

```

1  *   GENERAL AUTOMATION, INC, ALL RIGHTS RESERVED
2  *****
3  *
4  *   PROGRAM NAME   FPH-15
5  *
6  *   MODEL NUMBER   8F015
7  *
8  *   PURPOSE        FORTRAN PHASE-15
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 ** SUBSCRIPT OPTIMIZE
22 *****
23 *STATUS -VERSION 1, MODIFICATION 0
24 *
25 *FUNCTION/OPERATION-
26 *   * SCANS READ, WRITE, IF, CALL, AND ARITHMETIC
27 *   STATEMENTS FOR SUBSCRIPT EXPRESSIONS
28 *   * OPTIMIZES SUBSCRIPT CALCULATION BY MEANS OF
29 *   THE SUBSCRIPT EXPRESSION TABLE
30 *   * GENERATES SUBSCRIPT GENERATED TEMPORARIES AS
31 *   NECESSARY
32 *
33 *ENTRY POINTS-
34 *   START-PHASE 15 IS READ INTO CORE BY PHASE 14
35 *   TRANSFER TO ROLRX, EXECUTION IS BEGUN
36 *   AT LOCATION START,
37 *
38 *INPUT-THE STATEMENT STRING MAY BE CONSIDERED
39 *   INPUT
40 *
41 *OUTPUT-THE STATEMENT STRING AND THE SYMBOL TABLE
42 *   AS ALTERED BY THE PHASE MAY BE CONSIDERED
43 *   OUTPUT
44 *
45 *EXTERNAL REFERENCES-N/A
46 *
47 *EXITS-
48 *   NORMAL-
49 *   AT LOCATION OUT TRANSFER IS MADE TO THE
50 *   INTERPHASE ROLLER ROUTINE WHICH LOADS
51 *   PHASE 16,
52 *   ERROR-
53 *   THE FLOW OF COMPILER PROCESSING IS NOT
54 *   INTERRUPTED WHEN AN ERROR IS ENCOUNTERED,
55 *   INSTEAD, AN ERROR STMT REPLACES THE FAULTY
56 *   STMT ON THE STRING AND THE STMT SCANNING
57 *   PROCESS CONTINUES, AN EXCEPTION OCCURS WHEN
58 *   AN OVERLAP ERROR IS DETECTED, COMPILATION
59 *   IS THEN ABORTED AND CONTROL IS PASSES FROM

```

```

60 *      PHASE TO PHASE, EACH PHASE IMMEDIATELY
61 *      EXITING TO THE NEXT UNTIL PHASE 21 PRINTS
62 *      THE OVERLAP ERROR MESSAGE.
63 *      THE ERROR DETECTED BY PHASE 15 IS NO 63
64 *
65 *TABLES/WORK AREAS-
66 *  STRING
67 *  SYMBOL TABLE
68 *  FORTRAN COMMUNICATIONS AREA
69 *  SUBSCRIPT EXPRESSION TABLE-SEE BELOW
70 *  BOUND VARIABLE TABLE-
71 *      ENTRIES ARE UNIQUE VARIABLES FOUND IN THE
72 *      SUBSCRIPT EXPRESSION AT HAND,
73 *  SUBSCRIPT GENERATED TEMPORARY TABLE-
74 *      FOR HOLD OF LOCAL SGIS
75 *
76 *ATTRIBUTES-N/A
77 *
78 *NOTES-
79 *  THE SWITCHES USED IN PHASE 15 FOLLOW, IF NON-
80 *  ZERO, THE SWITCH IS TRANSFER 1, IF ZERO, THE
81 *  SWITCH IS NORMAL.
82 *      *SW1-T PURGE FLTNG P1 NON-DIM INTGR VBLE
83 *      *SW2-T I/O FOUND
84 *      *SW3-T PUT FP NON-D IVAR IN BOUND VBLE TBL
85 *      *SW4-T IN FUNCTION CALL
86 *      *SW5-T DO NOT OPTIMIZE
87 *      *SW6-T CLOSE STRING
88 *
89 *-----*
90 *      ABS REF CORE
91 *
92 *
93 *      MEMRY EQU 77777 CORE      MAXIMUM CORE SIZE
94 *      PHSIZ EQU 4*320      MAXIMUM PHASE SIZE
95 *      OVERL EQU MEMRY-PHSIZ      PHASES 2-29 START
96 *      FCOM EQU OVERL-22      FORTRAN COMM. TABLE
97 *      PHNTB EQU FCOM-56      PHASE TABLE
98 *      ROLRX EQU PHNTB-50      INTERPHASE CALL
99 *
100 *      FORTRAN COMMUNICATION AREA
101 *
102 *      ORG      FCOM
103 *      SOFS BSS 1      START OF STRING
104 *      EOFS BSS 1      END OF STRING
105 *      SOFST BSS 1      START OF SYMBOL TABLE
106 *      SOFNS BSS 1      START OF NON-STATEMENT NUMBERS
107 *      SOFXT BSS 1      START OF SUBSCRIPT TEMPORARIES
108 *      SOFGT BSS 1      START OF GENERATED TEMPORARIES
109 *      EOFST BSS 1      END OF SYMBOL TABLE
110 *      COMON BSS 1      NEXT AVAILABLE COMMON
111 *      CSIZE BSS 1      SIZE OF COMMON
112 *      ERROR BSS 1      OVERLAP ERROR
113 *      FNAME BSS 1      PROGRAM
114 *      BSS 1      NAME
115 *      SORF BSS 1      SUBROUTINE - OR FUNCTION
116 *      CCWD BSS 1      CONTROL CARD WORD
117 *      BIT 15 TRANSFER TRACE
118 *      BIT 14 ARITHMETIC TRACE
119 *      BIT 13 EXTENDED PRECISION

```

```

120 *          BIT 12 LIST SYMBOL TABLE
121 *          BIT 11 LIST SUBPROGRAM NAMES
122 *          BIT 10 LIST SOURCE PROGRAM
123 *          BIT 9 ONE WORD INTEGERS
124 IOCS  BSS      1    IOCS CONTROL CARD WORD
125 *
126 *          SEE PHASE ONE FOR BIT PATTERNS
127 *
128 DFCNT BSS      1    DEFINE FILE COUNT
129 LCOMN BSS      2    SIZE OF INSKEL COMMON
130 ICCER BSS      2    IOCS CONTROL CARD ERROR
131      BSS      2    SYSTEM LOADER USE
132 *          END OF FORTRAN COMM AREA
133 *
134 *
135      ORG      OVERL
136 START LD        ERROR      EXIT TO NEXT PHASE
137      BSC  L    OUT,Z      IF OVERLAP ERROR
138 *
139 *          INITIALIZE PHASE
140      LDX  I1  SOFS      XR1 PTS TO START OF STRING
141      LDX  L2  SETBL     PI XR2 TO SUBSCR EXPR TABL
142      LDX   3  15
143      SLA   16          NORMALIZE
144      STO  L  SW1      FF NON-D INTEGER SWITCH
145      STO  L  SW2      I/O FOUND SWITCH
146      STO  L  SW3      FF NON-D INTEGER VBLE SW
147 LOZZ  STO  2  7      ENTRY NOT USED PREVIOUSLY
148      STO  L3  GTTAB-1  SET
149      MDX  2  8          FOR
150      MDX  3  -1       EACH
151      MDX   LOZZ      SGT TABLE ENTRY
152      STO  L  BVTAB    SET BVTAB INDICATOR EMPRY
153 *
154 *          CHECK STATEMENT TYPE
155 TEST  LD        1  0    CHECK
156      SRA        1      FOR
157      AND      IDTPE    END
158      S        ENDC     STATEMENT
159      BSC  L    OUT, -   DONE, IF SO
160 *
161      S        FORMM    IF FORMAT STATEMENT
162      BSC  L    MOVE, -  MOVE TO NEXT
163      S        DFL      IF DEFINE FILE STATEMENT
164      BSC  L    MOVE, -  MOVE TO NEXT
165      S        CLE      IF CALL LINK/EXIT STMT
166      BSC  L    MOVE, -  MOVE TO NEXT
167      S        DATAS    IF DATA STATEMENT
168      BSC  L    MOVE, -  MOVE TO NEXT
169 *
170      LD        1  0    GO TO TEST1 IF STATEMENT
171      BSC  L    TEST1,E  HAS STATEMENT NUMBER
172 *
173 *          SET PRIORITY STATUS
174 TEST3 LDX  L2  SETBL
175      LDX   3  15
176 LOOP1 LD   2  7      GET SETBL ENTRY TYPE
177      BSC  L  SK, -    BR IF ENTRY USED PREVIOUSL
178      OR   L  MASK1    SET ENTRY NOT USED
179      STO  2  7      RESET

```

```

180 SK MDX 2 8 CONTINUE
181 MDX 3 -1 RESET
182 MDX LOOP1 LOOP
183 *
184 TEST2 LD L BVTAB BR TO B1 IF BOUND VFLE
185 BSC L B1, - TABLE IS EMPTY
186 *
187 * PURGE ENTRY FROM BVTAB
188 R4 LD I EDFS PURGE ASSOCIATED
189 BSI L PURG SUBSCRIPT EXPRESSION
190 MDX L EDFS,-1 TABLE ENTRY
191 MDX L BVTAB,-1 PURGE BOUND VAR TABLE ENTR
192 MDX B4 UNTIL DONE
193 MDX TEST2
194 *
195 * TEST FOR REFERENCED STATEMENT
196 TEST1 LD 1 1 STATEMENT NO
197 BSI L CHECK GET SYMBOL TABLE ID
198 AND L MASK2 CHECK FOR STMT FUNCTION
199 BSC L TLR,2 BR TO TLR IF ASF
200 LD 1 0 STATEMENT ID
201 SRA 1 CHECK IF
202 BSC E REFERENCED
203 MDX TLR1 YES
204 MDX TEST3 NO
205 TLR1 LD 3 0 REFERENCED
206 OR L MASK2 SET SYMBOL TBL ID
207 STO 3 0 TO INDICATE REFERENCED
208 MDX TEST3 GO RESET PRIORITY STATUS
209 *
210 * CLEAR SUBSCR EXPR TABLE
211 *
212 CLSET DC 0 RETURN LINK
213 LDX L2 SETBL POINT TO SUBSCR EXPR TABLE
214 LDX I3 C15 NUMBER OF ENTRIES
215 SLA 16 ZERO
216 LOOP STO 2 7 LOOP
217 MDX 2 8 TO
218 MDX 3 -1 CLEAR
219 MDX LOOP SEIBL
220 BSC I CLSET EXII
221 *
222 * NORMALIZE SWITCHES
223 TLR BSI CLSET GO CLEAR SUBSCR EXPR TABLE
224 B1 SLA 16 ZERO
225 STO L SW1 FP NON-D INTEGER SW
226 STO L SW2 I/O FOUND SW
227 STO L SW3 FP NON-D INTEGER VBLE SW
228 STO L SW4 IN FUNCTION CALL SW
229 STO L SW5 DO NOT OPTIMIZE SW
230 *
231 * BRANCH
232 B LD 1 0 TO
233 SRA 1
234 AND IOTPE TEST4
235 BSC L TEST4, - IF ARITHMETIC STMT
236 S CALLC TEST5
237 BSC L TEST5, - IF CALL STMT
238 S READC TEST8
239 BSC L TEST8, - IF READ STMT

```

```

240      S      WRITC      TEST7
241     BSC    L      TEST7, -   IF WRITE STMT
242      S      FINDC      TEST8
243     BSC    L      TEST8, -   IF FIND STAMENT
244      S      IFC        C
245     BSC    L      C, -       IF IF STATEMENT
246      S      DOC        IF NOT ANY OF ABOVE OR DO,
247     BSC    L      MOVE,Z     THEN MOVE TO NEXT STMT
248 *
249     BSI          CLSET      DO CLEARS SETBL
250 *
251  MOVE  LD      1  0      MOVE TO NEXT STATEMENT
252          SRA      2      GET
253          AND     L  IDNRM      NURM
254          STO      NXID 1     AND
255  NXID  MDX     L1  **      MOVE
256          MDX      TEST      GO TEST NEW STMT TYPE
257 *
258 *          CONSTANTS
259  SW1   DC      0          FP NON-D INTEGER SWITCH
260  SW2   DC      0          I/O FOUND SW
261  SW3   DC      0          PUT FP NON-D INTEGER VBLIS
262  SW4   DC      0          IN FUNCTION CALL SW
263  SW5   DC      0          DO NOT OPTIMIZE SW
264  IDTPE DC      /7C00     FOR EXTRACTING STRING IDTY
265  ENDC  DC      /0800     END TYPE
266  FORMM DC      /3000- /0800  FORMAT TYPE
267  DFL   DC      /7800- /3000  DEFINE FILE
268  CLE   DC      /7000- /7800  CALL LINK/EXIT
269  DATAS DC      /7C00- /7000  DATA STATEMENT
270  MASK1 DC      /8000     ENTRY NOT USED BITS
271  MASK2 DC      /0020     SYM IBL OR FOR REF STMT
272  CALLC DC      /1800     CALL TYPE
273  READC DC      /4800- /1800  READ TYPE
274  WRITC DC      /4400- /4800  WRITE
275  FINDC DC      /7400- /4400  FIND
276  IFC   DC      /3C00- /7400  IF
277  DOC   DC      /2C00- /3C00  DO
278 *
279 *          *****
280 *          * SUBSCRIPT EXPRESSION TABLE
281 *          *****
282 *          * 15 ENTRIES - 8 WORDS EACH
283 *          * WORD1  D4
284 *          *      2  I
285 *          *      3  D1
286 *          *      4  J
287  SETBL BSS      120 *      5  D2
288 *          *      6  K
289 *          *      7  D3
290 *          *      8  /8010 ENTRY NOT USED
291 *          *          /0010 USED THIS STMT
292 *          *          /0000 USED PREV STMT
293 *          *
294 *          *****
295 *
296  C15   DC      15          FIFTEEN
297  IDNRM DC      /01FF
298  IDSAV DC      0          ID SAVE
299  DOAC  DC      /002A      IMPLIED DO OPERATOR

```

```

300 DIOOP DC /0030 DISK I/O OPERATOR
301 DEFIL DC /7800-/0800 DF TEST CONSTANT
302 *
303 *
304 TEST4 MDX L SW3,1 ARITH STMT
305 MDX L SW1,1 SET TO PURGE FLPT ND IVA
306 MDX C TO PUT INTO BVTAB
307 * READ/FIND STMT
308 TEST8 MDX L SW1,1 SET SWITCHES 1,2
309 * WRITE STMT
310 TEST7 MDX L SW2,1 SET SWITCH 2
311 MDX C SW2 - I/O FOUND
312 * CALL STMT
313 TEST5 MDX L SW3,1 SET SWITCH 3
314 *
315 * INITIALIZE TO SCAN BODY
316 C STX 1 IDSAV SAVE STRING POINTER
317 LD 1 0 STRING POINTER
318 BSC E SKIP
319 MDX 1 1 OVER
320 MDX 1 1 STATEMENT NUMBER
321 SLA 16 NORMALIZE
322 STO L TOT TOT
323 *
324 LD 1 0
325 S DIOOP BRANCH TO AVAR
326 BSC L AVAR, - IF DISK I/O OPERATOR
327 *
328 F LD 1 0 BRANCH TO JAY
329 BSC L JAY, - IF SEMICOLON
330 *
331 LD L SW2 BRANCH TO HEH
332 BSC L HEH, - IF I/O FOUND
333 *
334 LD 1 0 BRANCH TO I
335 S DOAC IF
336 BSC L I,Z NOT IMPLIED DO
337 *
338 LD 1 -1 PURGE INDICY FROM
339 BSI L PURG SUBSCRIPT EXPRESSION TBL
340 *
341 EE SLA 16 NORMALIZE
342 STO L SW3 SW3
343 *
344 E MDX 1 1 MOVE STRING POINTER
345 MDX F RECHECK
346 *
347 * PUT ASSUC VBLE IN BVTAB
348 AVAR STX 1 AVARX 1 SAVE XR1
349 LDX I1 SOFS INITIALIZE TO SCAN STRING
350 *
351 AVAR2 LD 1 0 LOAD STMT ID
352 SRA 1
353 AND L IDTPE MASK FOR TYPE CHECK
354 *
355 S L ENDC BRANCH TO AVARX
356 BSC L AVARX, - IF END
357 *
358 S DEFIL BRANCH TO AVAR4
359 BSC L AVAR4,Z IF NOT DEFINE FILE

```

```

360 *
361 LD 1 4 GET ASSOCIATED VARIABLE
362 BSI PBVTB PUT INTO BOUND VBLE TABLE
363 *
364 AVAR4 LD 1 0 LOAD STATEMENT ID
365 SRA 2 GET
366 AND IDNRM STATEMENT NORM
367 STO AVAR6 1 AND READY FOR
368 AVAR6 MDX L1 *-- MOVE TO NEXT STATEMENT
369 MDX AVAR2 GO CHECK NEXT STATEMENT
370 *
371 AVARX LDX L1 *-- RESTORE XR1
372 MDX F CONTINUE
373 *
374 *
375 MASK3 DC /DBDE CONSTANTS
376 MASK4 DC /4000
377 *
378 I LD L SW3 BR TO B3
379 BSC L B3,Z IF SW3 SET
380 LD L SW1 BR TO D
381 BSC L D,- IF SW1 NOT SET
382 *
383 B3 LD 1 0 GET ID WORD
384 BSI L CHECK OF SYMBOL TABLE ENTRY
385 AND MASK3 CHECK
386 EOR MASK4 IF DIMENSIONED
387 BSC L D,Z BR IF DIMENSIONED
388 LD L SW3 BR TO PURGE SUBSCR EXPR TB
389 BSC L PGSET,- IF SW3 NOT SET
390 BSI PBVTB PUT NAME IN BOUND VBLE TBL
391 MDX E GO MOVE POINTER
392 *
393 PGSET LD 1 0 PURGE NAME FROM
394 BSI L PURG SUBSCRIPT EXPR TABLE
395 MDX E GO MOVE POINTER
396 *
397 PBVTB DC *-- LINK
398 ECHO LD 1 0 PUT SUBSCRIPT
399 MDX L EOF5,1 VARIABLE IN
400 STO I EOF5 BOUND VAR TABLE
401 MDX L BYTAB,1 INCREMENT BYTAB SIZE INDCT
402 LD L EOFST CHECK
403 A L TWO FOR
404 S L EOF5 OVERLAP
405 BSC L WAR, Z INDICATE OVERLAP AND GO ON
406 BSC I PBVTB RETURN
407 *
408 SUBZ DC /0018 LITERAL SUBSCRIPT INDICATO
409 SUB1 DC /0002 ONE-DIM SUBSCRIPT INDICATO
410 C7 DC 7 Y SWITCH VALUE
411 C11 DC 11 Y SWITCH VALUE
412 *
413 * DETERMINE DIM LEVEL OF SUBSCR
414 D LD 1 0
415 S SUBZ BR TO TST
416 BSC L TST,- IF LITERAL SUBSCRIPT
417 S SUB1 BR TO TST1
418 BSC L TST1,- IF ONE-DIM SUBSCRIPT
419 S SUB1 BR TO TST2

```


| | | | | | |
|-----|-------|-----|----|---------|----------------------------|
| 480 | | MDX | | LQUP3 | CHECK NEXT |
| 481 | * | | | | |
| 482 | GOG1 | LDX | I1 | SAVEM | SUBSCR EXPR NOT IN TABLE |
| 483 | | MDX | | G | GO PUT IN |
| 484 | * | | | | |
| 485 | * | | | | |
| 486 | Y | DC | | 0 | SE RANGE COUNTER |
| 487 | SAVEM | DC | | 0 | SAVE STRING POINTER |
| 488 | SAVET | DC | | 0 | SAVE SETBL POINTER |
| 489 | TOT | DC | | 0 | CLOSE STRING NUMBER |
| 490 | SAVEB | DC | | 0 | SAVET Y |
| 491 | C16 | DC | | 16 | LENGTH SETBL 1 |
| 492 | * | | | | |
| 493 | GTTAB | BSS | | 15 | LOCAL SGT TABLE |
| 494 | * | | | | |
| 495 | GTEMP | DC | | /3000 | STRING ST BASE PTR |
| 496 | GTPID | DC | | /4024 | SYM TBL SGT ID |
| 497 | SW6 | DC | | 0 | CLOSE STRING SWITCH |
| 498 | * | | | | |
| 499 | * | | | | |
| 500 | TST5 | LDX | I1 | SAVEM | TAG VARIABLE NAME |
| 501 | | S | | SAVET | WITH SUBSCRIPT |
| 502 | | SLA | | 11 | EXPRESSION TABLE |
| 503 | | OR | 1 | -1 | ENTRY NUMBER |
| 504 | | STO | 1 | -1 | DO IT |
| 505 | * | | | | |
| 506 | | LD | 1 | 0 | CHECK FOR |
| 507 | | S | | SJNZ | LITERAL SUBSCRIPT |
| 508 | | BSC | L | TST5A,Z | BR IF NOT |
| 509 | TTA | LDX | L3 | ** | GET LAST STRING PTR |
| 510 | | MDX | 3 | -7 | AND MOVE BACK |
| 511 | | MDX | | JAM 1 | GO ADD EXPR TO TABLE |
| 512 | * | | | | |
| 513 | TST5A | STX | 1 | SAVET | SAVE STRING PTR |
| 514 | | LD | | SAVET | OFFSET |
| 515 | | A | | Y | BY |
| 516 | | STO | | SAVEB | Y |
| 517 | | LDX | I2 | SAVEB | POINT TO SE END |
| 518 | HOG | LD | 2 | 0 | MOVE |
| 519 | | STO | 1 | 0 | STATEMENT UP |
| 520 | | BSC | | - | |
| 521 | | MDX | | HOG 7 | DONE |
| 522 | | MDX | 1 | 1 | INCREMENT |
| 523 | | MDX | 2 | 1 | STATEMENT PTRS |
| 524 | | MDX | | HOG | AND CONTINUE |
| 525 | * | | | | |
| 526 | | LDX | I1 | SAVET | RESTORE STR PTR |
| 527 | | MDX | L | SW6,1 | SET TO CLOSE STRING |
| 528 | | LD | | TOT | |
| 529 | | A | | Y | TOT TOT Y |
| 530 | | STO | | TOT | |
| 531 | | BSC | L | F | GO CHECK FOR SEMICOLON |
| 532 | * | | | | |
| 533 | * | | | | CHECK FOR OPENING IN SETBL |
| 534 | G | LDX | L3 | SETBL | POINT TO SUBSCR EXPR TABLE |
| 535 | | LDX | 2 | 15 | TBL COUNT |
| 536 | | LD | 3 | 7 | ENTRY COUNT-1 |
| 537 | | BSC | | - | SKIP IF ENTRY NOT USED PRE |
| 538 | | MDX | | JAM | USED PREV - GO ADD |
| 539 | | MDX | 3 | 8 | NEXT ENTRY |

```

540      MDX      2  -1      DECREMENT CNTR
541      MDX      G  3      CONTINUE
542      LDX      L3 SETBL   NEW LOOP
543      LDX      2  15     INITIALIZE
544  GG      LD      3  7     ENTRY STATUS INDICATOR
545      BSC      BSC      SKIP IF ENTRY USED THIS ST
546      MDX      JAM      NO - GO ADD
547      MDX      3  8      NEXT
548      MDX      2  -1     EIC
549      MDX      GG      CONTINUE
550      BSC      L  PIG    ERROR - MORE THAN 15 SEXPRS
551  *
552  *          ADD EXPRESSION TO SE TABLE
553  JAM      STX      2  SAVET  SAVE XR2
554      LDX      L2 BUFF   POINT TO LOCAL SE
555      STX      1  SAVEB   SAVE XR1
556      LDX      1  7      COUNT
557      LD      2  0      LOOP
558      STO      3  0      TO
559      MDX      2  1      PUT
560      MDX      3  1      EXPRESSION
561      MDX      1  -1     IN SETBL
562      MDX      JAM 5     CONTINUE
563      LD      C16      /0010
564      STO      3  0      SET FOR USED THIS STMNT
565      LDX      I3 SAVE   ARITHMETIC
566      LDX      I1 SAVEB  TO
567      LD      C16      TAG VARIABLE NAME
568      S      SAVET      WITH
569      STO      SAVEB     ENTRY
570      SLA      11      NUMBER
571      OR      1  -1
572      STO      1  -1
573  *
574      LDX      L2 GTTAB-1 XR2 PTS TO SGT TABLE
575      MDX      I2 SAVEB   OFFSET PTR
576      LD      2  0      ENTRY
577      BSC      L  OPEN1, - GO GENERATE SGT IF EMPTY
578      STO      VINE      SAVE ENTRY TEMP
579      MDX      OPEN      GO OPEN STATEMENT
580  *
581  *          GENERATE GT
582  OPEN1 LDX      I3 EOFST  POINT TO END OF SYM TBL
583      STX      3  VINE    SAVE XR3
584      LD      L  SOFST   PERFORM
585      A      L  C3      A
586      S      VINE      LUT
587      SRT      16      OF
588      D      L  C3      OPERATIONS
589      STO      VINE      TO
590      LD      SAVEB     COMPUTE
591      SLA      11      ENTRY NUMBER
592      OR      GTEMP    OR BASE
593      OR      VINE
594      STO      2  0      STORE IN SGT TABLE
595      STO      VINE
596      LD      GTPID    SYMBOL TABLE SGT ID
597      STO      3  0      STORE
598      LD      L  SAVEB  SGT
599      STO      3  1      ENTRY

```

| | | | | | |
|-----|-------|-----|----|-------------|----------------------------|
| 600 | | SLA | | 16 | IN |
| 601 | | STO | 3 | 2 | SYMBOL TABLE |
| 602 | | MDX | L | EOFST,-3 | ADJUST END PTR |
| 603 | | MDX | L | SDFGT,-3 | ADJUST START OF GTS PTR |
| 604 | * | | | | |
| 605 | | LD | L | EOFST | CHECK |
| 606 | | A | | TWO | FOR |
| 607 | | S | L | EOFS | OVERLAP |
| 608 | | BSC | L | OPEN,- | GO OPEN IF NOT |
| 609 | * | | | | |
| 610 | WAR | MDX | L | ERROR,1 | SET OVERLAP ERROR INDICATO |
| 611 | | BSC | L | OUT | EXIT TO NEXT PHASE |
| 612 | * | | | | |
| 613 | * | | | | OPEN STATEMENT TWO WORDS |
| 614 | OPEN | STX | 1 | ZEMP | CALCULATE |
| 615 | | LD | L | EOFS | DISTANCE |
| 616 | | S | | ZEMP | TO |
| 617 | | STO | | COVER 1 | END |
| 618 | COVER | LDX | L1 | *--* | OF |
| 619 | | MDX | 1 | 1 | STRING |
| 620 | | LDX | 12 | EOFS | EXTEND STRING |
| 621 | | MDX | L | EOFS,2 | TWO WORDS |
| 622 | | LD | 2 | 0 | LOOP |
| 623 | | STO | 2 | 2 | TO |
| 624 | | MDX | 2 | -1 | MOVE STRING UP TWO WORDS |
| 625 | | MDX | 1 | -1 | FROM STATEMENT PTR |
| 626 | | MDX | | COVER 7 | CONTINUE |
| 627 | * | | | | |
| 628 | | LDX | I1 | ZEMP | RESTORE STATEMENT POINTER |
| 629 | | LD | I | IDSAV | ADJUST |
| 630 | | A | | EIGHT | STATEMENT NORM |
| 631 | | STO | I | IDSAV | BY EIGHT |
| 632 | * | | | | |
| 633 | | LD | L | EOFST | CHECK |
| 634 | | A | | TWO | FOR |
| 635 | | S | L | EOFS | OVERLAP |
| 636 | | BSC | L | WAR, Z | BR TO WAR IF OVERLAP |
| 637 | * | | | | |
| 638 | | LD | | VINE | GET SGT |
| 639 | | STO | 1 | 1 | PUT ON STRING |
| 640 | | LD | | COMMC | GET COMMA |
| 641 | | STO | 1 | 2 | PUT ON STRING |
| 642 | * | | | | |
| 643 | | MDX | L | Y,2 | MOVE STRING POINTER |
| 644 | | MDX | I1 | Y | BY Y 2 |
| 645 | | BSC | L | F | GO CHECK FOR SEMICOLON |
| 646 | * | | | | |
| 647 | * | | | | CONSTANTS |
| 648 | VINE | DC | | 0 | TEMP SGT ON HAND |
| 649 | ZEMP | DC | | 0 | SAVE STRING POINTER |
| 650 | COMMC | DC | | /0012 | COMMA OPERATOR CODE |
| 651 | TWO | DC | | 2 | TWO |
| 652 | EIGHT | DC | | 8 | EIGHT - TO READ JUST NORM |
| 653 | CALOP | DC | | /002E | CALL OPERATOR |
| 654 | EQUAL | DC | | /000E-/002E | EQUAL OPERATOR |
| 655 | * | | | | |
| 656 | * | | | | |
| 657 | HEH | LD | 1 | 0 | TEST FOR |
| 658 | | S | | CALOP | CALL OPERATOR |
| 659 | | BSC | L | TOR,Z | GO TEST FOR EQUAL IF NOT |

```

660          MDX  L  SW4,1      SET IN FUNCTION CALL SWITC
661          LD   1 2          IS NEXT OP A SEMICOLON
662          BSC  L  A11, -    BR IF YES
663          *
664          JOH  MDX  L  SW5,1      NO-SET DO NOT OPTIMIZE
665          BSC  L  E          GO MOVE STRING POINTER
666          *
667          TOR  S          EQUAL    GO TEST FOR IN FUNC CALL
668          BSC  L  TOR1,Z      IF NOT EQUAL OP
669          STO  L  SW1        NORMALIZE SW1
670          STO  L  SW3        NORMALIZE SW3
671          MDX          JOH 2      GO MOVE POINTER
672          *
673          TOR1 LD  L  SW4        GO TEST SW3 IF NOT
674          BSC  L  I, -      IN FUNCTION CALL
675          LD   1 0          STRING ELEMENT
676          S          LPARN      IS IT LEFT PARENTHESIS
677          BSC  L  TOR2,Z      BR IF NOT
678          MDX  L  COUNT,1     INCREMENT COUNT
679          MDX          JOH 2     GO MOVE COUNTER
680          *
681          *
682          LPARN DC /0010      CONSTANTS
683          RPARN DC /0002-/0010 LEFT PAREN OP
684          COUNT DC 0          RIGHT PAREN OP
685          MASKR DC /2000      PARENTHESIS MATCHER
686          SIMP  DC 0          COMMON EXTRACTOR
687          SIMP2 DC 0          TEMP FOR JAY ROUTINE
688          *
689          *
690          TOR2 S          RPARN   GO TEST SW3 IF NOT
691          BSC  L  I,Z        RIGHT PAREN
692          MDX  L  COUNT,-1    DECREMENT PAREN MATCHER
693          MDX          JOH 2
694          *
695          LD   L  SW3        BR IF
696          BSC  L  DIE, -     SW3 NORMALIZED
697          *
698          *
699          A11  STX  1  OUTZ 1    PURGE COMMON VBLES FROM SET3L
700          LDX  L1 SETBL      POINT TO SUBSCR EXPP TABLE
701          LD   L  C15        SET COUNT
702          STO  COUNT        FOR NO OF ENTRIES
703          A1Z  LDX  2 6      COUNT OVER VBLES
704          LD   1 0          D-FACTOR
705          BSC  L  A2Z, -    BR IF ZERO
706          LD   1 1          I,J,K
707          BSI  L  CHECK      GO COMPUTE ST ADDR, GET I
708          AND  MASKR        IS IT COMMON VARIABLE
709          BSC  L  A2Z,Z      YES-BRANCH
710          MDX  1 2          NO-MOVE ENTRY PTR
711          MDX  2 -2         DECREMENT FOR NEXT VAR
712          MDX          A1Z 1  CONTINUE CK
713          A3Z  MDX  1 2          MOVE PTR TO END OF ENTRY
714          MDX  L  COUNT,-1   DECREMENT NO ENTS COUNT
715          MDX          A1Z     GO CHECK NEXT ENTRY
716          MDX          OUTZ    DONE
717          A2Z  STX  2 A2Z 2    ADJUST
718          MDX  L1 **        SETBL POINTER
719          SLA          16      ZERU

```

| | | | | | |
|-----|-------|-----|----|---------|----------------------------|
| 720 | | STO | 1 | 1 | CLEAR COMMON VBLE |
| 721 | | MDX | | A3Z | CONTINUE |
| 722 | * | | | | |
| 723 | OUTZ | LDX | L1 | *** | RESTORE STRING POINTER |
| 724 | | BSI | L | ANIML | PURGE BND VBL FROM SETBL |
| 725 | * | | | | |
| 726 | DIE | SLA | | 16 | NORMALIZE |
| 727 | | STO | L | SW4 | SW4 AND |
| 728 | | STO | L | SW5 | SW5 |
| 729 | | BSC | L | F | GO MOVE POINTER |
| 730 | * | | | | |
| 731 | * | | | | |
| 732 | JAY | BSI | L | ANIML | PURGE BND VBL FROM SETBL |
| 733 | | STX | 1 | HEMP 1 | SAVE POINTER |
| 734 | | LDX | I1 | IDSAV | GET INITIAL STMT PTR |
| 735 | | LD | 1 | 0 | GET STATEMENT ID |
| 736 | | SRA | | 1 | TEST FOR REFERENCED STMT |
| 737 | | BSC | | E | SKIP IF NOT |
| 738 | | MDX | | JAY2 | REFERENCED-GO CLEAR SETBL |
| 739 | HEMP | LDX | L1 | *** | PICK UP CURRENT STRING PTR |
| 740 | | LD | L | SW6 | BRANCH TO JAY1 |
| 741 | | BSC | L | JAY1,Z | IF STRING NEEDS CLOSING |
| 742 | | MDX | 1 | 1 | OTHERWISE INCR TO NEXT STM |
| 743 | | MDX | | XYZ | GO TO TEST NEXT |
| 744 | * | | | | |
| 745 | * | | | | CLOSE STRING BY TOT WORDS |
| 746 | JAY1 | MDX | 1 | 1 | XR1 NOW PTS TO NEXT STMT |
| 747 | | STX | 3 | SIMP1 1 | SAVE CURRENT XR3 |
| 748 | | STX | 1 | SIMP | SAVE CURRENT XR1 |
| 749 | | LD | | SIMP | INITIALIZE |
| 750 | | A | L | TOT | FOR CLOSE |
| 751 | | STO | | SIMP2 | XR1 CLOSE POINT |
| 752 | | LDX | I2 | SIMP2 | XR2 FROM POINT |
| 753 | | LD | L | EQFS | XR3 WORD COUNT |
| 754 | | S | | SIMP2 | . |
| 755 | | STO | | SIMP2 | . |
| 756 | | LDX | I3 | SIMP2 | . |
| 757 | | MDX | 3 | 1 | . |
| 758 | LAME | LD | 2 | 0 | LOOP |
| 759 | | STO | 1 | 0 | TO |
| 760 | | MDX | 1 | 1 | CLOSE |
| 761 | | MDX | 2 | 1 | STRING |
| 762 | | MDX | 3 | -1 | |
| 763 | | MDX | | LAME | CONTINUE |
| 764 | | MDX | 1 | -1 | RESET |
| 765 | | STX | L1 | EQFS | END OF STRING POINTER |
| 766 | | LDX | I1 | SIMP | RESTORE XR1 POINTING NEXT |
| 767 | * | | | | |
| 768 | SIMP1 | LDX | L3 | *** | RESTORE XR3 |
| 769 | | LD | I | IDSAV | GET STATEMENT ID |
| 770 | | SRT | | 2 | |
| 771 | | S | L | TOT | ADJUST NORM |
| 772 | | SLT | | 2 | |
| 773 | | STO | I | IDSAV | RESET |
| 774 | XYZ | BSC | L | TEST | GO TEST |
| 775 | * | | | | |
| 776 | * | | | | READ AND GO TO NEXT PHASE |
| 777 | OUT | BSI | L | POLRX | CALL DOWN PHASE 16 |
| 778 | | DC | | 16 | NEXT PHASE NUMBER |
| 779 | * | | | | |


```

840      OR      L  MASK2      INDICATE REFERENCED STMTN
841      STO      3  0          STORE IN SYMBOL TABLE
842      LD       1  0          RESET
843      AND      MC3          STATEMENTJ HEMP      GO BACK
844  *
845  *
846  *
847  PURG  DC      ***          SUBROUTINE TO PURGE INDICES
848      STO      TEMP          FROM SUBSCRIPT EXPR TABLE
849      LD       L  C15          LINK
850      STO      CNT          SAVE PURGE ELEMENT
851      LDX     L2 SETBL        SET CNT EQUAL TO NUMBER
852      STX     3  SAVE          OF SETBL ENTRIES
853      STX     2  TAP 4        POINT TO SETBL
854  PURR  LD       2  7          SAVE XR3
855      BSC     L  PURR 8,Z      SAVE INDEX PTR FOR MODIF
856      MDX     2  8            ENTRY INDICATOR
857      MDX     L  CNT,-1        BR IF NOT USED PREV STMT
858      MDX     PURR-1          NEXT ENTRY INDICATOR
859      MDX     OUT1           DECREMENT ENTRY COUNT
860      LDX     I3  C3          LOOP
861  LOOP4 LD       2  1          N/A
862      S       TEMP          COUNT FOR 3 INDICES
863      BSC     L  TAP, -        K,J,K
864      MDX     2  2            DOES INDICY COMPARE
865      MDX     3  -1           YES, GO PURGE
866      MDX     LOOP4          NO, CONTINUE-PT TO NEXT
867      MDX     2  2            DECREMENT INDEX COUNT
868      STX     2  TAP 4        LOOP
869      MDX     L  CNT,-1        POINT XR2 TO NEXT ENTRY
870      MDX     PURR           SAVE INDEX PTR FOR MODIF
871      MDX     OUT1           DECREMENT ENTRY COUNT
872  TAP   MDX     L  TAP 4,7     LOOP FOR NEXT ENTRY
873      SLA     15            N/A
874      STO     L  ***          MOVE TO ENTRY INDICATOR
875      MDX     2  2            CLEAR
876      MDX     3  -1           ENTRY INDICATOR
877      MDX     TAP 5          LOOP TO
878      MDX     LOOP4 7        SET UP
879  OUT1  LDX     I3  SAVE          XR2
880      BSC     I  PURG         AND GO SET FOR NEXT ENT CK
881  *
882  *
883  SAVE  DC      0            RESTORE XR3
884  CNT   DC      0            RETURN
885  C3    DC      3            XR3 SAVE
886  MC3   DC      /FFFFD      SETBL ENTRY COUNT
887  TEMP  DC      0            THREE
888  BVTAB DC      0            HOLD PURGE ELEMENT
889  *
890  BUFF  BSS     7            BOUND VBL TBL INDICATOR
891  *
892  *
893  *
894  *
895  *
896  CHECK DC      0            SUBSCR EXPRESSION AT HAND
897      BSC     L  SIM 3,-      SUBROUTINE TO COMPUTE SYMBOL
898      AND     HAB           TABLE ADDRESS AND TO LOAD
899      M       MC3          ACC WITH SYM TBL ID WORD

```

```

900      SLT      16      ;
901      A        C3      ;
902      A        L      SJFST ;
903      STO      SIM 1   ;
904  SIM  LDX    L3  ***   XR3 PTS TO ADDRESS
905      LD      3  0      GET SYM TBL ID WORD
906      BSC    I  CHECK   RETURN
907      *
908  MAB  DC      /07FF
909      *
910      *              SUBROUTINE TO PURGE BOUND V3L
911      *              TABLE FROM THE SUBSC EXPR TB
912  ANIML DC      0      LINK
913      LD      L  RYTAB  TABLE COUNTER
914      BSC    L  DIE1, -  GO OUT IF EMPTY
915      LD      I  EQFS   LD ELEMENT
916      BSI    L  PJRG    PURGE IT
917      MDX    L  EQFS,-1  DECREMENT TABLE
918      MDX    L  BVTAB,-1 DECREMENT TABLE SIZE
919      MDX    ANIML 5    LOOP
920  DIE1 BSC    I  ANIML  RETURN
921      *
922      BSS      OVERL-***320*4  PHASE-15 PATCH AREA
923      *
924      *
925      END      START

```