 LEAST ONE RETURN-STATEMENT
671   *
672          LD     L  SORF     LOAD SORF
673          BSC  L   RETRN,Z   BRANCH UNLESS MAINLINE
674          LD     3 RETCT-Z   DOES PROGRAM HAVE RETURN
675          BSC  L   P1044, -  STMNT - BRANCH IF NOT
676          MDX      ERRRR     ERROR IF YES
677   RETRN  LD     3 RETCT-Z   DOES PROGRAM HAVE RETURN
678          BSC  L   P1044,Z   STMNT - BRANCH IF YES
679   ERRRR  LD     3 HA008-Z   LOAD ERROR STMNT ID WORD
680          STO    2 0         PUT ON OUTPUT STRING
681          LD     3 C69-Z     LOAD ERROR NO,
682          STO    2 1         PUT ON OUTPUT STRING
683          LD     1 0         LOAD END STMNT
684          STO    2 2         OUTPUT
685          MDX    2 2         MOVE OUTPUT PT
686   *
687   P1044  STX  L2 EOFS       NEW END OF STRING ADDR
688   *
689   *
690          BSC  L   TMTE      BRANCH TO TERMINATE PHASE
691   *
692   *
693   Q1043  MDX    1 1         MOVE I/P PT
694          MDX    2 1         MOVE O/P PT
695          MDX      Q1042     CONTINUE LOOP
696   *
697   *                 OUTPUTS THE NAME AND PARAMETERS FOR
698   *                 STMNT FUNCTION STATEMENTS
699   *
700   Q1051  BSI    3 MOVAW-Z   OUTPUT WORD, MOVE PTS
701          LD     1 0         LOAD WORD
702          BSC  L   Q1051, Z  BRANCH NOT OPERATOR
703   *
704   *                 SAVE OUTPUT POINTER   1
705   *                 AT TEMPP
706   *
707   Q1061  STX  L2 TEMPP      SAVE OUTPUT PT
708          MDX  L TEMPP,1     MOVE O/P PT
709   *
710   *                 IDENTIFY NEXT OPERATOR
711   *
712   *
713   Q2011  LD     1 0         LOAD WORD
714          BSC  L   Q2041, -  BRANCH IF SEMI-COLON
715          S      3 DIOOP-Z   IS IT DISK I/O OPERATOR
716          BSC  L   Q2009,Z   BRANCH IF NOT
717          LD     2 0         LOAD LAST O/P WORD
718          S      3 EXPRO-Z   IS IT EXPRESSION OPTR
719          BSC  L   Q2012, -  BRANCH IF YES
```

```
720   *
721           LD     1 2          LOAD RECORD NUMBER
722           AND    3 H7800-Z    IS IT SUBSCRIPTED
723           BSC  L   D1014, -   BRANCH IF NOT
724   *
725   Q2012   MDX    1 1          MOVE I/P PT
726           BSI    3 SCHP1-Z    LOAD LITERAL SUBSC OFFSET
727           MDX    1 -1         DECREMENT I/P PT
728           LD     3 H6100-Z    LOAD ,MDX L1,
729           BSI    3 OUTP-Z     OUTPUT
730           LD     1 2          LOAD DIMENSION
731           AND    3 H87FF-Z    REMOVE SUBSCRIPT BITS
732           STO    1 2          RESTORE
733   *
734           BSI    3 OUTP-Z     OUTPUT
735           LD     3 H6000-Z    LOAD NOP  MDX 0
736           BSI    3 OUTP-Z     OUTPUT NOP
737           LD     3 H6200-Z    LOAD ,STX L1,
738           BSI    3 OUTP-Z     OUTPUT
739           LD     3 H8003-Z    LOAD
740           BSI    3 OUTP-Z     OUTPUT
741           BSC  L   D1014      OUTPUT EXPRESSION OPERATOR
742   *
743   *
744   Q2008   LD     1 2          LD RECD NO,
745           AND    3 H7800-Z    IS IT SUBSCRIPTED
746           BSC  L   D1021, -   BRANCH IF NOT
747           LD     3 EXPRO-Z    OUTPUT EXPRESSION OPERATOR
748           BSI    3 OUTP-Z     *
749           MDX      Q2012      CONTINUE
750   *
751   Q2009   LD     1 0          LOAD WORD
752   *
753           S      3 H002E-Z    IS IT CALL OPERATOR
754           BSC  L   Q4011, -   BRANCH IF YES
755           A      3 H0016-Z    IS IT LITERAL SUBSC OPTR
756           BSC  L   Q3051, -   BRANCH IF YES
757           BSC  L   Q2013,     IS IT SPECIAL PARENTHESIS
758           S      3 H0008-Z    FOR DIMENSIONED ARRAYS
759           BSC  L   Q3011,Z    BRANCH IF YES
760           BSC  L   Q2021, -   BRANCH IF UNARY MINUS
761   Q2013   LD     1 0          LOAD WORD
762           S      3 H0014-Z    IS IT IF OPERATOR
763           BSC  L   Q2031,Z    BRANCH IF NOT
764   *
765   *                 IF OPERATOR ENCOUNTERED
766   *                 TEST IF NEXT WD IS STMNT NUMBER
767           LD     1 1          LOAD NEXT WORD
768           BSI    3 SYMT-Z     GET SYM TBL ID WORD
769           AND    3 H0200-Z    IS IT STMNT NO,
770           BSC  L   MOVE4,Z    BRANCH IF YES
771           BSI    3 SCHP1-Z    LOAD LITERAL SUBSC OFFSET
772           BSI    3 MOVAW-Z    OUTPUT WORD, MOVE POINTERS
773   MOVE4   BSI    3 MOVAW-Z    OUTPUT WORD, MOVE POINTERS
774   MOVE3   BSI    3 MOVAW-Z    OUTPUT WORD, MOVE POINTERS
775   MOVE2   BSI    3 MOVAW-Z    OUTPUT WORD, MOVE POINTERS
776   MOVE1   BSI    3 MOVAW-Z    OUTPUT WORD, MOVE POINTERS
777           MDX      Q2011      IDENTIFY NEXT OPERATOR
778   *
779   *                 UNARY MINUS ENCOUNTERED
```

```
780   *                TEST IF NEXT WORD IS OPERATOR
781   Q2021 LD     1 1         LOAD NEXT WORD
782         BSC  L  MOVE1,-     BRANCH IF NEXT WORD OPERATO
783   *
784   *
785   Q2022 BSI    3 SCHP1-Z   LOAD LITERAL SUBSC OFFSET
786         MDX      MOVE2     MOVE WORD
787   *
788   *                .OTHER OPERATOR, ENCOUNTERED
789   Q2031 LD     1 2         LOAD WORD
790         BSC  L  Q2022,-     BRANCH IF OPERATOR
791         LD     1 0         LOAD WORD
792         S      3 H000E-Z   IS IT ASSIGN
793         BSC  L  Q2032,Z     BRANCH IF NOT
794   *
795   *                .ASSIGN, FOLLOWED BY TWO NAMES
796   *                ENCOUNTERED, EXCHANGE WORDS AT
797   *                POINTER 1 AND POINTER 2
798   *
799         LD     1 1         LOAD FIRST WORD
800         RTE      16        PUT IN EXTENSION
801         LD     1 2         LOAD 2ND WORD
802         STO    1 1         PLACE IN FIRST WORD LOCATIO
803         RTE      16        PUT 1ST WORD IN ACCUMULATOR
804         STO    1 2         PLACE IN 2ND WORD LOCATION
805   Q2032 BSI    3 SCHP1-Z   SUBSCR CHECK, PTR 1
806         BSI    3 MOVAW-Z   OUTPUT WORD, MOVE POINTERS
807         BSI    3 MOVAW-Z   OUTPUT WORD, MOVE POINTERS
808         BSI    3 SCHP2-Z   LOAD LITERAL SUBSC OFFSETS
809         MDX      MOVE1     MOVE WORD
810   *
811   *                SEMICOLON ENCOUNTERED
812   *                MOVE LAST WORD AND TEST SWITCH
813   *
814   Q2041 BSI    3 MOVAW-Z   OUTPUT WORD, MOVE POINTERS
815         MDX    1 -1        DECREMENT I/P PT
816         LD     3 STXSW-Z   TEST STX SWITCH
817         BSC  L  Q5011,Z     BRANCH IF SET
818         BSC  L  Q1021       GO TO NEXT STMNT
819   *
820   *                SUBSCRIPT OPERATOR ENCOUNTERED
821   *
822   Q3011 BSI        SBSN    PROCESS SUBSCR OPTR
823         MDX        Q2011   IDENTIFY NEXT OPERATOR
824   *
825   *                LITTERAL SUBSC OPERATOR ENCOUNTERED
826   Q3051 BSI        SBSLT   PROCESS LITERAL SUBSCRIPT
827         MDX        Q2011   IDENTIFY NEXT OPERATOR
828   *
829   *
830   *                GENERATES THE CALL FOR SUBSCRIPT
831   *                CALCULATION WITH ARGUMENTS
832   *                MAKES THE SGT TBL ENTRY
833   *
834   SBSN  DC         0       ENTRY POINT
835         A      3 H0008-Z   GET DIMENSION COUNT
836         STO    3 DIMCT-Z   STORE DIMENSION COUNT
837         LD     3 H1900-Z   LOAD ,CALL SUBSC,
838         BSI    3 OUTP-Z    OUTPUT CALL
839         LD     1 1         LOAD WORD
```

```
840          AND    3 H87FF-Z   GET SYM TBL ENTRY NO,
841          BSI    3 OUTP-Z    OUTPUT
842          LD     1 1         LOAD WORD
843          AND    3 H7800-Z   GET PSEUDO INDEX
844          SRA      11        RIGHT JUSTIFY
845          STO    3 CURR-Z    PUT IN CURR
846          LD     1 2         LOAD D4
847          SLA      1         DELETE
848          SRT      1         BIT 0
849          BSI    3 OUTP-Z    OUTPUT
850   *
851   *                INSERT ,NON-LITTERAL, TO SGT-TABLE
852          LD     1 1         LOAD WORD
853          BSI    3 SGTAD-Z   COMPUTE SGT TABLE ADDR
854          LD     3 H8000-Z   LOAD NON-LITERAL SGT WORD
855          STO  I   SGTI      PUT IN TABLE
856   *
857          MDX    1 3         MOVE I/P PT
858   *
859   *                OUTPUT REMAINING ARGUMENTS
860   SBS1   LD     1 0         LOAD WORD
861          BSI    3 OUTP-Z    OUTPUT
862          LD     1 1         LOAD WORD
863          BSI    3 OUTP-Z    OUTPUT
864          MDX    1 2         MOVE O/P PT
865          MDX  L   DIMCT,-2  DECREMENT DIMENSION COUNT
866          MDX      SBS1      OUTPUT REMAINING ARGUMENTS
867   *
868   *                TAG LAST ARGUMENT IN LIST
869          LD     2 0         LOAD LAST AGUMENT
870          OR     3 H8000-Z   TAG ARGUMENT
871          STO    2 0         PUT BACK IN O/P STRING
872   *
873          BSC  I   SBSN      RETURN
874   *
875   *                MAKES THE LITERAL SUBSCRIPT ENTRY
876   *                INTO THE SGT TABLE
877   *
878   SBSLT  DC       0         ENTRY POINT
879          LD     1 1         LOAD NAME
880          BSI    3 SGTAD-Z   COMPUTE ADDR IN SGT-TABLE
881          LD     1 2         LOAD D4
882          SLA      1         DELETE BIT 0
883          SRT      1
884          STO  I   SGTI      PUT IN SGT-TABLE
885          MDX    1 3         MOVE POINTER
886          BSC  I   SBSLT     RETURN
887   *
888   *                SCANS THE TWO-WORD CALL ARGUMENT LIS
889   *                FOR SUBSCRIPTED NAMES
890   *                GENERATES THE INSTRUCTIONS FOR
891   *                OBJECT-TIME ADDRESS INSERTION FOR
892   *                SUBSCRIPTED VARIABLES
893   *
894   Q4011  STX    1 Q4021 1   SAVE I/P PT
895   Q4012  LD     1 1         LOAD WORD
896          BSC  L   Q4021,-   BRANCH IF OPERATOR
897          STO    3 TEMP1-Z   SAVE ARGUMENT
898          AND    3 H7800-Z   GET PSEUDO INDEX BITS
899          BSC  L   Q4014, -  BRANCH IF NO PSEUDO INDEX
```

```
900          BSI    3 SQTAD-Z  COMPUTE SGT TABLE ADDRESS
901          BSI    3 SCHP1-Z  LOAD LITERAL SUBSC OFFSETS
902          LD     3 H6100-Z  LOAD ,MDX L1,
903          BSI    3 OUTP-Z   OUTPUT
904          LD     1 1        LOAD WORD
905          BSI    3 OUTP-Z   OUTPUT
906          LD     3 H6000-Z  LOAD ,NOP,  MDX 0
907          BSI    3 OUTP-Z   OUTPUT
908          LD     3 H6200-Z  LOAD ,STX L1,
909          BSI    3 OUTP-Z   OUTPUT
910   *
911   *                RESTORE WORD AT PT 1
912   *                WITH ITS PSEUDO INDEX
913   *
914          LD     3 TEMP1-Z  LOAD NAME
915          STO    1 1        PUT BACK ON STRING
916   *
917          BSI    3 OUTP-Z   OUTPUT WORD
918   *                SERVES LATER AS IDENTIFICATION
919   *                AT STX L1-SCAN
920   *                ZERO ,CURRENT SGT, *
921   *                 SGT DESTROYED AT OBJECT TIME
922   *                BY MDX L1-INSTR
923   *
924          SLA      16       CLEAR ACCUMULATOR
925          STO    3 CURR-Z   CLEAR CURR
926          LD     3 H8000-Z  SET CURRENT D4
927          STO    3 CURD4-Z  TO NON-LITERAL
928          STO  I   SGTI     SET LAST SGT TO NON-LITERAL
929          MDX  L   STXSW,1  SET STX-SWITCH
930   *
931   Q4014  MDX    1 1        MOVE POINTER
932          MDX      Q4012    PROCESS SUBSCRIPTED ARGUMEN
933   *
934   *                OUTPUTS THE 2-WORD SUBP CALL WITH
935   *                ARGUMENTS
936   *
937   Q4021  LDX  L1 *-*        RESTORE POINTER
938   Q4022  BSI    3 MOVAW-Z  OUTPUT WORD, MOVE POINTERS
939          LD     1 0        LOAD WORD
940          BSC  L   XXXXX,-  BRANCH IF END OF ARGUMENTS
941   *                        - OPERATOR ENCOUNTERED
942          MDX      Q4022    PROCESS NEXT ARGUMENT
943   *
944   *                COMPUTES DISPLACEMENT VALUES FOR
945   *                 STX  L1  INSTRUCTIONS USED FOR
946   *                STORING THE COMPUTED SUBSCRIPT OFFSE
947   *                INSERTS THOSE VALUES INTO THE SECOND
948   *                WORD OF THE INSTRUCTION
949   Q5011  STX    1 Q5015 1  SAVE POINTER
950          LDX   I1 TEMPP    LOAD TEMPORARY POINTER
951   Q5013  LD     1 0        LOAD WORD
952          BSC  L   Q5021,Z  BRANCH IF NOT SEMI-COLON
953   Q5015  LDX  L1 *-*        RESTORE POINTER
954          BSC  L   Q1021    PROCESS NEXT STMNT
955   *
956   Q5021  BSC  L   Q5031,-  BRANCH IF NOT NAME
957   *
958   Q5022  MDX    1 1        MOVE POINTER
959          MDX      Q5013    CHECK FOR END OF STMNT
```

```
960   *
961   Q5031 S      3 H1900-Z  IS WORD ,CALL SUBSCR,
962         BSC  L  Q5035,Z   BRANCH IF NOT
963         BSI    3 PASSA-Z  MOVE POINTER PAST ARGUMENTS
964         MDX      Q5013    CHECK FOR END OF STMNT
965   Q5035 LD     1 0        LOAD WORD
966         S      3 H6200-Z  IS IT  ,STX L1,
967         BSC  L  Q5041, -   BRANCH IF YES
968         S      3 DIFF1-Z  IS IT ,LDX L1,
969         BSC  L  Q5022,Z   BRANCH IF NOT
970         MDX    1 1        MOVE POINTER
971         MDX      Q5022    MOVE PT, CHECK NEXT WORD
972   *
973   *               COMPUTE Z IN INSTRUCTION
974   *               STX L1 Z
975   *               AND INSERT ON STRING
976   *
977   Q5041 SLA       16       CLEAR EXTERNAL COUNTER
978         STO    3 COUNT-Z
979         MDX    1 1        MOVE POINTER
980         STX   L1 TEMPX    SAVE POINTER
981   *
982   *               SEARCH FOR IDENTIFIER  IN
983   *               CALL-ARGUMENT LIST
984   *
985   Q5042 MDX    1 1        MOVE POINTER
986   Q5043 LD     1 0
987         S      3 H1900-Z  IS WORD ,CALL SUBSC,
988         BSC  L  Q5045,Z   BRANCH IF NOT
989         BSI    3 PASSA-Z  MOVE PT PAST ARGUMENTS
990         MDX      Q5043    CHECK NEXT WORD
991   *
992   Q5045 LD     1 0
993         S      3 H5B00-Z  IS WORD ,LDX L1,
994         BSC  L  Q5046,Z   BRANCH IF NOT
995         MDX    1 1        MOVE POINTER
996         MDX      Q5042    SEARCH FOR IDENTIFIER
997   *
998   Q5046 LD     1 0        LOAD WORD
999         BSC  L  Q5047,+Z   BRANCH IF NOT OPERATOR V1M
1000        S      3 H002E-Z   IS IT A CALL OPERATOR  V1M
1001        BSC  L  Q5042,Z   BRANCH NO               V1M
1002        MDX      Q5045+4  BRANCH YES              V1M
1003  Q5047 BSI      GETID    GET ID                   V1M
1004        SLA      11
1005        BSC  L  CONTU,-   BRANCH IF NOT EXTERNAL
1006        MDX  L  COUNT,1   INCREMENT EXTERNAL COUNTER
1007        MDX      Q5042    GO MOVE POINTER
1008  CONTU LD     1 0        RELOAD STRING ARGUMENT
1009        S    I  TEMPX     IS IT ADDR IN STX L1 INSTR
1010        BSC  L  Q5042,Z   BRANCH IF NOT
1011  *
1012  *               REMOVE PSEUDOINDEXBITS FROM
1013  *               CALL-ARGUMENT WORD
1014  *
1015        LD     1 0        LOAD WORD
1016        AND    3 H87FF-Z  REMOVE PSEUDO INDEX BITS
1017        STO    1 0        PUT BACK ON STRING
1018  *
1019  *               COMPUTE AND INSERT SECOND HALF
```

```
1020  *              OF STX L1-INSTRUCTION
1021  *
1022         STX   L1 TEMP1     SAVE POINTER
1023         LD     3 TEMP1-Z   LOAD ADDRESS OF PT
1024         S      3 TEMPX-Z   SUBTRACT STORED PT
1025         OR     3 H8000-Z   PUT IN SIGN BIT
1026         A      3 COUNT-Z   ADD EXTERNAL COUNT
1027         STO   I  TEMPX     PUT IN STX  L1 INSTRUCTION
1028         LDX   I1 TEMPX     RESTORE POINTER
1029         MDX      Q5022     MOVE PT, CHECK NEXT WORD
1030  *
1031  *
1032  *              SGT-TABLE
1033  *
1034  SGTB   DC       /8000     NON-LITERAL CONSTANT
1035         DC       /8000     NON-LITERAL CONSTANT
1036         DC       /8000     NON-LITERAL CONSTANT
1037         DC       /8000     NON-LITERAL CONSTANT
1038         DC       /8000     NON-LITERAL CONSTANT
1039         DC       /8000     NON-LITERAL CONSTANT
1040         DC       /8000     NON-LITERAL CONSTANT
1041         DC       /8000     NON-LITERAL CONSTANT
1042         DC       /8000     NON-LITERAL CONSTANT
1043         DC       /8000     NON-LITERAL CONSTANT
1044         DC       /8000     NON-LITERAL CONSTANT
1045         DC       /8000     NON-LITERAL CONSTANT
1046         DC       /8000     NON-LITERAL CONSTANT
1047         DC       /8000     NON-LITERAL CONSTANT
1048         DC       /8000     NON-LITERAL CONSTANT
1049  *
1050  *
1051  *              GETS THE ID WORD OF A SIM TBL ENTRY
1052  *
1053  SYMT   DC       0         ENTRY POINT
1054  GETID  EQU      SYMT
1055         AND    3 H07FF-Z   GET SYM TBL ENTRY NO.
1056         S      3 ONE-Z
1057         STO      SYMT1 1   SAVE ENTRY NO.
1058         LD     L SOFST     LOAD START OF SYM TBL ADDR
1059  *
1060         S        SYMT1 1   GET ADDRESS OF ENTRY
1061         S        SYMT1 1   BY MULTIPLYING ENTRY NO. BY
1062         S        SYMT1 1   THREE AND SUBTRACTING IT
1063         STO      SYMT1 1   FROM TABLE ORIGIN
1064  *
1065  SYMT1  LD     L  *-*       LOAD SYM TBL ID WORD
1066         BSC    I  SYMT      RETURN
1067  *
1068  *              SUBROUTINE
1069  *              MOVES THE POINTER PAST THE
1070  *              ARGUMENTS OF  LIBF SUBSC
1071  *
1072  PASSA  DC       0         ENTRY POINT
1073         MDX    1 3         MOVE POINTER
1074  PASS1  MDX    1 2         MOVE POINTER
1075         LD     1 -1        RETURN IF LAST
1076         BSC    I PASSA,Z   ARGUMENT TAGGED
1077         MDX      PASS1     BRANCH IF NOT
1078  *
1079  *
```

```
1080  *              SUBROUTINE
1081  *              MOVE WORD AT I/P PT TO O/P STRING
1082  *              INCREMENT POINTERS
1083  MOVAW DC       0          ENTRY POINT
1084        LD    1  0          LOAD WORD
1085        BSI   3  OUTP-Z     OUTPUT WORD, MOVE O/P PT
1086        MDX   1  1          MOVE I/P PT
1087        BSC   I  MOVAW      RETURN
1088  *
1089  *              SUBROUTINE
1090  *              OUTPUT
1091  *              MOVES CONTENTS OF ACC TO
1092  *              OUTPUT STRING, INCR NORM AND
1093  *              CHECKS IF OVERLAP ERROR
1094  *
1095  OUTP  DC       0          ENTRY POINT
1096        STO   2  1          OUTPUT WORD
1097        MDX   2  1          MOVE O/P PT
1098        LD    I  OUTID      LOAD STMNT I/D WORD
1099        A     3  FOUR-Z     INCREMENT NORM BY ONE
1100        STO   I  OUTID      PUT ID WORD ON O/P STRING
1101        STX   1  OUTP3
1102        STX   2  OUTP4      CHECK FOR
1103        LD       OUTP3      OVERLAP
1104        S        OUTP4      ERROR
1105        S     3  TWO-Z
1106        BSC   I  OUTP,Z-    RETURN IF NO OVERLAP ERROR
1107        BSC   L  OVLAP      OVERLAP ERROR
1108  OUTP3 DC       0          TEMPORARY STORAGE OF PT
1109  OUTP4 DC       0          TEMPORARY STORAGE OF PT
1110  *
1111  *
1112  *              SUBROUTINE
1113  *              CONVERTS PSEUDO INDEX IN ACC
1114  *              INTO SGT-TABLE ADDRESS,
1115  *              STORE AT ,SGTI,
1116  *
1117  SGTAD DC       0          ENTRY POINT
1118        AND   3  H7800-Z    GET PSEUDO INDEX
1119        SRA      11         RIGHT JUSTIFY INDEX
1120        A     3  ADRC-Z     GET ACTUAL
1121        STO      SGTI       SGT-TABLE ADDRESS
1122        BSC   I  SGTAD      RETURN
1123  SGTI  DC       0          COMPUTED SGT-TABLE ADDRESS
1124  *
1125  *
1126  *              GENERATES THE  LDX  INSTRUCTION
1127  *              USING THE SGT TABLE
1128  *
1129  *
1130  *              ENTRY FOR ARGUM   POINTER   0
1131  *
1132  SCHP2 DC       0          ENTRY PT
1133        LD       SCHP2      MOVE RETURN ADDRESS
1134        STO      SCHP1      TO SCHP1
1135        LD    3  ZERO-Z     LOAD ZERO
1136        MDX      SCH11      CONTINUE
1137  *
1138  *              ENTRY FOR ARGUM   POINTER   1
1139  *
```

```
1140  SCHP1 DC      0          ENTRY POINT
1141        LD    3 ONE-Z      LOAD ONE
1142  SCH11 STX   1 ADDR       SAVE INPUT POINTER
1143        A       ADDR       MODIFY INPUT POINTER
1144        STO     ADDR       SAVE ADDR
1145        LD    I ADDR       LOAD WORD AT I/P PT   1
1146        AND   3 H7800-Z    GET PSEUDO INDEX
1147        BSC   I SCHP1, -   RETURN IF NO PSEUDO INDEX
1148        SRA     11         RIGHT JUSTIFY
1149        S     3 CURR-Z     IS IT CURRENT SGT
1150        BSC   L SCH15, -   BRANCH IF YES
1151        A     3 CURR-Z     RESTORE WORD
1152        STO   3 CURR-Z     PUT IN CURRENT SGT
1153        LD    I ADDR       LOAD WORD
1154        BSI   3 SGTAD-Z    COMPUTE SGT-TABLE ADDRESS
1155        LD    I SGTI       LOAD ADDRESS
1156        S     3 H8000-Z    IS IT NON-LITERAL SGT
1157        BSC   L SCH21, -   BRANCH IF YES
1158  *
1159  *               LITERAL SUBSCRIPT ENCOUNTERED
1160  *               TEST IF IT MATCHES WITH
1161  *               CURRENT LITERAL D4
1162  *
1163  *               OUTPUT ,LDX L1, AND D4
1164        LD    3 H5B00-Z    LOAD ,LDX L1,
1165        BSI   3 OUTP-Z     OUTPUT
1166        LD    I SGTI       LOAD SGT TABLE WORD
1167        STO   3 CURD4-Z    STORE AS CURRENT D4
1168        BSI   3 OUTP-Z     OUTPUT WORD
1169  *                        *****  LDX L1  SGT TBL WD
1170  *               DELETE PSEUDO INDEX-BITS
1171  *               FROM STRING WORD, RETURN
1172  *
1173  SCH15 LD    I ADDR       LOAD WORD
1174        AND   3 H87FF-Z    DELETE PSEUDO INDEX BITS
1175        STO   I ADDR       RESTORE WORD
1176        BSC   I SCHP1      RETURN
1177  *
1178  *               OUTPUT ,LDX I1, AND SGT
1179  SCH21 LD    3 H4800-Z    LOAD ,LDX  I1,
1180        BSI   3 OUTP-Z     OUTPUT ,LDX I1,
1181        LD    L SOFST      LOAD START OF SYM TBL ADDR
1182        S     L SOFXT      SUBTRACT START OF SUBSCRIPT
1183        SRT     16         TEMPORARIES
1184        D     3 THREE-Z    DIVIDE BY THREE
1185        A     3 CURR-Z     ADD CURRENT SGT
1186        OR    3 H8000-Z    PUT SIGN BIT FOR ADDRESS
1187        BSI   3 OUTP-Z     *****  LDX I1  ADDRESS
1188        LD    3 H8000-Z    INSERT NON LITERAL
1189        STO   3 CURD4-Z    CURRENT D4
1190        MDX     SCH15      DELETE PSEUDO INDEX WORDS
1191  ADDR  DC      0          ADDRESS STORAGE
1192  *
1193  USWIT DC      0          UNFORMATTED I/O SWITCH
1194  ZERO  DC      0          ZERO
1195  Z     EQU     ZERO
1196  ONE   DC      1          ONE
1197  TWO   DC      2          TWO
1198  THREE DC      3          THREE
1199  FOUR  DC      4          FOUR
```

```
1200  *
1201  LDXI1 DC       /C000     ,LDX  L1,
1202  FGOTO DC       /0F80      CALL GO TO TRACE
1203  COMGO DC       /0D80      CALL COMPUTED GOTO SUBR
1204  LOP   DC       /0028     LIST-OPERATOR
1205  H1800 DC       /1800     DIMENSION MASK - SYM TBL ID
1206  *
1207  FIOIX DC       /1700     ,CALL SIOIX,
1208  FIOFX DC       /1680     ,CALL SIOFX,
1209  FIOI  DC       /1880     ,CALL SIOI,
1210  FIOF  DC       /1800     ,CALL SIOF,
1211  FIOAI DC       /1580     ,CALL SIOAI,
1212  FIOAF DC       /1600     ,CALL SIOAF,
1213  GENLO DC       /5E01     STORED GENERAL LABEL OPTR
1214  DOAOP DC       /002A     DOA-OPERATOR
1215  DOTOP DC       /004A     DO TEST OPERATOR
1216  AL    DC       /9800     ,A    L,
1217  STOL  DC       /9000     ,STO  L,
1218  SL    DC       /A000     ,S    L,
1219  MDXL1 DC       /5101     ,MDX  L1,
1220  LDL   DC       /8800     ,LD   L,
1221  H0018 DC       /0018     LITERAL OPERATOR
1222  HA008 DC       /A008     ERROR STMNT ID WORD
1223  C69   DC       69        ERROR NO, 69
1224  H5F00 DC       /5F00     ,BLANK,
1225  *
1226  *              SUBROUTINE
1227  *              OUTPUT CONTENTS OF ACC AND WORD AT
1228  *              POINTER   1 WITH DISK I/O TEST
1229  *
1230  OUTIO DC       0         ENTRY POINT
1231        BSI    3 DTEST-Z   OUTPUT CALL
1232        LD     1 1         LOAD WORD
1233        AND    3 H87FF-Z   DELETE PSEUDO INDEX BITS
1234        BSI    3 OUTP-Z    OUTPUT WORD
1235        BSC  I   OUTIO     RETURN
1236  *
1237  *              SUBR  OUTPUT ,LD L, PTR 1
1238  OULDL DC       0         ENTRY POINT
1239        LD     1 1         LOAD WORD
1240        AND    3 H07FF-Z   GET SYM TBL ENTRY NO,
1241        A      3 LDL-Z     COMBINE WITH ,LD L,
1242        BSI    3 OUTP-Z    OUTPUT INSTRUCTION
1243        BSC  I   OULDL     RETURN
1244  FLD   DC       /0480     ,CALL FLD,
1245  FRED  DC       /1400     ,CALL FRED,
1246  FWRT  DC       /1480     ,CALL FWRT,
1247  FCOMP DC       /1500     ,CALL FCOMP,
1248  FIND  DC       /2680-/2180 /1400  ,CALL FIND,
1249  *                         CALL FIND ,
1250  *                        CHANGED LATER TO /2680
1251  *                        BY  DTEST
1252  *
1253  FNDID DC       /E800      ,FIND, ID
1254  *
1255  RETCT DC       0         RETURN-COUNT
1256  BSCI  DC       /5080     ,BSC I,
1257  H5080 EQU      BSCI
1258  H4020 DC       /4020     DEFINED INTEGER VAR ID
1259  H0220 DC       /0220     REFERENCED STMNT NO, ID
```

```
1260  *
1261  H002E DC      /002E     CALL OPERATOR
1262  H0016 DC      /0016     ,CALL,
1263  H0008 DC      /0008     CONSTANT FOR DIM COUNT
1264  H000E DC      /000E     ASSIGN-OPERATOR
1265  H0200 DC      /0200     STMNT NO, IN SYM TBL ID WD
1266  H1900 DC      /1900     ,CALL SUBSCR,
1267  H7800 DC      /7800     MASK TO GET PSEUDO INDEX
1268  H87FF DC      /87FF     MASK TO ELIMINATE PSEUDO ID
1269  H07FC DC      /07FC     MASK TO GET STMNT NORM
1270  H07FF DC      /07FF     MASK TO GET ADDR
1271  ADRC  DC      SGTB-1    MASK FOR SYM TBL ENTRY NO,
1272  H0014 DC      /0014     IF OPERATOR
1273  DIFF1 DC      /5B00-/6200   ,LDX  L1,
1274  H8000 DC      /8000     NON-LITERAL SGT
1275  H8003 DC      /8003     USEFUL CONSTANT
1276  H6100 DC      /6100     ,MDX  L1,
1277  H6000 DC      /6000     ,NOP,     MDX 0
1278  H6200 DC      /6200     ,STX  L1,
1279  H5B00 DC      /5B00     ,LDX  L1,
1280  H4800 DC      /4800     ,LDX  I1,
1281  HF800 DC      /F800     ID TYPE MASK
1282  HF803 DC      /F803     MASK TO ELIMINATE STMNT NOR
1283  H1000 DC      /1000     END STMNT ID TYPE
1284  TEND  EQU     H1000
1285  *
1286  STTYP DC      0         STATEMENT TYPE
1287  OUTID DC      0         ADDRESS OF O/P STMNT ID WOR
1288  INID  DC      0         ADDRESS OF I/P STMNT ID WOR
1289  NORM  DC      0         STMNT NORM
1290  STXSW DC      0         STX-SWITCH
1291  TEMPX DC      0         POINTER STORAGE
1292  TEMPP DC      0         POINTER STORAGE
1293  TEMP1 DC      0         POINTER STORAGE
1294  DIMCT DC      0         DIMENSION COUNT
1295  CURR  DC      0         CURRENT SGT
1296  SOFIS DC      0         START OF INPUT STRING
1297  SSQST DC      0         STORED START OF I/P STRING
1298  *
1299  CURD4 DC      0         CURRENT D4
1300  *                         /8000 IF NOT LITERAL
1301  *
1302  *
1303  *             NOT USED IN CARD SYSTEM
1304  *
1305  IOOPR DC      /0026     IO-OPERATOR
1306  DIOOP DC      /0030     DISK IO OPERATOR
1307  EXPRO DC      /0054     EXPRESSION OPERATOR
1308  DSWIT DC      0         DISK SWITCH
1309  TEMPY DC      0         TEMPORARY STORAGE
1310  COUNT DC      0         EXTERNAL COUNTER
1311  *
1312  *             SUBROUTINE
1313  *             OUTPUT, WITH TEST OF DISKSWITCH
1314  *
1315  DTEST DC      0         ENTRY POINT
1316        STO     DTES1     SAVE CALL
1317        LD    3 DSWIT-Z   IS IT DISK OPERATION
1318        BSC  L  ADI, -    BRANCH IF NOT
1319        LD      CDIFF     SET DISK NAME
```

```
1320            MDX       ADDN      TABLE OFFSET
1321   ADI      LD     3  USWIT-Z   IS IT UNFORMATTED I/O
1322            BSC    L  ADDN, -   BRANCH IF NOT
1323            LD        DTES1     CHECK THE NO, OF THE ENTRY
1324            S         H1500     TO SEE WHICH PART OF THE
1325            BSC    L  ADQ,Z-    TABLE IN O/P PHASE HOLDS TH
1326            LD        UDIF1     CORRESPONDING UNFORMATTED
1327            MDX       ADDN      I/O NAME AND PUT THE PROPER
1328   ADQ      LD        UDIF2     DISPLACEMENT
1329   ADDN     A         DTES1     ADD BASE ADDRESS
1330   *                            VALUE OF DTEST1 IS UNCHANGE
1331   *                            UNLESS DISK OR UNFORMATTED
1332   *                            I/O IS PRESENT
1333            BSI    3  OUTP-Z    OUTPUT WORD, MOVE O/P PT
1334            BSC    I  DTEST     RETURN
1335   *
1336   H1500    DC        /1500     CONSTANT FOR CHECKING TBL
1337   DTES1    DC        0         TEMPORARY STORAGE
1338   CDIFF    DC        /2180-/1400  DISK NAME TABLE
1339   UDIF1    DC        /1200-/1400  UNFORMATTED - PART 1
1340   UDIF2    DC        /1D80-/1580  I/O NAME TBL - PART 2
1341   *
1342   *
1343   *                    TRANSFERS TO THE COL ROUTINE TO LOAD
1344   *                    THE NEXT PHASE
1345   *
1346   EXIT     BSI    L  ROLRX      CALL DOWN PHASE 18
1347            DC        18              NEXT PHASE NUMBER
1348   *
1349   *
1350   *
1351            DC        ***       TO BE FILLED
1352   XXXXX    SLA       16
1353            STO    3  CURR-Z    CLEAR CURRENT SGT
1354            BSC    L  Q2011     GET NEXT OPERATOR
1355            BSS       OVERL-*+320*4  PHASE-17 PATCH AREA
1356            END       START
```