```
1  *    GENERAL AUTOMATION, INC.  ALL RIGHTS RESERVED
2  ******************************************************
3  *
4  *    PROGRAM NAME   FPH-25
5  *
6  *    MODEL NUMBER   8F025
7  *
8  *    PURPOSE        FORTRAN PHASE-25
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 ** LIST CONSTANTS
22 ******************************************************
23 *STATUS-VERSION 1, MODIFICATION 0
24 *
25 *FUNCTION/OPERATION-
26 *    * LIST CORE REQUIREMENTS
27 *    * CONVERT CONSTANTS TO DECIMAL AND LIST
28 *      CONSTANTS AND THEIR ADDRESSES AT USER S
29 *      REQUEST. REAL CONSTANTS ARE LISTED FIRST,
30 *      FOLLOWED BY INTEGER CONSTANTS
31 *
32 *ENTRY POINTS-
33 *    * NEQ - PHASE 25 IS LOADED BY PHASE 24 VIA A
34 *            CALL TO POLRX AND EXECUTION IS BEGUN
35 *            AT LOCATION LABLED NEQ.
36 *
37 *INPUT-
38 *    * THE STATEMENT STRING
39 *    * THE SYMBOL TABLE
40 *    * THE FORTRAN COMMUNICATIONS AREA
41 *
42 *OUTPUT-
43 *    * THE STATEMENT STRING
44 *    * THE SYMBOL TABLE
45 *    * THE FORTRAN COMMUNICATIONS AREA
46 *    * LISTING OF REAL AND INTEGER CONSTANTS
47 *    * LISTING OF CORE REQUIREMENTS
48 *
49 *EXTERNAL REFERENCES-
50 *    * SUBROUTINES-
51 *      POLRX
52 *
53 *EXITS-
54 *    * NORMAL-
55 *          PHASE 26 IS LOADED VIA A CALL TO THE
56 *          POLRX ROUTINE AND CONTROL IS PASSED.
57 *
58 *    * ERRORS-
59 *      OVERLAP-
```

```
*          IF A PREVIOUS PHASE HAS DETECTED AN
*          OVERLAP ERROR THEN AN IMMEDIATE EXIT
*          TAKES PLACE.  NO OVERLAP ERROR IS
*          DETECTED IN THIS PHASE
*      SYNTAX-
*          NO SYNTAX ERRORS ARE DETECTED IN PHASE
*
*TABLES/WORK AREAS-
*    * THE STATEMENT STRING
*    * THE SYMBOL TABLE
*    * THE FORTRAN COMMUNICATIONS AREA
*    * ORG+120, A 120 WORD MESSAGE BUILDING AND
*      OUTPUTTING BUFFER.
*
*ATTRIBUTES-N/A
*
*NOTES-PROGRAM SWITCHES
*      THE SWITCH USED IN THE PHASE IS TRANSFER=T
*      IF POSITIVE OR NORMAL=N IF ZERO.
*    * STPS
*      N=NORMAL=SYMBOL TABLE SCAN FOR LISTING
*        REAL CONSTANTS.
*      T=TRANSFER=SYMBOL TABLE SCAN FOR LISTING
*        INTEGER CONSTANTS.
**************************************************
*
         ABS      REF CORE
         HONG     MPX FORTRAN ** LIST CONSTANTS
*
*        SYSTEM AND FORTRAN EQUATES
*
MEMRY    EQU      ~~77FFF~~ CORE     MAXIMUM CORE SIZE
PHSIZ    EQU      4*320              MAXIMUM PHASE SIZE
OVERL    EQU      MEMRY-PHSIZ        PHASES 2-29 START
FCOM     EQU      OVERL-22           FORTRAN COMM. TABLE
PHNTB    EQU      FCOM-56            PHASE TABLE
ROLRX    EQU      PHNTB-50           INTERPHASE CALL
AREA     EQU      OVERL+3*320+100    PRINT DATA ADDRESS
PRINT    EQU      AREA+1             PRINT ENTRANCE
*
*        FORTRAN COMMUNICATIONS AREA
*
         ORG      FCOM               START FORTRAN COMM AREA
SOFS     BSS      1                  START OF STRING
EOFS     BSS      1                  END OF STRING
SOFST    BSS      1                  START OF SYMBOL TABLE
SOFNS    BSS      1                  LENGTH OF PROGRAM
SOFXT    BSS      1                  SIZE OF WORK AREA (VARIES)
SOFGT    BSS      1                  SIZE OF CONSTANTS AREA
EOFST    BSS      1                  END OF SYMBOL TABLE
COMON    BSS      1                  RELATIVE ENTRY POINT
CSIZE    BSS      1                  SIZE OF COMMON
ERROR    BSS      1                  ERROR FLAG
*                                    BIT 15 OVERLAP ERROR
*                                    BIT 14 OTHER ERROR
FNAME    BSS      1                  PROGRAM NAME
         BSS      1                  2ND WORD OF NAME
SORF     BSS      1                  SUBR (-) OR FUNC (+)
CCWD     BSS      1                  CONTROL CARD WORD
*                    BIT 15 TRANSFER TRACE
```

```
*                BIT 14 ARITHMETIC TRACE
*                BIT 13 EXTENDED PRECISION
*                BIT 12 LIST SYMBOL TABLE
*                BIT 11 LIST SUBPROGRAM NAMES
*                BIT 10 LIST SOURCE PROGRAM
*                BIT  9 ONE WORD INTEGERS
IOCS  BSS     1     IOC  IOCS CONTROL CARD WORD
*
*        SEE PHASE ONE FOR BIT PATTERNS
*
OFCNT BSS     1          DEFINE FILE COUNT
*
LCOMN BSS     2          INSKEL COMMON
*
ICCER BSS     2          IOCS CONTROL CARD ERROR
*
      BSS     2          SYSTEM LOADER USE
*
*                    END OF FORTRAN COMMUNICATION
*                    AREA
*
      ORG     OVERL
*
NEQ   BSC  L  ENT        BRANCH TO INITLZ PROGRAM
WDCNT DC      *-*        WORD COUNT OF PRINT BUFFER
BUF   DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
      DC      /4000      BLANK
*
*               INITIALIZE PHASE
*               CHECK FOR SYMBOL TABLE OVERLAP
*
*
*               INITIALIZE TRANSFERVECTOR
ENT   LDX  L3 Z          GET ADDR OF CONSTANT AREA
      STX  L3 NEQ        SAVE ADDR
*
      LD   L  ERROR      TEST ERROR SW FR PREV PHASE
      BSC  L  EXIT,Z     BR TO EXIT IF OVERLAP ERROR
*
*               PRINT CORE REQUIREMENTS IF REQUESTED
*               SET UP TO LIST CONSTANTS.
*               CHECK FOR EXTENDED PRECISION.
*               MAKE MODIFICATIONS TO CONSTANTS IF
*               *EXTENDED PRECISION.
*
```

```
L4011 LD     3 0          PUT ZERO IN
      STO    3 STPS-Z     *SYMBOL TABLE PASS SWITCH
*
*                 BR TO .STATE CORE REQUIREMENTS.
*                 IF NO LISTING REQUIRED
*
      LD     L CCWD       LOAD CONTROL CARD WORD
      SLA      12         SHIFT LIST SYM TBL BIT
      BSC    L COREQ,-    BR IF NO LIST REQUESTED
*
*
*                 TEST IF EXTENDED PRECISION
*
      SLA      1          SHIFT EXTENDED PREC BIT
      BSC    L L4012,-    BR IF NO EXTENDED PRECISION
*
*                 MODIFY PROGRAM IF
*                 EXTENDED PRECISION
*
      MDX    L RVARF,1    INCR REAL VARIABLE FORMAT
      MDX    L INCR,4     INCR FLT POINT FORMAT FROM
*                         20 TO 24
      MDX    L L4024,3    INCR NO. DIGITS PAST DECML
      MDX    L L4025,4    INCR FIELD WIDTH
*
      BSC    L L4012
      ORG      BUF 120    MAKE A MESSAGE AREA
*
TEXT1 DC       /D900      R
      DC       /C500      E
      DC       /C100      A
      DC       /D300      L
      DC       /4000      BLANK
      DC       /C300      C
      DC       /D600      O
      DC       /D500      N
      DC       /E200      S
      DC       /E300      T
      DC       /C100      A
      DC       /D500      N
      DC       /E300      T
      DC       /E200      S
*
PTEXT LD     3 PAPIN-Z    INT VALUE PRINT AREA POINT
      STO    L AREA       SAVE IN MESSAGE ADDRESS
      LDX    3 1          INCREMENT XR3
      STX    L3 WDCNT     WDCNT FOR BLNK LN PRINT
      BSI    L PRINT      BR TO PRINT BLNK LINE
      LDX    L3 TEXT1     PUT ADDR OF REAL CONSTANTS
      STX    L3 AREA      *MSG IN PRINT MSG LOC
      LDX    3 7          SET XR3 TO WDCNT OF 7
      STX    L3 WDCNT     STORE IT IN CALL
      BSI    L PRINT      BR TO PRINT THE HEADER
      LDX    L3 PHEAD+2   SET UP PHEAD ROUTINE TO
      STX    3 PHEAD+1    *CONT RATHER THAN BR OUT
      STX    L3 BLKPA     SET BLKPA TO RTN TO PHEAD+
      LDX    L3 Z         RESET TRANSFER VECTOR
      BSC    L BLKPA+1    BR TO BLANK OUT PRINT AREA
L4012 LD     L SOFXT      INITIALIZE LOCATION
      STO    3 CLOC-Z     OF CURRENT CONSTANT
```

```
240  L401X LDX   I1 SOFST     INITLZ SYM TBL POINTER
241        STX   L1 STP
242  L4013 LD     3 STP-Z     LOAD SYM TBL POINTER
243        S     L  EOFST     TEST FOR END OF SYMBOL TABL
244        BSC   L  L4033,    BR IF END OF SYMBOL TABLE
245  L4021 LD    I1 0         LOAD SYMBOL TABLE ID WORD
246        BSC   L  L4032,-   BR IF NOT CONSTANT
247        SLA      1
248        BSC   L  L4031, Z  BR IF INTEGER CONSTANT
249        LD    3  STPS-7    LD SYMBOL TABLE PASS SWITCH
250  *
251        BSC   L  L4032,Z   BR IF SWITCH NONZERO
252  ****************************************************
253  *
254  *              CONVERT REAL CONSTANT AND PRINT
255  *              THE CONSTANT AND ITS LOCATION
256  *
257  ****************************************************
258  *
259  *              PRINT HEADER IF NOT PRINTED BEFORE
260  *              NOTE--NEXT INSTRUCTION IS REPLACED B
261  *              *A  BSC  L  PHEAD 2 AFTER THE HEADER
262  *              *IS PRINTED.
263  *
264  PHEAD BSC   L  PTEX1
265  *
266  *
267  *              INSERT PARAMETERS FOR CONVERTING
268  *              OF FLOATING POINT CONST TO
269  *              E-TYPE, EBC-CODE
270  *
271        LD    3 RVARF-Z    INSERT VARIABLE FORMAT
272        AND   3 1          MASK OUT ALL BUT BIT 15
273        STO   L PREC       SAVE PRECISION
274        BSC   L L4022,E    BR IF EXTENDED PRECISION
275  *
276  *              INSERT NORMAL PREC CONSTANT
277  *              FROM SYMBOL TABLE
278  *
279        LD    I1 1         GET 1/2 OF CON FR SYM TBL
280        STO   L CONVT-3    SAVE IN CONVERT BUFFER
281        LD    I1 2         GET 2ND PART OF CONSTANT
282        SRT      8         SHIFT OFF BITS 8-15
283        SLA      8         SHIFT BITS 0-7 BACK
284        STO   L CONVT-2    SAVE IN CONVERT BUFFER
285        SLT      8         SHIFT BITS 8-15 BACK TO ACC
286        STO   L BIN        SAVE BINARY POINT
287        MDX      L4024     BR TO CONTINUE
288  *
289  *              INSERT EXTENDED PRECISION CONSTANT
290  *              FROM SYMBOL TABLE
291  *
292  L4022 LD    I1 0         GET SYM TBL ID WORD
293        SLA      8         SHIFT OFF BITS 0-7
294        SRA      8         RIGHT JUSTIFY REMAINDER
295        STO   L BIN        SAVE IN BINARY POINT
296        LD    I1 1         GET 1/2 OF CON FR SYM TBL
297        STO   L CONVT-3    SAVE IN CONVERT BUFFER
298        LD    I1 2         GET 2ND PART OF CONSTANT
299        STO   L CONVT-2    SAVE IN CONVERT BUFFER
```

```
300  *
301  *                INSERT OTHER PARAMETERS FOR CONVERT
302  L4024 LDX   2 6          NUMBER OF DIGITS AFTER DECM
303        STX   L2 DD         *POINT  6 OR 9
304  L4025 LDX   2 13         FIELD WIDTH
305        STX   L2 WW         * 13 OR 17
306        LD    3 1           GET A CONSTANT   1
307        STO   L  FRMAI      SET FLT PT INDICATOR
308        SLA     16          CLEAR ACC
309        STO   L  FORTP      SET DATA TYPE O/P   F TYPE
310        BSI   L  BINRY      BR TO CONVERT DATA TO DECML
311        LD    3 H7E00-Z     LOAD AN EQUAL SIGN
312        BSI   3 TOPAU-Z     BR TO PUT OUT EQUAL
313  *
314  *                CONVERT ADDRESS
315  *                AND MOVE TO PRINTAREA
316  *
317        BSI   3 ATPA-Z      BR TO CONVERT AND MOVE SUBR
318        BSI   3 TOPAB-Z     BR TO PUT OUT EXTRA BLANK
319  *
320        LD    3 CLOC-Z      LOAD CONSTANT LOCATION
321        A     3 RVARF-Z     INCR BY REAL VAR FMT BITS
322        STO   3 CLOC-Z      SAVE BACK IN CONSTANT LOC
323  *
324  *                INCREMENT PRINT AREA POINTER
325  *                AND TEST IF LINE IS FULL
326  *
327  CKEND LD    3 PAP-Z       LD PRINT AREA POINTER
328        S     3 LNEND-Z     SUBTRACT END OF LINE
329        BSC   L  L4032, Z   BR IF LINE NOT FULL
330        BSI   3 PRINN-Z     PRINT A LINE
331        MDX     L4032       BR TO CONTINUE
332  *
333  ****************************************************
334  HTES  DC        *-*       HEADER TEST SWITCH
335  *
336  L4031 LD    3 STPS-Z      LD SYMBOL TABLE PASS SWITCH
337        BSC   L  L4032, -   BR IF SWITCH ZERO
338        LD      HTES        LD HEADER TEST SWITCH
339        BSC   L  INTC,Z     BR IF NON-ZERO
340        BSI   3 BLKPA-Z     BLANK OUT PRINT AREA
341        BSI   3 PRINN-Z     PRINT BLANK LINE
342        LD    3 TEX2A-Z     GET ADDR .INTEGER CONSTANTS
343        BSI   3 PTEXT-Z     BR TO MOVE TEXT TO PR AREA
344        MDX   L  HTES,1     SET HEADER TEST SW NON-ZERO
345        BSI   3 PRINN-Z     PRINT HEADER
346  INTC  BSI   3 INTLS-Z     BR TO CONVERT INTEGER CONS
347        LD    1 1           GET WORD FROM SYMBOL TABLE
348        STO   L  CONVT-3    SAVE IN CONVERSION AREA
349        EOR   3 H8000-Z     REMOVE BIT 0
350        BSC   L  INTD,Z     BR IF WD NOT ZERO
351        LD    3 H8000-Z     LOAD BIT 0   1
352        SRT     1           PUT 1 S IN BITS 0 AND 1
353        MDX   L  BIN,1      INCREMENT BINARY POINTER
354        STO   3 CONVT-3-Z   SAVE CONVERT WORD
355  INTD  BSI   L  BINRY      BR TO CONVERT DATA TO DECIM
356        LD    3 H7E00-Z     LD   SIGN
357        BSI   3 TOPAU-Z     BR TO PUT IN PRINT BUFFER
358        BSI   3 ATPA-Z      MOVE ADDR OF CON TO PR AREA
359        MDX   L  CLOC,1     INCR CONSTANT LOCATION
```

```
360          MDX      CKEND    BR TO TEST IF PR LINE FULL
361   *
362   *               GET NEXT CONSTANT.
363   *               MOVE SYMBOL TABLE POINTER
364   *
365   L4032  BSI   3  MSTP-Z   MOVE SYMBOL TABLE POINTER
366          BSC  L   L4013    BR TO TEST FOR SYM TBL END
367   *
368   *               END OF SYMBOL TABLE ENCOUNTERED
369   *
370   L4033  LD    3  PAP-Z    GET PRINT AREA POINTER
371          S     3  PAPIN-Z  SUBTRACT INITIAL VAL PR ARE
372          BSI  L   PRINN*Z  PR LN IF DIFFERENCE GT ZERO
373   *
374   *               IF SYMBOL TABLE PASS SW IS NONZERO
375   *               GO TO LISTING OF CORE REQUIREMENTS
376   *
377          LD    3  STPS-Z   LD SYM TBL PASS SWITCH
378          BSC  L   COREQ,Z  BR IF NONZERO
379   *
380          LD    3  1        SET SYM TBL PASS SWITCH
381          STO   3  STPS-Z   *TO NONZERO
382   *
383          BSC  L   L401X    GO TO SECOND PASS OF
384   *                        SYM TBL  FOR LISTING OF
385   *                        *INTEGER CONSTANTS
386   *
387   *               LIST CORE REQUIREMENTS
388   *
389   COREQ  BSI   3  BLKPA-Z  BLANK PRINT AREA
390          BSI   3  PRINN-Z  PRINT BLANK LINE
391          LD    3  CORA-Z   GET ADDR .CORE REQUIREMENTS
392          BSI   3  PTEXT-Z  BR TO MOVE TEXT TO PR AREA
393          LDX  L1  FNAME-1  LD PROGRAM NAME POINTER
394          BSI   3  TOPA-Z   CONVERT CHAR 1-MOVE TO PRIN
395          LD    1  1        LD WD1 OF PROG NAME
396          SLA      6        SHIFT TO POSITION
397          BSI   3  TOPA-Z   CONVERT CHAR 2-MOVE TO PRIN
398          LD    1  2        LD WD2 OF PROG NAME
399          RTE      16       SHIFT TO EXTENSION
400          LD    1  1        LOAD WD1 OF PROG NAME
401          SLT      12       SHIFT ACC AND EXTENSION
402          BSI   3  TOPA-Z   CONVERT CHAR 3-MOVE TO PRIN
403          LD    1  2        LOAD WD2 OF PROG NAME
404          SLA      2        SHIFT TO POSITION
405          BSI   3  TOPA-Z   CONVERT CHAR 4-MOVE TO PRIN
406          LD    1  2        LOAD WD2 OF PROG NAME
407          SLA      8        SHIFT TO POSITION
408          BSI   3  TOPA-Z   CONVERT CHAR 5-MOVE TO PRIN
409          BSI   3  PRINN-Z  PRINT A LINE
410   *
411   *               COMMON SIZE TO PRINT AREA
412   *
413          LD    3  COMA-Z   GET ADDR .COMMON.
414          BSI   3  PTEXT-Z  BR TO MOVE TEXT TO PR AREA
415          BSI   3  INTLS-Z  INITLZ INTEGER CON CONVERT
416          LD   L   CSIZE    GET SIZE OF COMMON
417          STO  L   CONVT-3  SAVE IN CONVERSION AREA
418          BSI  L   BINRY     GO CONVERT TO DECIMAL
419   *
```

```
420  *                INSKEL COMMON SIZE TO PRINT AREA
421  *
422          LD     3 INSKA-Z    GET ADDR .INSKEL COMMON.
423          BSI    3 PTEXT-Z    MOVE TEXT TO PRINT AREA
424          BSI    3 INTLS-Z    INITLZ INTEGER CON CONVERT
425          LD   L   LCOMN      GET SIZE OF INSKEL COMMON
426          STO  L   CONVT-3    SAVE IT IN CONVERSION AREA
427          BSI  L   BINRY      GO CONVERT TO DECIMAL
428          BSI    3 PRINN-Z    GO PRINT A LINE
429  *
430  *                SIZE OF WORK AREA TO PRINT AREA
431  *
432          LD     3 VARA-Z     GET ADDR .VARIABLES.
433          BSI    3 PTEXT-Z    MOVE TEXT TO PRINT AREA
434          BSI    3 INTLS-Z    INITLZ INTEGER CON CONVERT
435          LD   L   SOFXT      GET SIZE OF WORK AREA
436          STO  L   CONVT-3    SAVE IN CONVERT WORD
437          BSI  L   BINRY      CONVERT TO DECIMAL
438  *
439  *                CONSTANTS AND PROGRAM AREA SIZE
440  *                TO PRINT AREA
441  *
442          LD     3 PROA-Z     GET ADDR .PROGRAM.
443          BSI    3 PTEXT-Z    MOVE TEXT TO PRINT AREA
444          BSI    3 INTLS-Z    INITLZ INTEGER CON CONVERT
445          LD   L   SOFNS      COMPUTE LENGTH OF PROGRAM
446  *                            AREA
447          S    L   SOFXT      SUBTRACT SIZE OF WORK AREA
448          STO  L   SOFNS      SAVE LENGTH OF PROGRAM AREA
449          STO  L   CONVT-3    SAVE IN CONVERSION AREA
450          BSI  L   BINRY      CONVERT TO DECIMAL
451          BSI    3 PRINN-Z    PRINT OUTPUT
452  *
453  *                CHECK FOR CORE SIZE ERRORS
454  *
455          SLA      16
456          STO      MSTP       CLEAR TEMPORARY ACCUMULATOR
457  *
458          LD   L   CSIZE      CHECK COMMON AREA SIZE
459          BSI      CS000      BR TO CHECK CORE SIZE ERROR
460  *
461          LD   L   SOFXT      CHECK VARIABLE AREA SIZE
462          BSI      CS000      BR TO CHECK CORE SIZE ERROR
463  *
464          LD   L   SOFNS      CHECK PROGRAM AREA SIZE
465          BSI      CS000      BR TO CHECK CORE SIZE ERROR
466  *
467          BSC  L   EXIT       EXIT
468  *
469  *
470  CS000   DC       *-*        LINK
471  *
472          BSC  L   CS050, Z   ERROR IF SIZE NEGATIVE
473  *
474          A        MSTP
475          STO      MSTP
476          BSC  L   CS050, Z   BRANCH IF OVERFLOW
477  *
478  CS010   BSC  I   CS000      RETURN
479  *
```

```
480  *                INDICATE SIZE ERROR
481  *
482  CS050 LD    L  FIVE       PUT ERROR CODE IN ERRO V1M
483        STO   L  SOFS       PARAMETER FOR PHASE 28 V1M
484        BSI   L  ROLRX      GET PHASE-28
485        DC       28
486  *
487  *
488  *
489  *                SUBR. MOVE SYMBOL TABLE POINTER
490  MSTP  DC         0        LINK
491        LD     1  0        GET SYM TBL ID WORD
492        AND    3  MASK3-Z  TEST FOR DIMENSION
493        BSC       Z        SKIP IF NOT DIMENSIONED
494        LD     3  CM3-Z    LD MINUS 3 IF DIMENSIONED
495        S      3  3        DECR BY 3
496        A      3  STP-Z    DECR SYM TBL PT BY 3 OR 6
497        STO    3  STP-Z    SAVE SYM TBL POINTER
498        LDX    I1 STP      PUT SYM TBL PT IN XR1
499        BSC    I  MSTP     RETURN
500  *
501  *                MOVE ADDRESS TO PAREA
502  ATPA  DC         *-*      LINK ENTRY POINT
503        LD     3  CLOC-Z   GET HEX LOC OF CURRENT CON
504        SRT       12       SHIFT OFF LOWER 3 HEX CHARS
505        BSI       TOPAH    BR TO PUT HEX CHAR PR AREA
506        SLT       4        SHIFT NXT CHAR FR EXTENSION
507        BSI       TOPAH    BR TO PUT HEX CHAR PR AREA
508        SLT       4        SHIFT NXT CHAR FR EXTENSION
509        BSI       TOPAH    BR TO PUT HEX CHAR PR AREA
510        SLT       4        SHIFT LAST CHAR FR EXTENSIO
511        BSI       TOPAH    BR TO PUT HEX CHAR PR AREA
512        BSI       TOPAB    PUT BLANK IN PRINT AREA
513        BSC    I  ATPA     RETURN
514  *
515  *                SET UP CONVERSION ROUTINE FOR
516  *                     INTEGER CONSTANTS.
517  *
518  INTLS DC         *-*      ENTRY POINT
519        LDX    2  6        SET FIELD WIDTH
520        STX    L2 WW       *TO 6
521        SLA       16       CLEAR 2ND WD OF CONVERSION
522        STO    L  CONVT-2  *AREA
523        STO    L  FRMAI    SET INTEGER CONVERSION CODE
524        LD     3  0143-Z   SET BINARY DECIMAL POINT
525        STO    L  BIN      *FOR INCREMENTING
526        LDX    2  2        SET DATA TYPE TO
527        STX    L2 FORTP    *I FORMAT
528        BSC    I  INTLS    RETURN
529  *
530  *                PUT TEXT IN PRINT AREA.
531  *                ADDRESS OF TEXT IN A-REG ON ENTRY
532  *
533  PTEXT DC         0        ENTRY POINT
534        STO       HTES1 1  SAVE ADDR OF TEXT
535  HTES1 LDX    I2 *-*      PUT WD COUNT IN XR2
536        SLT       16       CLEAR ACC
537        LDX    I3 PAP      PUT PR AREA POINTER XR3
538  LOOPP MDX    L  HTES1 1,1 INCR ADDR POINTER
539        LD     I  HTES1 1  LD PACKED CHARACTER
```

```
540          SRT      8          SHIFT RIGHT CHAR EXTENSION
541          SLA      8          SHIFT LEFT TO PRINT POSITIO
542          STO   3  0          SAVE IN PRINT LINE
543          SLT      16         SHIFT RIGHT CHAR TO POSITIO
544          STO   3  1          SAVE IN PRINT LINE
545          MDX   3  2          INCR PRINT LINE POINTER
546          MDX   2  -1         DECR WORD COUNT
547          MDX      LOOPP      LOOP IF COUNT NOT FINISHED
548          STX   3  PAP        SAVE NEW PRINT AREA POINTER
549          LDX   L3 Z          RESET XR3 FOR CONSTANTS
550     HTES3 BSC   I  PTEXT     RETURN
551     *
552     *
553     *                SUBROUTINE
554     *                MOVE CHAR IN ACC TO PR AREA UNCHANGE
555     *
556     TOPAU DC         0          LINK
557           STO        STOCH      SAVE CHAR TO BE PRINTED
558     TOPAX LD         STOCH      LD CHAR TO BE PRINTED
559           STO   I    PAP        SAVE IN PRINT AREA
560           MDX   L    PAP,1      MOVE PRINT AREA POINTER
561           BSC   I    TOPAU      RETURN
562     *
563     STOCH DC         *-*        CHARACTER TO BE PRINTED
564     H000F DC         /000F      CONSTANT
565     H000A DC         /000A      CONSTANT
566     H0039 DC         /0039      CONSTANT
567     *
568     *                SUBROUTINE
569     *                CONVERT CHAR IN ACC INTO
570     *                EBC-CODE, THEN MOVE TO PRINT AREA
571     *
572     TOPA  DC         0          LINK
573           AND   3    H3F00-Z    MASK BITS 2-7
574           BSC   L    TOPA2, -   BR IF BLANK
575     *
576     *                NOTE
577     *                IF OTHER SPECIAL CHARACTERS THAN
578     *                BLANK ARE EXPECTED, TESTING SHOULD
579     *                BE HERE
580           OR         HC000      ADD FIRST 2 BITS ALPHA-NOS
581     TOPA1 STO        STOCH      SAVE IN CHAR STORAGE WD
582           LD         TOPA       MOVE LINK
583           STO        TOPAU      *TO OUTPUT
584           MDX        TOPAX      BR TO PRINT CHAR IN EBC
585     TOPA2 LD      3  H4000-Z    LOAD EBC BLANK
586           MDX        TOPA1      BR TO PUT IN BUFFER
587     *
588     *
589     *                SUBROUTINE
590     *                MOVE HEX CHAR TO PRINT AREA
591     *
592     TOPAH DC         0          LINK
593           AND        H000F      SAVE ONLY BITS 12-15
594           S          H000A      TEST FOR CON LT A
595           BSC        Z          SKIP IF GT 9
596           A          H0039      RESTORE CON   EBC BITS
597           A       3  1          ADD 1
598           SLA        8          SHIFT TO EBC FORMAT
599           BSI        TOPA       CONVERT AND MOVE
```

```
600          BSC  I  TOPAH    RETURN
601   *
602   *                 PUT ONE BLANK IN PRINT AREA
603   TOPAB  DC       0        ENTRY POINT
604          SLA      16       CLEAR ACC
605          BSI      TOPA     PUT OUT BLANK
606          BSC  I   TOPAB    RETURN
607   *
608   *                 SUBROUTINE
609   *                 BLANK TO PRINT AREA
610   *
611   BLKPA  DC       0        LINK
612          LD     3 H4000-Z  PUT EBC BLANK ACC
613          LDX  L3  121      LD BFR SIZE IN XR3
614   BLKPI  STO  L3  BUF-1    SAVE BLANK IN PR AREA
615          MDX    3 -1       DECR BUFFER COUNT
616          MDX      BLKPI    CONTINUE IF CNT NOT 0
617          LDX  L3  Z        RESET XR3 WITH CON ADDR
618          BSC  I   BLKPA    RETURN
619   *
620   *                 CONSTANTS AREA
621   *                 NOTE--GREAT CARE SHOULD BE USED
622   *                 IN CHANGING CONSTANTS AROUND AS SOME
623   *                 OF THE LOCATIONS ARE REFERENCED RLTV
624   *                 TO THE START OD THE AREA.
625   *                 XR3 IS LOADED WITH THE ADDRESS OF Z
626   *                 FOR REFERENCE TO THE CONSTANTS
627   *
628   Z      DC       0         CONSTANT0
629          DC       1         CONSTANT ONE
630          DC       2         CONSTANT TWO
631          DC       3         CONSTANT THREE
632          DC       4         CONSTANT FOUR
633   PAPIN  DC       BUF       INITIAL VALUE OF PAP
634   PAP    DC       BUF       PRINT AREA POINTER
635   INCR   DC       20        INCREMENT OF PRINT AREA
636   *                           IS CHANGED INTO 24 WHEN
637   *                         EXTENDED PRECISION
638   HC000  DC       /C000     CONSTANT MASK
639   RVARF  DC       /0002     REAL VARIABLE FORMAT
640   STPS   DC       0         SYMBOL TABLE PASS SWITCH
641   CLOC   DC       0         LOCATION OF CURRENT CONSTAN
642   STP    DC       0         SYMBOL TABLE POINTER
643   MASK3  DC       /1800     CONSTANT MASK
644   CM3    DC       /FFFD     CONSTANT   MINUS 3
645   H7E00  DC       /7E00     EQUAL SIGN
646   LNEND  DC       BUF+60     LINE END CONSTANT
647   D143   DC       143       CONSTANT
648   H3F00  DC       /3F00     CONSTANT MASK
649   H4000  DC       /4000     CONSTANT MASK
650   TEX2A  DC       TEXT2     ADDR OF MSG TO PRINT
651   CORA   DC       CORTX     ADDR OF MSG TO PRINT
652   COMA   DC       COMTX     ADDR OF MSG TO PRINT
653   INSKA  DC       INSKX      ADDR OF MSG TO PRINT
654   VARA   DC       VARTX     ADDR OF MSG TO PRINT
655   PROA   DC       PROTX     ADDR OF MSG TO PRINT
656   D60    DC       60        CONSTANT
657   *
658   *                 SUBROUTINE
659   *                 SET UP TO PRINT A LINE AND GO PRINT
```

```
660  *                *THE LINE
661  *
662  PRINN DC         *-*        ENTRY POINT
663        LD     3   PAPIN-Z    GET INT VALUE OF PAP
664        STO    3   PAP-Z      RESET PAP
665        STO  L     AREA       SAVE PRINT AREA
666        LD     3   D60-Z      SET WORD COUNT
667        STO  L     WDCNT      * 60
668        BSI  L     PRINT      BR TO PRINT A LINE
669        BSI    3   BLKPA-Z    CLEAR PRINT AREA
670        BSC  I     PRINN      RETURN
671  *
672  *
673  C128  DC         128        CONSTANT
674  MONE  DC         -1         CONSTANT
675  TEN   DC         10         CONSTANT
676  H00C5 DC         /00C5      CONSTANT MASK
677  MULT  DC         *-*        RETURN ADDRESS
678        BSC  L     MULT1      GO TO MULTIPLY SUBROUTINE
679  NORM  DC         *-*        RETURN ADDRESS
680        BSC  L     NORM1      GO TO NORMALIZE ROUTINE
681  FIVE  DC         5          CONSTANT
682  H8000 DC         /8000      CONSTANT MASK
683  MINUS DC         /0060      EBC MINUS RIGHT JUSTIFIED
684  BLANK DC         /0040      EBC BLANK RIGHT JUSTIFIED
685  SIX   DC         6          CONSTANT
686  *
687  *                DATA STORAGE AREA FOR BINARY-DECIMAL
688  *                CONVERSION SUBROUTINE
689  *
690  CHAR  DC         0          CHARACTER STORAGE
691  FRMAI DC         0          INTEGER OR DECML FLAG
692  FORTP DC         0          TYPE OF DATA TO BE OUTPUT
693  LKTYP DC         0          TEMPORARY STORAGE AREA
694  X     EQU        CHAR       REFERENCE PT FOR DATA AREA
695        BSS  E     0
696  TWO   DC         2          CONSTANT              EVEN LOC
697  ONE   DC         1          CONSTANT              ODD  LOC
698  H004B DC         /004B      CONSTANT              EVEN LOC
699  H00F0 DC         /00F0      CONSTANT              ODD  LOC
700  JCON  DC         0          CHAR STO MULTIPLY     EVEN LOC
701        DC         0          *SUBROUTINE           ODD  LOC
702  BB    DC         0          EBC REPRESENTATION    EVEN LOC
703        DC         0          *OF EXPONENT          ODD  LOC
704  EENCT DC         0          EXPONENT SIGN         EVEN LOC
705        DC         0          FIVE                  ODD  LOC
706        DC         0          *WORD                 EVEN LOC
707        DC         0          *CONVERSION           ODD  LOC
708        DC         0          *AREA                 EVEN LOC
709  CONVT DC         0          *                     ODD  LOC
710  SIGN  DC         0          SIGN COUNT            EVEN LOC
711  DIVCT DC         0          DIVIDE COUNT          ODD  LOC
712  MANSN DC         0          MANTISSA SIGN         EVEN LOC
713  TEMP  DC         0          TEMPORARY STORAGE     ODD  LOC
714  WW    DC         0          FIELD WIDTH           EVEN LOC
715  DD    DC         0          DECIMAL COUNT         ODD  LOC
716  PREC  DC         0          PRECISION             EVEN LOC
717  BIN   DC         0          BINARY POINT CNT      ODD  LOC
718  *
719  *                SUBROUTINE
```

```
720  *              MULTIPLY 5 CONVERSION CHARACTERS BY
721  *              10
722  *
723  MULTI  STO   2 JCON 1-X STORE CHARACTER
724         LDX   1 5        SET UP INDEX LOOP FOR 5
725  MULT3  LD    L1 EENCT    GET WDS TO CONVERT
726         M     2 TEN-X     MULTIPLY BY 10
727         BSC     Z         SKIP IF RESULT NOT -
728         A     2 TEN-X     ADD 10 TO RESULT
729         AD    2 JCON-X    ADD PRESENT CHARACTER
730         STO   2 JCON 1-X STORE MOST SIGNIFICANT BIT
731         SLT     16        SHIFT LEAST SIGNIFICANT TO
732         STO   L1 EENCT    *ACC AND SAVE
733         MDX   1 -1        DECR WD CNT BY 1
734         MDX     MULT3     LOOP UNLESS CNT EXHAUSTED
735         BSC   I MULT      RETURN TO CALLER
736  *
737  *              NORMALIZATION SUBROUTINE
738  *
739  NORMI  LD    2 CONVT-4-X IS LEFTMOST WORD OF NO.
740         BSC   L NORMT, -  YES, GO CHECK FURTHER
741         BSI     NORRT     NO, GO NORMALIZE RIGHT
742         MDX     NORM1     BRANCH BACK TO CHECK NORM
743  NORMT  LD    2 CONVT-3-X IS 2ND WORD FROM LEFT NEG.
744         BSC   I NORM, Z   YES, RETURN TO CALLING PROG
745         BSI     NORLT     NO, GO TO NORM LEFT
746         MDX     NORMI     BRANCH BACK TO CHECK NORM
747  *
748  *              NORMALIZE LEFT SUBROUTINE
749  *
750  NORLT  DC      *-*       NORMALIZE LEFT
751         SLT     32        SHIFT
752         LDX   1 5         *ANSWER
753  NORM7  LD    L1 EENCT    *LEFT
754         RTE     31        *ONE
755         STO   L1 EENCT    *BIT
756         SLT     15        THIS MEANS ALL 5
757         MDX   1 -1        *WORDS OF
758         MDX     NORM7     *CONVERSION AREA
759         LD    2 BIN-X     DECR BINARY
760         S     2 ONE-X     *POINT COUNTER
761         STO   2 BIN-X     *BY 1
762         BSC   I NORLT     RETURN
763  *
764  *              NORMALIZE RIGHT SUBROUTINE
765  *
766  NORRT  DC      *-*       NORMALIZE RIGHT ENTRY
767  NORM4  LDX   1 -5        SHIFT
768         SLT     32        *ALL
769         LD    L1 SIGN     *FIVE
770         RTE     1         *WORDS
771         STO   L1 SIGN     *OF CONVERSION
772         RTE     15        *AREA
773         MDX   1 1         *RIGHT
774         MDX     NORM4 2   *ONE BIT
775         LD    2 BIN-X     INCR BINARY POIN
776         A     2 ONE-X     *COUNTER
777         STO   2 BIN-X     *BY ONE
778         BSC   I NORRT     RETURN
779  *
```

```
780  *                    DIVIDE 5 WORD ANSWER BY 10
781  *
782  DIVID DC       *-*       ENTRY POINT
783        LDX   1  -4        LOAD INDEX FOR FOUR DIVIDES
784        SLT      32        CLEAR ACC AND EXTENSION
785  DIV1  LD    2  ONE-X     LD CONSTANT ONE
786        STO   2  LKTYP-X   SAVE LKTYP
787        LD    L1 SIGN      LD NO.TO BE DIVIDED
788        RTE      16        EXCHANGE ACC AND EXTENSION
789        S     2  FIVE-X    SUBTRACT 5 FOR ROUNDING
790        BSC   L  * 2, Z    BR IF NEGATIVE
791        MDX   L  LKTYP,-2  SET LKTYP  -1 AND SKIP
792        A     2  FIVE-X    RESTORE 5 PREVIOUSLY DELETE
793        D     2  TEN-X     DIVIDE BY 10
794        MDX   L  LKTYP,-2  SET LKTYP  -1,SKIP 00-3,CON
795        OR    2  H8000-X   MASK IN NEGATIVE SIGN
796        STO   L1 SIGN      SAVE NO BACK IN AREA
797        MDX   1  1         INCR AREA POINTER
798        MDX      DIV1      BR IF LOOP NOT FINISHED
799  RETZY BSC   I  DIVID     RETURN TO CALLING PROGRAM
800  *
801  *                  ENTRY POINT TO BINARY-DECIMAL
802  *                  CONVERSION SUBROUTINE
803  *
804  BINRY DC       *-*       ENTRY POINT
805        STX   L1 IR13 1    SAVE XR1
806        STX   L2 IR13 3    SAVE XR2
807        LDX   L2 X         GET CON AREA POINTER
808        LDX   I1 FORTP     GET TYPE OF DATA TO BE O/P
809        BSC   I1 *         BR TO CORRESPONDING PROG LO
810        DC       BIN4      E TYPE FORMAT
811        DC       BIN7      F TYPE FORMAT
812        DC       BIN3      I TYPE FORMAT
813  BIN14 LD    2  BIN-X     NORM BINARY PT TO 128
814        S     2  C128-X    IS POINT GR 128
815        BSC      Z         SKIP IF YES
816        MDX      BIN17     NO,GO TO NORM RIGHT
817        BSC   L  BIN19,    BR IF BINARY PT   128
818        BSI      NORLT     BINARY PT GT 128, NORM LEFT
819        MDX      BIN14     BR BACK TO CHECK BINARY PT
820  BIN17 BSI      NORRT     BR TO NORMALIZE RIGHT
821        MDX      BIN14     BR BACK TO CHECK BINARY PT
822  *
823  *                  E TYPE DATA OUTPUT
824  *
825  BIN4  LD    2  SIX-X     SET UP COUNT FOR 2 SIGNS
826        MDX      BIN7 1    GO CHECK INPUT MODE
827  *
828  *                  I TYPE DATA OUTPUT
829  *
830  BIN3  LD    2  MONE-X    SET UP A FALSE DECIMAL
831        STO   2  DD-X      *COUNT FOR
832  BIN7  LD    2  TWO-X     *COMMON CALCULATION FOR
833        STO   2  TEMP-X    *AVAILABLE PR POSITIONS
834  BIN5  LD    2  WW-X      LOAD FIELD WIDTH
835        S     2  DD-X      *LESS DECIMAL COUNT
836        S     2  TEMP-X    *LESS NO. SIGNS TO GET
837        STO   2  TEMP-X    *NO. AVAIL PRINT POSITIONS
838        SLT      32        CLEAR ACC AND EXTENSION
839        STO   2  CONVT-4-X CLR 1ST WD CONVERSION AREA
```

```
840          STD   2 CONVT-1-X CLR  4TH AND 5TH WORDS
841          STO   2 DIVCT-X  CLEAR DIVIDE COUNT
842          LD    2 BLANK-X  PUT BLANKS IN
843          STO   2 EENCT-X  *EXPONENT AND
844          STO   2 MANSN-X  *MANTISSA SIGNS
845          LDX   1 2        SET XR1 TO NO. WDS TO CHK
846 BIN52    LD    L1 EENCT 1 CHECK CONVERSION WD
847          BSC      Z       SKIP IF ZERO
848          MDX      BIN51   BR OUT IF NOT
849          MDX   1 -1       DECR WD COUNT
850          MDX      BIN52   BR BACK IF MORE WDS TO CHEC
851          LDD   2 H00F0-X  LOOP FINISHED, PUT EBC 0
852          STD   2 BB-X     *IN O/P EXPONENT
853          LD    2 FORTP-X  CHECK FOR I TYPE FORMAT TO
854          S     2 TWO-X    *OUTPUT ZEROS INSTEAD OF
855          BSC      Z       *BLANKS FOR A REAL
856          MDX      BIN19   *ZERO COMPUTATION
857          LD    2 ONE-X    SET FLAG TO OUTPUT
858          STO   2 DIVCT-X  *ZEROS
859          MDX      BIN19   BRANCH TO OUTPUT
860 BIN51    LDD   2 CONVT-3-X LD MANTISSA  ACC EXTENSION
861          BSC      -       SKIP IF NEGATIVE
862          MDX      BIN9    BR TO NORMALIZE
863          MDX   L  MANSN,32 SET MANTISSA SIGN NEGATIVE
864          SLT      32      CLEAR ACC-EXTENSION TO GET
865          SD    2 CONVT-3-X *ABSOLUTE VALUE OF MANTISS
866 BIN9     SLT      1       SHIFT OUT SIGN BIT
867          STD   2 CONVT-3-X SAVE ABS VALUE W/O SIGN
868          LD    2 FRMAT-X  TEST FOR INTEGERS
869          BSC   L  BIN13, - BR IF INTEGERS
870          LD    2 ONE-X    SET UP FOR ROUNDING, ACC 1
871          MDX   L  PREC    IS PRECISION EXTENDED
872          SRT      8       YES, SHIFT TOTAL OF 15
873          SRT      7       NO, SHIFT 7
874          AD    2 CONVT-3-X ADD 2WDS TO BE CONVERTED
875          BSC      C       TEST FOR OVERFLOW
876          MDX      * 1     OVERFLOW,BRANCH OVER NXT WD
877          MDX      IR11    NO OVERFLOW, BRANCH OUT
878          SRT      1       SHIFT 1 MORE TO ROUND
879          OR    2 H8000-X  PUT BIT IN SIGN POSITION
880          STD   2 CONVT-3-X SAVE   CONVERSION WD
881          LD    2 BIN-X    INCR BINARY
882          A     2 ONE-X    *POINT COUNTER
883          STO   2 BIN-X    *BY 1
884          MDX      BIN13 1 BR PAST NXT 2 WDS
885 IR11     STD   2 CONVT-3-X STORE IN CONVERSION
886 BIN13    LD    2 BIN-X    IS BINARY PT GE 128
887          S     2 C128-X   GREATER THAN 128
888          BSC      Z       SKIP IF GE 128
889          MDX      BIN10   GO HANDLE
890          BSC   L  BIN11,  BR IF   128
891 BIN12    BSI      DIVID   GO DIVIDE ANSWER BY 10
892          BSI   2 NORM-X   GO NORMALIZE ANSWER
893          MDX   L  DIVCT, 1 INCR DIVIDE COUNT BY 1
894          NOP
895          LD    2 BIN-X    TEST IF BINARY PT LE 128
896          S     2 C128-X   *AFTER DIVISION
897          BSC      -Z      SKIP IF YES
898          MDX      BIN12   NO, DIVIDE AGAIN
899 BIN11    LD    2 FORTP-X  TEST OUTPUT F TYPE
```

```
900          BSC      Z          SKIP IF E TYPE
901          MDX      BIN14      NOT E TYPE, BR TO NORM
902          LD     2 DIVCT-X    TEST DIVIDE COUNT
903          BSC  L   BIN16,-    IF POSITIVE OR ZERO, BR OUT
904   BIN15  LD     2 MINUS-X    PUT NEGATIVE SIGN OUT FOR
905          STO    2 EENCT-X    *EXPONENT SIGN
906          SLA      16         CLEAR ACC
907          S      2 DIVCT-X    GET ABS VALUE OF DIVIDE CNT
908   BIN16  RTE      16         PUT DIVIDE CNT IN EXTENSION
909          SLA      16         CLEAR ACC
910          STO    2 DIVCT-X    CLEAR DIVIDE COUNT WORD
911          D      2 TEN-X      DIVIDE DIVIDE CNT BY 10
912          AD     2 H00F0-X    ADD EBC 0 TO CONVERT TO EBC
913          STD    2 BB-X       SAVE EBC EXPONENT
914          MDX      BIN14      BRANCH TO NORMALIZE
915   BIN10  LD     2 FORTP-X    TEST FOR E TYPE OUTPUT
916          BSC      Z          SKIP IF YES
917          MDX      BIN14      NO, GO HANDLE NEG F TYPE
918          BSI    2 MULT-X     MULTIPLY ANSWER BY 10
919          BSI    2 NORM-X     GO NORMALIZE ANSWER
920          MDX  L   DIVCT,-1   DECR DIVIDE COUNT BY 1
921          MDX      BIN13      GO CHECK BINARY POINT
922          MDX      BIN13      *AGAIN AFTER MULTIPLYING
923   *
924   *                SUBROUTINE
925   *                OUTPUT CHARACTER TO PRINT BUFFER
926   *
927   CHAR0  DC       *-*        ENTRY POINT
928          SLA      8          SHIFT CHAR TO LEFT HALF OF
929          STO  I   PAP        *WD AND STORE IN PRINT BFR
930          MDX  L   PAP,1      INCR BUFFER POINTER
931   CHAR3  BSC  I   CHAR0      RETURN TO CALLING PROGRAM
932   *
933   *                OUTPUT CONVERTED DECIMAL NUMBER
934   *
935   BIN19  LD     2 TEMP-X     GET NO. AVAILABLE POSITIONS
936          S      2 DIVCT-X    COMPARE WITH DIVIDE COUNT
937          BSC                 SKIP IF NO. POSITIONS RIGGE
938          MDX      BIN18      BR TO PUT OUT CHARACTERS
939          STO    2 TEMP-X     STORE EXCESS NO. POSITIONS
940   BIN20  LD     2 BLANK-X    PUT OUT NO. OF
941          BSI      CHAR0      *BLANK CHARACTERS EQUAL
942          MDX  L   TEMP,-1    *TO EXCESS
943          MDX      BIN20      CONTINUE UNTIL LOOP DONE
944   BIN18  LD     2 MANSN-X    LD MANTISSA SIGN
945          BSI      CHAR0      PUT IN PRINT BUFFER
946          LD     2 FRMAI-X    CHECK FOR INTEGER
947          BSC      Z          SKIP IF IT IS AN INTEGER
948          MDX      * 2        ELSE, BRANCH AHEAD
949          LD     2 MONE-X     SET LARGE VALUE IN LAST
950          STO    2 CONVT-X    *WORD OF CONVERSION AREA
951          LD     2 DIVCT-X    LD DIVIDE COUNT
952          BSC                 SKIP IF NOT   ZERO
953          MDX      BIN23      GO HANDLE ZERO
954   BIN28  SLA      16         CLEAR ACC
955          STO    2 CONVT-4-X  CLEAR CONVERSION WORD 1
956          BSI    2 MULT-X     MULTIPLY BY 10
957          OR     2 H00F0-X    ADD EBC 0 TO CONVERT
958          BSI      CHAR0      *OVERFLOW, O/P CHAR
959          MDX  L   DIVCT,-1   DECR DIVIDE COUNT BY 1
```

```
960           MDX       BIN28     BR TO HANDLE MULTIPLY
961   BIN23   LD    2   DD-X      TEST DECIMAL COUNT OF FMT
962           BSC       Z         SKIP IF 0 OR POSITIVE
963           MDX       BIN25     IF NOT, BR TO HANDLE I TYPE
964           LD    2   H004B-X    GET EBC DECIMAL POINT
965           BSI       CHARO     MOVE TO PRINT BUFFER
966           LD    2   DD-X      LD DECIMAL COUNT
967           BSC                 SKIP IF GT 0
968           MDX       BIN25     ZERO, BRANCH AHEAD
969           STO   2   DIVCT-X   PUT NO. DECML DIGITS-DIVCT
970           LD    2   MONE-X    PUT -1 IN
971           STO   2   DD-X      *NO. OF DECML DIGITS
972           MDX       BIN28     GO PROCESS
973   BIN25   LD    2   FORTP-X   GET OUTPUT TYPE
974           BSC       Z         SKIP IF R TYPE
975           MDX       IR13      GO RETURN TO CALLER
976           LD    2   H00C5-X   LOAD EBC E
977           BSI       CHARO     PUT IN PRINT BUFFER
978           LD    2   EENCT-X   LOAD EXPONENT SIGN
979           BSI       CHARO     PUT IN PRINT BUFFER
980           LD    2   BB-X      LOAD 1 WD OF EXPONENT VALUE
981           BSI       CHARO     PUT IN PRINT BUFFER
982           LD    2   BB 1-X    LOAD 2 WD OF EXPONENT VALUE
983           BSI       CHARO     PUT IN PRINT BUFFER
984   IR13    LDX   L1  *-*       RESTORE XR1
985           LDX   L2  *-*       RESTORE XR2
986           BSC   I   BINRY     RETURN TO CALLER
987   *
988   *                 MESSAGES TO BE PRINTED THIS PHASE
989   *
990   *
991   TEXT2   DC        9           WORD COUNT
992           EBC       .INTEGER CONSTANTS .
993   CORTX   DC        11          WORD COUNT
994           EBC       .CORE REQUIREMENTS FOR .
995   COMTX   DC        4           WORD COUNT
996           EBC       . COMMON .
997   INSKX   DC        8           WORD COUNT
998           EBC       .  INSKEL COMMON .   MESSAGE
999   VARTX   DC        6           WORD COUNT
1000          EBC       .  VARIABLES .
1001  PROTX   DC        5           WORD COUNT
1002          EBC       .  PROGRAM .
1003  *
1004  *                 PHASE EXIT
1005  *
1006  EXIT    BSI   L   BLKPA       SET PRINT BUFFER BLANK
1007          BSI   L   PRINN       GO PRINT THE BLANK LINE
1008          BSI   L   ROLRX       CALL DOWN PHASE 26
1009          DC        26              NEXT PHASE NUMBER
1010          BSS       OVERL-*+320*3  PHASE-25 PATCH AREA
1011          END       NEQ
```