```
1  *    GENERAL AUTOMATION, INC,  ALL RIGHTS RESERVED
2  ************************************************
3  *
4  *    PROGRAM NAME  FPH-06
5  *
6  *    MODEL NUMBER  8F006
7  *
8  *    PURPOSE       FORTRAN PHASE-06
9  *
10 *    PROGRAMMER    DICK WALLMANN, MODS-MARK ELFIELD
11 *
12 *****************   REVISION LIST   **************
13 *
14 *    RV DATE     SCO   BY  REASON FOR CHANGE
15 *    -- -------- ----- --- ----------------------
16 *
17 *    01 11/16/70 NONE  INITIAL RELEASE
18 *
19 ************************************************
20 ************************************************
21 * GA 18/30  FORTRAN COMPILER   05/01/70
22 *STATUS-VERSION 1, MODIFICATION 0
23 *FUNCTION/OPERATION-
24 *    * SCANS ALL IF, CALL, AND ARITHMETIC STMNTS
25 *      FOR VALID REAL CONSTANTS,
26 *    * CONVERTS REAL CONSTANTS TO STANDARD OR
27 *      EXTENDED PRECISION FORMAT, AS SPECIFIED BY
28 *      THE FORTRAN COMMUNICATIONS AREA INDICATORS
29 *      DERIVED FROM THE CONTROL RECORDS IN PHASE 1,
30 *    * CHECKS SYMBOL TABLE FOR PREVIOUS ENTRY OF
31 *      CONSTANT, IF FOUND, NO NEW ENTRY IS MADE,
32 *      THE SYMBOL TABLE ADDRESS OF THE CONSTANT AND
33 *      THE CONSTANT OPERATOR REPLACE THE CONSTANT
34 *      IN THE STATEMENT STRING, IF NOT FOUND, THE
35 *      CONVERTED CONSTANT IS ADDED TO THE SYMBOL
36 *      TABLE,  THE CONSTANT OPERATOR FOLLOWED BY
37 *      THE SYMBOL TABLE ADDRESS REPLACES THE
38 *      CONSTANT IN THE STATEMENT STRING, THE STRING
39 *      IS CLOSED UP AFTER ALTERATION,
40 *ENTRY POINTS-
41 *    * START-PHASE 6 IS READ INTO CORE BY PHASE 5
42 *            VIA THE ROLRX ROUTINE AND EXECUTION
43 *            BEGUN AT LOCATION START
44 *INPUT-
45 *    * THE STATEMENT STRING FROM PREVIOUS PHASES
46 *    * THE SYMBOL TABLE FROM PREVIOUS PHASES
47 *    * FCOM
48 *OUTPUT-
49 *    * THE MODIFIED STATEMENT STRING
50 *    * THE MODIFIED SYMBOL TABLE
51 *    * FCOM
52 *EXTERNAL REFERENCES-
53 *    SUBROUTINES-
54 *      * ROLRX
55 *EXITS-
56 *    NORMAL-
57 *      * EXITS VIA A CALL TO THE ROLRX ROUTINE
58 *        WHICH WILL READ DOWN THE FOLLOWING PHASE
59 *    ERROR-
```

```
60  *       * AN OVERLAP ERROR CAUSES THE PHASE TO EXIT
61  *         AFTER SETTING A WORD IN FCOM.  PROCESSING
62  *         OF THE STRING IS TERMINATED.
63  *       * ERRORS DETECTED ARE NUMBERS 23,50.
64  *TABLES-WORK AREAS
65  *       * THE STATEMENT STRING, THE SYMBOL TABLE
66  *         AND THE FCOM AREA.
67  *ATTRIBUTES- NONE.
68  *NOTES- THE SWITCHES USED IN PHASE 6 FOLLOW.  IF
69  *    NON-ZERO, THE SWITCH IS TRANSFER=T.  IF ZERO,
70  *    THE SWITCH IS NORMAL=N.
71  *       SW1 - DECIMAL POINT
72  *           N= DECIMAL POINT ALLOWABLE
73  *       SW2 - E IN EXPONENT
74  *           N= E IN EXPONENT ALLOWABLE
75  *       SW3 - SIGN
76  *           T= SIGN ALLOWABLE
77  *       SW4 - SIGN
78  *           T= SIGN WAS MINUS
79  *       SW5 - NON-ZERO CHARACTER
80  *           T= NON-ZERO CHARACTER ENCOUNTERED
81  *       SW6 - STRING CLOSURE
82  *           T= STRING CLOSURE REQUIRED
83  *       SW12- CHARACTER BUFFER
84  *           T= BUFFER EMPTY
85  *       SW14- FIRST OF TWO WORDS
86  *           N= WORKING ON FIRST OF TWO WORDS
87  *       SW16- FIRST CONSTANT
88  *           N= FIRST CONSTANT OF STATEMENT
89          ABS     REAL CORE
90  *       SYSTEM AND FORTRAN EQUATES
91  MEMRY EQU     ~~32FFF~~ CORE MAXIMUM CORE SIZE
92  PHSIZ EQU     4*320       MAXIMUM PHASE SIZE
93  OVERL EQU     MEMRY-PHSIZ     PHASES 2-29 START
94  FCOM  EQU     OVERL-22    FORTRAN COMM. TABLE
95  PHNTB EQU     FCOM-56     PHASE TABLE
96  ROLRX EQU     PHNTB-50    INTERPHASE CALL
97  *       FORTRAN COMMUNICATION AREA
98  *                       FORTRAN COMMUNICATION AREA
99          ORG     FCOM
100 SOFS  BSS     1           START OF STRING
101 EOFS  BSS     1           END OF STRING
102 SOFST BSS     1           START OF SYMBOL TABLE
103 SOFNS BSS     1           START OF NON-STMNT NUMBERS
104 SOFXT BSS     1           START OF SUBSC TEMPS
105 SOFGT BSS     1           START OF GENERATED TEMPS
106 EOFST BSS     1           END OF SYMBOL TABLE
107 COMON BSS     1           NEXT AVAILABLE COMMON
108 CSIZE BSS     1           SIZE OF COMMON
109 ERROR BSS     1           OVERLAP ERROR
110 FNAME BSS     1           PROGRAM NAME
111         BSS     1           *
112 SORF  BSS     1           SUBR (-) OR FUNCTION (+)
113 CCWD  BSS     1           CONTROL CARD WORD
114 *                   BIT 15 TRANSFER TRACE
115 *                   BIT 14 ARITHMETIC TRACE
116 *                   BIT 13 EXTENDED PRECISION
117 *                   BIT 12 LIST SYMBOL TABLE
118 *                   BIT 11 LIST SUBPROGRAM NAMES
119 *                   BIT 10 LIST SOURCE PROGRAM
```

```
120  *                BIT  9 ONE WORD INTEGERS
121  *                BIT  8 PUNCH
122  *                BIT  7 NONPROCESS PROGRAM
123  IOCS  BSS     1        IOCS CONTROL CARD WORD
124  *        SEE PHASE ONE FOR BIT PATTERNS
125  DFCNT BSS     1        NO. OF WDS IN FILE TABLE
126  LCOMN BSS     2        SIZE OF INSKEL COMMON
127  ICCER BSS     2        IOCS CONTROL CARD ERROR
128        BSS     2        SYSTEM LOADER USE
129  *                  END OF FORTRAN COMMUNICATION
130  *                  AREA
131        ORG     OVERL
132  START LD      ERROR    IF OVERLAP ERROR THEN BR T
133        BSC  L  OUT,Z    READ IN NEXT PH AND EXECUT
134  *               INITIALIZE PHASE
135  ORGIN LDX  I1 SOFS     START OF FIRST STMNT ADDR
136        STX   1 IDSAV
137  *               CHECK STATEMENT TYPE
138  CAP   LD    1 0        FIRST WD OF STMNT
139        SRA     1        POSITION TO TEST STMNT TYP
140        AND     IDTPE    STMNT TYPE (BITS 1-5)
141        S       ENDC     END STMNT CONSTANT
142        BSC  L  OUT,+-   BR TO RD NEXT PHASE
143        S       ARITC    ARITH STMNT CONSTANT
144        BSC     Z        SKIP NEXT IF ARITH STMNT
145        S       IFC      IF STMNT CONSTANT
146        BSC     Z        SKIP NEXT ON IF STMNT
147        S       CALLC    CALL STMNT CONSTANT
148        BSC     Z        SKIP NEXT IF CALL STMNT
149        S       READC    READ STMNT CONSTANT
150        BSC     Z        SKIP NEXT IF READ STMNT
151        S       WRITC    WRITE STMNT CONSTANT
152        BSC     Z        SKIP NEXT IF WRITE STMNT
153        S       FINDC    FIND STMNT CONSTANT
154        BSC  L  RCC,+-   BR IF FIND STMNT
155        S       DATAC    DATA STMNT CONSTANT
156        BSC  L  MOVE,Z   BR TO NEXT STMNT NOT DATA
157        STX  L0 SW7      SET DATA STMNT SW ON
158        MDX     RCCSS    BR TO INITIALIZE STMNT
159  *               MOVE TO NEXT STATEMENT
160  MOVE  LDX  I1 IDSAV    RESTORE START OF STMNT ADD
161        LD    1 0        STMNT ID WORD
162        SRA     2
163        AND     IDNRM    WORD COUNT MASK
164        STO     NXID+1   STMNT WORD COUNT
165  NXID  MDX  L1 0        INCR START ADDR BY WD CNT
166        MDX     ORGIN+2  BR TO CHECK STMNT TYPE
167  *               CONSTANTS AND WORK AREA
168  IDSAV DC        0      START OF STMNT ADDR TEMP
169  ARITC DC        0-/0800  ARITH STMNT CONSTANT
170  IFC   DC        /3C00    IF STMNT CONSTANT
171  CALLC DC        /1800-/3C00  CALL STMNT CONSTANT
172  READC DC        /4800-/1800  READ STMNT CONSTANT
173  WRITC DC        /4400-/4800  WRITE STMNT CONSTANT
174  FINDC DC        /7400-/4400  FIND STMNT CONSTANT
175  DATAC DC        /7C00-/7400  DATA STMNT CONSTANT
176  IDNRM DC        /01FF    STMNT WD COUNT MASK
177  DECP  DC        /000B    DECIMAL POINT CONSTANT
178  IDTPE DC        /7C00    STATEMENT TYPE MASK
179  ENDC  DC        /0800    END STMNT CONSTANT
```

```
180  SW6    DC       0          STRING CLOSURE SWITCH
181  T      DC       0          INDEX REG 1 TEMPORARY
182  SW16   DC       0          FIRST CONSTANT OF STMNT SW
183  *               INITIALIZE STATEMENT
184  RCC    SLA      16         CLEAR ACCUMULATOR
185         STO   L  SW7        RESET DATA STMNT SWITCH
186  RCCSS  STO      SW16       RESET FIRST CONSTANT SWITC
187         STO   L  CNT        ZERO STMNT WORD COUNT
188  *               TEST FOR CALL PDUMP
189  *               OUTPUT ERROR NUMBER 50 IF NOT A
190  *               SUBROUTINE OR FUNCTION, OR IF A
191  *               MAINLINE PROGRAM WITH NO CALLED
192  *               IOCS.
193         STX    1 *+1        SET XR3 TO POINT AT
194         LDX   L3 *-*        *STMNT ID,
195         LD     3 1          LOAD STMNT ID
196         BSC      E          SKIP IF NON-NUMBERED STMNT
197         MDX    3 1          INCR PT IF NUMBERED STMNT
198         SRA      1          DETERMINE IF STMNT IS CALL
199         AND      IDTPE      *
200         S        CALLP      *
201         BSC   L  ZNCPD,Z    BRANCH IF NOT A CALL STMNT
202         LD     3 1          LOAD NAME WORD 1
203         S        CPDN1      IS WD 1 PART OF ,PDUMP,
204         BSC   L  ZNCPD,Z    BRANCH IF NOT
205         LD     3 2          LOAD NAME WORD 2
206         S        CPDN2      IS WD 2 PART OF ,PDUMP,
207         BSC   L  ZNCPD,Z    BRANCH IF NOT
208         LD       SORF       TEST IF SUBROUTINE, IF
209         OR       IOCS       FUNCTION, OR IOCS
210         BSC   L  ZNCPD,Z    BRANCH IF YES
211         LDX    2 50         SET UP ERROR 50
212         BSC   L  REP        GO OUTPUT ERRORS
213  *               CONSTANTS
214  CALLP  DC       /1800      CALL TEST WORD
215  CPDN1  DC       /AE24      ,PDUMP, WORD 1
216  CPDN2  DC       /C517      ,PDUMP, WORD 2
217  ZNCPD  EQU      *          CONTINUE
218  *               MOVE POINTER
219         MDX    1 1          MOVE STMNT POINTER
220  Z3     LD     1 0          CHECK FOR OPERATOR
221         BSC      -          SKIP NEXT IF NOT OPERATOR
222         MDX      RC         BR IF OPERATOR
223  Z      MDX    1 1          MOVE STMNT POINTER
224         MDX      Z3         BR TO CHECK FOR OPERATOR
225  RC     S     L  SEMIC      SEMICOLON CONSTANT
226         BSC      Z          SKIP NEXT IF SEMICOLON
227         MDX      Z1         BR TO CHECK FOR DECIMAL PT
228         LD       SW6        STRING CLOSURE SWITCH
229         BSC   L  CLSUP,Z    BR TO CLOSE STRING
230         MDX    1 1          MOVE STMNT POINTER
231         MDX      ORGIN+2    BR TO CHECK STMNT TYPE
232  Z1     LD     1 0          CHECK FOR DECIMAL POINT
233         S        DECP       DECIMAL POINT CONSTANT
234         BSC   L  ZZ,+-      BR IF DECIMAL POINT
235         MDX      Z          BR TO MOVE POINTER
236  ZZ     LD       SW16       FIRST CONSTANT SWITCH
237         BSC   L  Z33,+-     BR IF FIRST CONSTANT
238         STX    1 T          CHECK FOR LEGAL CONSTANT
239         LD       T          CURRENT STMNT POINTER
```

```
240          S     L  TT          POINTER BEFORE COLLECT REA
241          BSC   L  JBR,+-      BR TO SET INVALID CON ERRO
242  Z33     MDX    1 -1          MOVE PT BACK TO OPERATOR
243          LD     1 0           IF SIGN POSITION IS ZERO
244          BSC      +           SKIP TO SAVE STMNT POINTER
245          MDX      *-4         BR TO MOVE BACK POINTER
246          STX   L1 TT          SAVE STMNT PT AT OPERATOR
247  *                INITIALIZE TO COLLECT REAL CONSTANT
248          SLA      16
249          LDX    2 5           COUNTER= 5
250  AMP     STO   L2 JANS-5      ZERO FIVE WORDS OF BUFFER
251          MDX    2 -1          DECR COUNTER
252          MDX      AMP         BR TO CONTINUE CLEARING
253          LDX    2 5           COUNTER= 5
254          STO   L2 SW5-5       ZERO SW1 TO SW5
255          MDX    2 -1          DECR COUNTER
256          MDX      *-4         BR TO CONTINUE CLEARING SW
257          STO   L  SW12        ZERO SWITCH 12
258          MDX    1 1           MOVE STMNT POINTER
259          STO      QCNT        ZERO CHAR COUNT
260          STO   L  SW14        ZERO FIRST OF 2 WDS SWITCH
261          STO   L  BIN         ZERO BIN
262  RC1     BSI   L  GET         BR TO GET NEXT ELEMENT
263          S     L  JCON1       LEAVES DIGIT ONLY
264          STO      X           SAVE CHARACTER
265          BSC   L  RC7,+-      BR IF ZERO
266          BSC   L  SW1T,+      BR NOT A DIGIT
267          S        JCON2       DECIMAL NINE
268          BSC   L  RC5,+       BR IF DIGIT
269  SW1T    LD       SW1         DECIMAL POINT ALLOWABLE SW
270          BSC   L  RC2,Z       BR IF DECML PT NOT ALLOWED
271          LD       X           CHARACTER
272          S        JCON3       DECIMAL POINT CONSTANT
273          BSC   L  RC8,+-      BR IF DECIMAL POINT
274  RC2     LD       SW2
275          BSC   L  RC3,Z       BR IF E NOT ALLOWABLE
276          LD       X           CHARACTER
277          S        JCON4       ALPHABETIC E CONSTANT
278          BSC   L  RC9,+-      BR IF ALPHA E
279  RC3     LD       SW3
280          BSC   L  RC10,+-     BR IF SIGN NOT ALLOWABLE
281          LD       X           CHARACTER
282          S        JCON5       PLUS SIGN CONSTANT
283          BSC   L  RC4,+-      BR IF PLUS SIGN
284          S        JCON6       MINUS SIGN CONSTANT
285          BSC   L  RC10,Z      BR IF SIGN NOT MINUS
286          STX    0 SW4         SET SW4 ON, SIGN WAS MINUS
287  RC4     STO      SW3         SET SW3= SIGN NOT ALLOWED
288          MDX      RC1         BR TO GET NEXT CHARACTER
289  *                CONSTANTS AND WORK AREA
290  X       DC       0           CHARACTER TEMPORARY
291  JCON1   DC       /0030       HIGH 2 BITS CONSTANT
292  JCON2   DC       9           DECIMAL NINE CONSTANT
293  JCON3   DC       /000B-/0030 DECIMAL POINT CONSTANT
294  JCON4   DC       /0005-/0030 ALPHA E CONSTANT
295  JCON5   DC       /000E-/0030 PLUS SIGN CONSTANT
296  JCON6   DC       /0020-/000E MINUS SIGN CONSTANT
297  JCON7   DC       100         DECIMAL 100 CONSTANT
298  JCON8   DC       192         DECIMAL 192 CONSTANT
299  JCON9   DC       1           DECIMAL   1 CONSTANT
```

```
300  LCON1  DC       5          DECIMAL   5 CONSTANT
301  TILL   DC       0          NO. OF EXPONENT DIGITS TEM
302  SW1    DC       0          DECIMAL POINT ALLOWABLE SW
303  SW2    DC       0          E IN EXPONENT ALLOWABLE SW
304  SW3    DC       0          SIGN ALLOWABLE SWITCH
305  SW4    DC       0          MINUS SIGN SWITCH
306  SW5    DC       0          NON-ZERO SWITCH
307  EXP    DC       0          EXPONENT TEMPORARY
308  JC011  DC       10         DECIMAL TEN CONSTANT
309  ERRNO  DC       0          ERROR NUMBER TEMPORARY
310  ERID   DC       /A008      ERROR ID CONSTANT
311  GCNT   DC       0          CONSTANT CHARACTER COUNT
312  RC5    LD       SW2
313         BSC   L  RC6,Z      BR E NOT ALLOWABLE
314         STX   0  SW5        SET SW5 NON-ZERO ENCOUNTER
315  RC5A   LD       X          MPY ANSWER BY 10 AND ADD X
316         BSI   L  MULT       BR TO MULTIPLY SUBROUTINE
317  RC5B   LD       SW1
318         BSC   L  RC1,+-     BR IF DECML PT ALLOWABLE
319         MDX   L  DEC,1      ADD 1 TO DEC
320         MDX      RC1        BR TO GET NEXT CHARACTER
321  RC6    LD       EXP        MPY EXPONENT BY 10, ADD X
322         M        JC011      TEN
323         SLT      16         SHIFT PRODUCT TO A REG
324         A        X          ADD X
325         STO      EXP        RESTORE EXPONENT
326         SLA      16
327         STO      SW3        RESET SIGN ALLOWABLE SWITC
328         LD       EXP        CHECK EXPONENT FOR LESS
329         S        JCON7      THAN 100.
330         BSC   L  JBR,-      BR IF 100 OR GREATER
331         MDX   L  TILL,1     ADD 1 TO TILL
332         MDX      RC1        BR TO GET NEXT CHARACTER
333  JBR    LDX   2  23         SET UP ERROR 23
334  REP    STX   2  ERRNO      SET ERROR NUMBER
335         LDX   I1 IDSAV      START OF STMNT ADDRESS
336         LD    1  0          STMNT ID WORD
337         SRA      2
338         AND   L  IDNRM      EXTRACT STMNT WORD COUNT
339         A     L  IDSAV      START OF NEXT STMNT ADDRES
340         STO      *+1
341         LDX   L2 0          NEXT STMNT ADDR
342         LD    L  EOFS       END OF STRING ADDRESS
343         S        *-4        NEXT STMNT ADDR
344         STO      *+1
345         LDX   L3 0          RANGE
346         MDX   3  1          NUMBER OF WDS TO MOVE
347         LD    1  0          STMNT ID WORD
348         AND      JCON9      TURN ON ERROR INDICATORS
349         OR       ERID
350         STO   1  0          RESTORE ID WORD
351         EOR      JCON9
352         BSC   L  *+3,E      BR IF ODD
353         A        LCON1
354         STO   1  0          STORE IN ID WORD
355         MDX   1  1          INCREMENT STMNT POINTER
356         LD       ERRNO      ERROR NUMBER
357         STO   1  1          REPLACE STMNT WITH ERROR N
358  RC6A   LD    2  0          NEXT WD TO MOVE
359         STO   1  2          CLOSE UP STRING
```

```
360          MDX    2 1          INCR NEXT WD TO MOVE ADDR
361          MDX    1 1          INCR ADDR TO STORE NEXT WD
362          MDX    3 -1         DECR WORDS TO MOVE COUNTER
363          MDX      RC6A       BR TO CONTINUE CLOSING
364          MDX    1 1          INCR ADDR TO STORE NEXT WD
365          STX   L1 EOFS       NEW END OF STRING ADDRESS
366          BSC   L  MOVE       BR TO EXAMINE NEXT STMNT
367  RC7     LD       SW2        E IN EXPONENT SWITCH
368          BSC   L  RC6,Z      BR IF E NOT ALLOWABLE
369          LD       SW5        NON-ZERO CHARACTER SWITCH
370          BSC   L  RC5A,Z     BR IF NON-ZERO TO MPY BY 1
371          MDX      RC5B       BR IF TEST SWITCH 1
372  RC8     MDX   L  SW1,1      TURN ON SWITCH 1
373          SLA      16         CLEAR ACCUMULATOR
374          STO      DEC        ZERO DIGIT COUNTER
375          BSC   L  RC1        BR TO GET NEXT CHARACTER
376  RC9     MDX   L  SW2,1      SET SW 2, E NOT ALLOWABLE
377          MDX   L  SW3,1      SET SW 3, SIGN ALLOWABLE
378          MDX   L  SW1,1      SET SW 1 = NO DECML POINT
379          SLA      16         CLEAR ACCUMULATOR
380          STO      TILL       ZERO TILL
381          STO      EXP        ZERO EXPONENT
382          BSC   L  RC1        BR TO GET NEXT CHARACTER
383  *                CHECK FOR VALID CONSTANT
384  RC10    LD     1 -1         STMNT ID WORD
385          BSC   L  JBR,+Z     BR IF NEG TO SET ERROR 23
386          LD       GCNT       CONSTANT CHARACTER COUNT
387          S     L  TWO        DECIMAL TWO
388          BSC   L  JBR,+      BR IF LESS THAN 3 TO ERROR
389          STX   L1 S          SAVE INDEX REGISTER 1
390          LD       SW5        NON-ZERO SWITCH
391          BSC   L  RC19,+-    BR TO CHECK PRECISION
392          LD       JCON8      MAXIMUM 4 WORD EXPONENT
393          STO   L  BIN        SET EXPONENT TO +64
394          LD       SW2        E IN EXPONENT ALLOWABLE SW
395          BSC   L  RC13,+-    BR IF E ALLOWABLE
396          LD       TILL       NO. OF EXPONENT CHARACTERS
397          S        JCON9      ONE
398          BSC   L  RC11,+-    BR IF ONE EXPONENT CHAR
399          S     L  JCON9      ONE
400          BSC   L  JBR,Z      BR MORE THAN 2 CHARS- ERRO
401  RC11    LD    L  SW4        EXPONENT SIGN SWITCH
402          BSC   L  RC12,+-    BR IF SIGN POSITIVE
403          LD       DEC        CALCULATE DECIMAL POINT
404          A     L  EXP        E EXPONENT VALUE
405          STO      DEC        NO. PLACES AFTER DECML PT
406          MDX      RC13       BR TO CHECK FOR NORMALIZE
407  RC12    LD       DEC        CALCULATE DECIMAL POINT
408          S     L  EXP        E EXPONENT VALUE
409          STO      DEC        NO. PLACES AFTER DECML PT
410  RC13    LD    L  JANS-4     CHECK FOR NORMALIZED RIGHT
411          BSC   L  RC17,Z     BR IF SHIFT RIGHT NEEDED
412          LD    L  JANS-3     CHECK FOR NORMALIZED LEFT
413          BSC   L  RC15,-     BR IF SHIFT LEFT NEEDED
414          LD       DEC        NO. PLACES AFTER DECML PT
415          BSC   L  RC14,+Z    BR TO MPY BY 10 IF NEGATIV
416          BSC   L  RC18,+     BR TO CHK EXPONENT IF ZERO
417          MDX   L  DEC,-1     DECR PLACES AFTER DECML PT
418          NOP
419  *                DIVIDE CONSTANT BY 10
```

```
420          LDX   2 -4          WORD LENGTH OF CONSTANT
421          SLT     32
422  JDIV    LD    L2 JANS+1     NEXT WORD OF CON TO DIVIDE
423          RTE     16          PREVIOUS REMAINDER INTO AC
424          LDX   3 1           INITLZ SIGN FLAG TO PLUS
425          S       JC010       DIVIDEND IN RANGE +4 TO -5
426          BSC     -           SKIP NEXT IF RANGE WAS OK
427          MDX   3 -2          SET SIGN FLAG NEG AND SKIP
428          A       JC010       ADD BACK 5 IF RANGE WAS OK
429          D     L JC011       DIVIDE BY TEN
430          MDX   3 -2          TEST SIGN FLAG,SKIP IF PLU
431          EOR     JC012       RESTORE SIGN BIT TO RESULT
432          STO   L2 JANS+1     REPLACE PARTIAL QUOTIENT
433          MDX   2 1           INCR DIVIDE COUNTER
434          MDX     JDIV        BR TO CONTINUE 4-WD DIVIDE
435          MDX     RC13        BR TO NORMALIZE CONSTANT
436  *               CONSTANT AND WORK AREA
437  JC010   DC      5           RANGE CHANGE CONSTANT
438  JC012   DC      /8000       SIGN CONSTANT
439  DEC     DC      0           DECIMAL POINT TEMPORARY
440  RC14    MDX   L DEC,1       INCR PLACES AFTER DECML PT
441  *               MULTIPLY CONSTANT BY 10
442          SLA     16
443          BSI   L MULT        BR TO MULTIPLY SUBROUTINE
444          MDX     RC13        BR TO NORMALIZE CONSTANT
445  RC15    BSC   L RC16,Z      BR TO NORMALIZE LEFT 1 BIT
446          LDX   2 -3          NORMALIZE LEFT ONE WORD
447          LD    L2 JANS+1     WORD TO MOVE LEFT
448          STO   L2 JANS       STORE ONE WORD LEFT
449          MDX   2 1           INCR MOVE COUNTER
450          MDX     RC15+3      BR TO MOVE NEXT WORD
451          SLA     16
452          STO   L JANS        ZERO LEAST SIGNIFICANT WOR
453          MDX   L BIN,-16     DECR EXPONENT FOR WD SHIFT
454          MDX     RC13        BR TO NORMALIZE CONSTANT I
455          MDX     RC13        SKIP OR IF NO SKIP.
456  *               NORMALIZE LEFT ONE BIT
457  RC16    LDX   2 4           NUMBER OF WORDS TO SHIFT
458          SLT     32          CLEAR ACC AND EXTENSION
459          LD    L2 JANS-4     NEXT LEAST SIGNIFICANT WOR
460          RTE     31          EFFECTIVE SHIFT ONE LEFT
461          STO   L2 JANS-4     RESTORE
462          SLT     15          ATTACH HIGH BIT TO NEXT WD
463          MDX   2 -1          DECR CNT AND SKIP IF ZERO
464          MDX     RC16+2      BR TO LOAD NEXT WD TO SHIF
465          MDX   L BIN,-1      DECR BINARY EXPONENT AND
466          MDX     RC13        BRANCH TO NORMALIZE IF SKI
467          MDX     RC13        OR NO SKIP.
468  *               NORMALIZE RIGHT ONE BIT
469  RC17    LDX   2 -5          MINUS NO. OF WDS TO SHIFT
470          SLT     32          CLEAR ACC AND EXTENSION
471          LD    L2 JANS+1     NEXT MOST SIGNIFICANT WORD
472          RTE     1           SHIFT ONE RIGHT
473          STO   L2 JANS+1     RESTORE
474          RTE     15          ATTACH LOW BIT TO NEXT WOR
475          MDX   2 1           INCI CNT AND SKIP IF ZERO
476          MDX     RC17+2      BR TO LOAD NEXT WD TO SHIF
477          MDX   L BIN,1       INCR BINARY EXPONENT AND
478          MDX     RC13        BRANCH TO NORMALIZE IF SKI
479          MDX     RC13        OR NO SKIP.
```

```
480  *                BINARY EXPONENT RANGE CHECK
481  RC18  LD         BIN        BINARY EXPONENT
482        BSC  L     JBR,+Z     BR ERR IF LESS THAN ZERO
483        S          JC014      DECIMAL 256
484        BSC  L     JBR,-      BR ERR GREATER THAN 255
485  *                CHECK IF EXTENDED PRECISION REQUIRED
486  RC19  LD   L     CCWD       CONTROL CARD WORD
487        SLA        13         EXTENDED INDR INTO SIGN PO
488        BSC  L     RC20,-     BR NOT EXTENDED PRECISION
489  *                COMBINE EXTENDED CONSTANT
490        LDD        JANS-3     TWO MOST SIGNIFICANT WORDS
491        SRT        1          VACATE SIGN POSITION
492        BSC        Z          MAKE SIGN PLUS
493        EOR        MASK
494        STD        JANS-3     RESTORE MANTISSA PORTION
495        LD         BIN        BINARY EXPONENT
496        OR         MASK       REAL CONSTANT INDICATOR
497        STO        JANS-1     STORE EXPONENT-ID WORD
498        MDX        RC21       BR TO CHECK DATA STMNT SW
499  *                COMBINE STANDARD CONSTANT
500  RC20  LDD        JANS-3     TWO MOST SIGNIFICANT WORDS
501        SRT        1          VACATE SIGN POSITION
502        BSC        Z          MAKE SIGN PLUS
503        EOR        MASK
504        RTE        16         LEAST SIGNIFICANT WD TO AC
505        AND        MASK1      CUT OFF LOWER EIGHT BITS
506        OR         BIN        EXPONENT INTO LOW 8 BITS
507        RTE        16         MOST SIGNIFICANT WD TO ACC
508        STD        JANS-3     COMPLETED STANDARD CONSTAN
509        LD         MASK       REAL CONSTANT INDICATOR
510        STO        JANS-1     SYMBOL TABLE ID WORD
511  *                CHECK FOR OPEN STRING
512  RC21  LD         SW7        DATA STMNT SWITCH
513        BSC  L     RC22,Z     BR TO OPEN STRING
514  *                CHECK FOR CONSTANT IN SYMBOL TABLE
515        LDX  I3    SOFNS      START OF NON-STMNT NUMBERS
516        MDX        HACK       BR TO CHECK END OF SEARCH
517  LOOP  LD    3 1             NEXT SYMBOL TABLE ENTRY
518        S          JANS-3     FIRST HALF OF CONSTANT
519        BSC        Z          SKIP NEXT IF EQUAL
520        MDX        EAT        BR NOT EQUAL - CONTINUE
521        LD    3 2             SECOND HALF OF TABLE ENTRY
522        S          JANS-2     SECOND HALF OF CONSTANT
523        BSC  L     EAT,Z      BR NOT EQUAL - CONTINUE
524        LD    3 0             SYMBOL TABLE ID WORD
525        S          JANS-1     CHK LOW 8 EQUAL (EXPONENT
526        SLA        8          IF EXTENDED PRECISION),
527        BSC  L     RC22,+-    BR IF CONSTANT FOUND
528  EAT   MDX   3 -3             DECR TABLE SEARCH ADDRESS
529  HACK  STX   3 SAVE           INDEX REGISTER 3 TEMPORARY
530        LD         SAVE       CURRENT SEARCH ADDRESS
531        S     L    EOFST      END OF SYMBOL TABLE ADDRES
532        BSC        Z          SKIP NEXT IF END OF SEARCH
533        MDX        LOOP       BR TO CONTINUE SEARCH
534  *                PUT CONSTANT IN SYMBOL TABLE
535        LD         JANS-1     ID WORD
536        STO   3 0              SYMBOL TABLE ID WORD
537        LD         JANS-2     LEAST SIGNIFICANT HALF CON
538        STO   3 2
539        LD         JANS-3     MOST SIGNIFICANT HALF CON
```

```
540           STO   3 1
541           MDX   L   EOFST,-3    DECR END OF STRING ADDRESS
542           MDX   L   SOFXT,-3    DECR START OF SUBSC TEMPS
543           MDX   L   SOFGT,-3    DECR START GENERATED TEMPS
544   *                 CHECK FOR SYMBOL TABLE OVERLAP
545           LD    L   EOFST       END OF SYMBOL TABLE ADDRES
546           A     L   TWO         TWO
547           S     L   EOFS        END OF STRING ADDRESS
548           BSC   L   RC22,-      BR IF NO ERROR
549           MDX   L   ERROR,1     SET OVERLAP ERROR ON
550   *                 GO TO NEXT PHASE
551   OUT     BSI   L   ROLRX       CALL DOWN NEXT PHASE
552           DC        07          NEXT PHASE NUMBER
553           BSS   E   0           MAKE ADDR BOUNDARY EVEN
554   NXTPH   DC        37          ID OF NEXT PHASE
555           BSS       3           LOADER TABLE FOR NEXT PHAS
556   *                 CONSTANTS AND WORK AREA
557   TT      DC        0           STATEMENT POINTER TEMPORAR
558           BSS   E   0           MAKE ADDR BOUNDARY EVEN
559   MASK    DC        /8000       SIGN BIT CONSTANT
560   MASK1   DC        /FF00       EXPONENT BIT MASK
561   JC014   DC        256         EXPONENT LIMIT CONSTANT
562   JC016   DC        3           SYMBOL TABLE MULTIPLE CON
563   BIN     DC        0           BINARY EXPONENT
564           DC        0           FIVE
565           DC        0           WORD
566           DC        0           REAL
567           DC        0           CONSTANT
568   JANS    DC        0           BUFFER
569   S       DC        0           STMNT POINTER TEMPORARY
570   SW7     DC        *-*         OPEN STRING SWITCH
571   SAVE    DC        0           INDEX REGISTER 3 TEMPORARY
572   *                 CHECK TO OPEN STRING
573   RC22    LD        TT          CURRENT STMNT POINTER
574           STO       RC22A+1
575           MDX   L   TT,4        INCR STMNT POINTER
576           LD        SW7         DATA STMNT SWITCH
577           BSC   L   RC22A,+-    BR IF NOT DATA STMNT
578           MDX   L   TT,1        ADJUST IT FOR 5 OR 6 WORDS
579           LD    L   CCWD        DEPENDING ON WHETHER
580           SLA       13          PRECISION IS EXTENDED OR
581           BSC   L   *+2,-       STANDARD,
582           MDX   L   TT,1        INCR ONE MORE FOR EXTENDED
583           LD        TT          CURRENT STRING POINTER
584           S         S           NEXT STMNT ADDR
585           BSC   L   OPEN,-Z     BR TO OPEN STRING IF PLUS
586   RC22A   LDX   L1  *-*         CURRENT STMNT POINTER
587           LD        SW7         DATA STMNT SWITCH
588           BSC   L   DATA,Z      BR IF DATA STMNT
589   *                 PUT CONSTANT AND OPERATOR ON STRING
590           LD        CONOP       OPERATOR
591           STO   1 1             STORE ON STRING
592           STX   3 SAVE          REGISTER 3 TO TEMPORARY
593           LD    L   SOFST       START OF SYMBOL TABLE ADDR
594           S         SAVE        CURRENT SYMBOL TBL ADDRESS
595           SRT       16          FORM NORMAL DIVIDEND
596           D     L   JC016       THREE
597           A         SIGN1       SYMBOL TABLE POINTER
598           STO   1 2             STORE ON STRING
599           MDX   1 3             MOVE STMNT POINTER
```

```
600   *                 CHECK IF STATEMENT CLOSURE REQUIRED
601   RC22B LD          TT          CURRENT STMNT POINTER
602         S           S           NEXT STMNT ADDRESS
603         BSC  L      REDOS,-     BR IF NO CLOSURE REQUIRED
604   *                 CLOSE UP STATEMENT
605         STO         NUMB        NUMBER OF WORDS TO CLOSE
606   NUMM  LDX  I2 S               ADDR FIRST WD AFTER GAP
607         LD    2 -1              NEXT WD TO MOVE BACK
608         S           SEMIC       SEMICOLON
609         BSC  L      OUTT,+-     BR TO END CLOSURE IF ZERO
610         LD    2 -1              MOVE WORD TO CLOSE STMNT
611         STO   1 0
612         MDX   1 1               INCR STMNT POINTER
613         MDX   2 1               INCR CLOSE POINTER
614         MDX         NUMM+2      BR TO CONTINUE CLOSING
615   OUTT  A           SEMIC       SEMICOLON
616         STO   1 0               MOVE LAST WORD
617         MDX  L      SW6,1       SET STRING CLOSURE SW ON
618         LD          CNT         ADJUST STMNT WORD COUNT
619         S           NUMB        BY THE AMOUNT OF CLOSURE
620         STO         CNT         AND RESTORE.
621   *                 INITIALIZE POINTER
622   REDOS LDX  I1 TT              CURRENT STMNT POINTER
623         MDX   1 -1
624         STX   1 TT              SAVE IN TEMPORARY
625         MDX  L      SW16,1      SET SW16 NOT FIRST CONSTAN
626         BSC  L      Z3          BR TO LOOK FOR OPERATOR
627   *                 OPEN STATEMENT
628   OPEN  LDX  I1 EOFS            END OF STRING ADDRESS
629         LDX  I2 EOFS
630         STO         *+1         NUMBER OF WORDS TO OPEN
631         MDX  L2     *-*         INCR BY NO. OF WDS TO OPEN
632   OPAMT EQU         *-1         LABEL FOR ADDR LAST COMMAN
633         STX  L2 EOFS            NEW END OF STRING ADDRESS
634         LD   L      EOFS
635         S           TT          CURRENT STMNT POINTER
636         STO         *+1         NUMBER OF WORDS TO MOVE
637         LDX  L3     *-*
638         MDX   3 2
639   OPENL LD    1 0               MOVE STRING TO OPEN
640         STO   2 0
641         MDX   1 -1              DECR MOVE FROM ADDR
642         MDX   2 -1              DECR MOVE TO ADDR
643         MDX   3 -1              DECR NO. OF WDS TO MOVE
644         MDX         OPENL       BR TO CONTINUE OPENING
645         LD   I      IDSAV       STMNT ID WORD
646         SRT         2
647         A           OPAMT       NO. WDS ADDED TO STMNT
648         SLT         2
649         STO  I      IDSAV       UPDATED STMNT ID WORD
650   *                 CHECK FOR OVERLAP ERROR
651         LD   L      EOFS        END OF STRING ADDRESS
652         S    L      EOFST       END OF SYMBOL TABLE
653         BSC  L      RC22A,+     BR IF NO OVERLAP ERROR
654         MDX  L      ERROR,1     SET OVERLAP ERROR ON
655         BSC  L      OUT         BR TO READ NEXT PHASE
656   *                 CHECK PRECISION OF CONSTANT
657   DATA  LD   L      CCWD        CONTROL CARD WORD
658         SLA         13          PRECISION BIT TO SIGN POS
659         BSC  L      DATA1,-     BR IF STANDARD PRECISION
```

```
660  *                    PLACE EXTENDED CONSTANT ON STRING
661          LD          JANS-1      BINARY EXPONENT AND ID
662          EOR         MASK        REMOVE REAL CONSTANT TAG
663          STO    1 2              BINARY EXPONENT ONLY
664          LD          JANS-3      MOST SIGNIFICANT HALF CON
665          STO    1 3
666          LD          JANS-2      LEAST SIGNIFICANT HALF CON
667          STO    1 4
668          LD          CONOP       OPERATOR
669          STO    1 1              PLACE ON STRING
670          MDX    1 5              INCR STMNT POINTER
671          MDX         RC22B       BR TO CHK IF CLOSE REQUIRE
672  *                    PLACE STANDARD CONSTANT ON STRING
673  DATA1   LD     L    JANS-3      MOST SIGNIFICANT HALF CON
674          STO    1 2
675          LD     L    JANS-2      LEAST SIGNIFICANT HALF CON
676          STO    1 3
677          LD          CONOP       OPERATOR
678          STO    1 1              PLACE ON STRING
679          MDX    1 4              INCR STMNT POINTER
680          MDX         RC22B       BR TO CHK IF CLOSE REQUIRE
681  *                    CONSTANTS AND WORK AREA
682  CONOP   DC          /5E         CONSTANT OPERATOR
683  SIGN1   DC          /8001       SYMBOL TABLE PT CONSTANT
684  NUMB    DC          0           NO. OF WDS TO CLOSE TEMP
685  CNT     DC          0           NO. OF WDS IN STMNT TEMP
686  TWO     DC          2           DECIMAL TWO CONSTANT
687  SEMIC   DC          /001E       SEMICOLON CONSTANT
688  *                    CLOSE UP STRING
689  CLSUP   STX    1    NUMB        CURRENT STMNT ADDRESS
690          LD          NUMB
691          A           CNT         NO. OF WORDS IN STMNT
692          STO         *+1
693  X2      LDX    L2 0             ADDR NEXT STMNT ON STRING
694          LD     L    EOFS        END OF STRING ADDR
695          S           X2+1        ADDR OF NEXT STMNT
696          STO         *+1
697          LDX    L3 0
698          MDX     3 1             NO. OF WORDS TO MOVE
699  LOOPY   LD      2 0             NEXT WD TO MOVE TO CLOSE
700          STO     1 0             CLOSE UP STRING
701          MDX     2 1             INCR MOVE FROM ADDR
702          MDX     1 1             INCR MOVE TO ADDR
703          MDX     3 -1            DECR WDS TO MOVE COUNTER
704          MDX         LOOPY       BR TO CONTINUE CLOSING
705          STX    L1 EOFS          NEW END OF STRING ADDRESS
706          LDX    I1 NUMB          RESTORE CURRENT STMNT ADDR
707          MDX    L    EOFS,-1     ADJUST END OF STRING ADDR
708          LD     I    IDSAV       STMNT ID WORD- ADJUST NORM
709          SRT         2           RIGHT JUSTIFY NORM
710          S           CNT         REDUCE NORM BY CLOSE AMOUN
711          SLT         2           REPOSITION TO NORMAL FORM
712          STO    I    IDSAV       RESTORE STMNT ID WORD
713          MDX     1 1             INCR STMNT POINTER
714          SLA         16          CLEAR ACCUMULATOR
715          STO         CNT         ZERO CNT
716          STO    L    SW6         ZERO STRING CLOSURE SWITCH
717          BSC    L    ORGIN+2     BR TO INITIALIZE PHASE
718  *                    CONSTANTS AND WORK AREA
719  JN7     DC          10          DECIMAL TEN MPY CONSTANT
```

```
720          BSS  E  0          MAKE ADDRESS BOUNDARY EVEN
721   JC020  DC      0          PARTIAL PRODUCT TEMPORARY
722          DC      0          PARTIAL PRODUCT TEMPORARY
723   QCON1  DC      0          CHARACTER UNPACK TEMPORARY
724          DC      0          CHARACTER UNPACK TEMPORARY
725   QCON2  DC      0          NEXT CHARACTER TEMPORARY
726   QSIGN  DC      /7E00      6 BIT CHARACTER MASK
727   QCON3  EQU     QSIGN
728   SW12   DC      0          WORD IN BUFFER SWITCH
729   SW14   DC      0          FIRST OF TWO WDS SWITCH
730   ABT    DC      0          CHARACTER POSITION POINTER
731   *              SUBROUTINE TO GET CHARACTERS
732   GET    DC      0          RETURN ADDRESS
733          MDX  L  GCNT,1     INCR CONSTANT CHARACTER CN
734          LD      SW12       WORDS IN BUFFER SWITCH
735          BSC  L  QAA,Z      BR IF WORDS IN BUFFER
736   QAB    LDX  2  2          SET INDEX REG TO 2
737          STX  2  ABT        CHARACTER POINTER EQUAL 2
738          LD   1             NEXT STMNT WORD
739          BSC  L  QAC,-      BR IF OPERATOR
740          MDX  L  SW12,1     SET SW12 ON, GET NEXT 2 WD
741          LD   1  1          LAST HALF OF NEXT 2 WD
742          RTE     16         SHIFT TO EXTENSION
743          SLT     1          SHIFT OUT NON-CHARACTER BI
744          LD   1             FIRST HALF OF NEXT 2 WD
745          STD     QCON1      TEMPORARY
746          MDX  1  1          MOVE POINTER
747   QAE    LDD     QCON1      CHARACTER BUFFER
748          AND     QSIGN      EXTRACT NEXT CHARACTER
749          BSC  L  QAG,+-     BR IF NO CHARACTER
750   SQAG   LDD     QCON1      CHARACTER BUFFER
751          STO     QCON2      HIGH WORD
752          SLT     6          SHIFT OFF HIGH CHARACTER
753          STD     QCON1      RESTORE TO BUFFER
754          LD      QCON2      HIGH WORD
755          AND     QCON3      EXTRACT NEXT CHARACTER
756          SRA     9          RIGHT JUSTIFY CHARACTER
757   QAD    BSC  I  GET        RETURN WITH CHAR IN ACC
758   QAC    MDX  1  1          MOVE STMNT POINTER
759          MDX     QAD        BR TO RETURN
760   QAA    MDX  L  ABT,-1     DECR CHAR POINTER
761          MDX     QAE        BR TO GET NEXT CHAR IF ANY
762          MDX  1  1          MOVE POINTER
763          LD      SW14       FIRST OF 2 WORD SWITCH
764          BSC  L  QAF,+-     BR IF WORKING ON FIRST OF
765   QAH    MDX  1  -1         DECR STMNT POINTER
766   QAG    SLA     16         CLEAR ACC
767          STO     SW14       SET SW14 TO FIRST OF TWO
768          STO     SW12       SET SW12 NO WDS IN BUFFER
769          MDX     QAB        BR TO SET CHAR POINTER =2
770   QAF    LD   1  -1         BACK UP PT TO OPERATOR
771          BSC  L  QAH,-      BR TO BACK UP POINTER
772          LDX  2  3
773          STX  2  ABT        SET CHAR PT =3
774          MDX  L  SW14,1     SET SW14 TO SECOND OF TWO
775          MDX     QAE        BR TO GET NEXT CHAR IF ANY
776   *              SUBROUTINE TO MULTIPLY 5 WORD
777   *              CONSTANT BY 10 AND ADD CONTENT OF AC
778   MULT   DC      0          RETURN ADDRESS
779          STO     JC020+1    NUMBER TO ADD TO PRODUCT
```

```
780           LDX    2 5           NO, WDS TO MPY COUNTER
781           LD     L2 JANS-5     NEXT LEAST SIGNIFICANT WOR
782           M         JN7        MPY BY 10
783           BSC       +Z         SKIP NEXT IF PLUS OR ZERO
784           A         JN7        ADD 10
785           AD        JC020      ADD MOST HSLF LAST PRODJCT
786           STO       JC020+1    STORE MOST SIGNIFICANT 16
787           SLT       16         LEAST SIGNIFICANT 16 BITS
788           STO    L2 JANS-5     RESTORE CONSTANT
789           MDX    2 -1          DECR COUNTER
790           MDX       MULT+3     BR TO CONTINUE MPY
791           BSC    I  MULT       RETURN
792  *                  END OF PHASE 06
793           BSS       50         PATCH AREA
794           BSS       OVERL-**320*3  PHASE-06 PATCH AREA
795           END       START
```