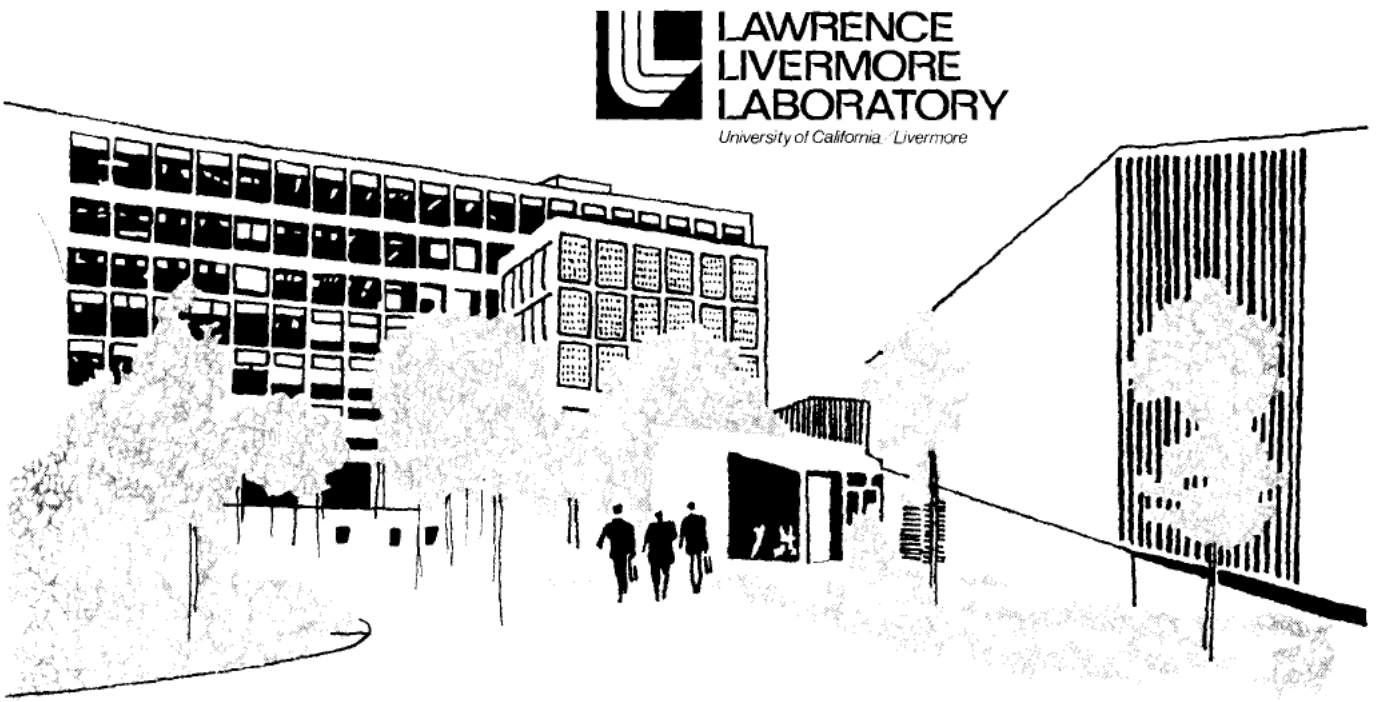
UCRL-51940

FLOATING-POINT PACKAGE FOR INTEL 8008 AND 8080 MICROPROCESSORS

Michael D. Maples

October 24, 1975

Prepared for U.S. Energy Research & Development Administration under contract No. W-7405-Eng-48



Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy $___*; Microfiche $2.25

*Pages	NTIS Selling Price
1-50	$4.00
51-150	$5.45
151-325	$7.60
326-500	$10.60
501-1000	$13.60

Distribution Category
UC-32

LAWRENCE LIVERMORE LABORATORY
University of California / Livermore, California / 94550

UCRL-51940

FLOATING-POINT PACKAGE FOR INTEL 8008 AND 8080 MICROPROCESSORS

Michael D. Maples

MS. Date: October 24, 1975

Contents

FLOATING-POINT PACKAGE FOR INTEL 8008 AND 8080 MICROPROCESSORS

Abstract

The Lawrence Livermore Laboratory has used a scientific-notation mathematics package that performs floating-point arithmetic with Intel 8008 and 8080 microprocessors. The execution times for the mathematical operations -- add, subtract, multiply, divide, and square root -- range from 3 to 77 ms. Instructions for using the floating-point package and a source listing of it are included.

Introduction

For the last two years, Lawrence Livermore Laboratory has used a scientific-notation mathematics package (floating-point package) with the Intel 8008 and 8080 microprocessors.* This package allows addition, subtraction, multiplication, division, and square root operations. Table 1 shows the execution times for these operations. The program listing of the complete 8080 floating-point package is in the Appendix. The package uses some I/O calls from an octal debug routine (ODT) that has become a standard part of all inhouse microcomputers, but this need not be necessary. The appropriate ODT calls (6 or 7) in the I/O routines can easily be placed by assembly language equivalents.

Table 1. Worst-case execution times for the 8080 microprocessor using a 0.5-μs clock with the package in programmable read-only memory (PROM).

Operation	Execution times (ms)
Add	3
Subtract	3
Multiply	7
Divide	8
Square root	77

*Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Energy Research & Development Administration to the exclusion of others that may be suitable.

The floating-point package uses 24 bits of mantissa for approximately 7-1/2 digits of accuracy in expressing numeric data. Obviously, this decreases rapidly when complex iterative computations are used. Nevertheless, the package is functioning quite satisfactorily in many experiments with accuracy requirements of one part per hundred thousand.

The package also indicates underflows and overflows by placing zeros in the mantissa and a 100 (octal) in the exponent word.

Selection and Use of Operations

All registers described in this paper point to four-word internal mathematical storage areas unless otherwise stated. Also, before performing any mathematical operation, all needed operands must be placed in the same random access memory (RAM) along with any needed scratch areas (i.e., all must reside in the same page of RAM).

The first problem is how to get the decimal numbers into the correct format for use in the floating-point package. The routine INPUT performs the conversion for all teletypewriter input. Also, it easily adapts to converting any BCD numeric inputs from either digital panel meters (DPM) or thumbwheel switches. To use INPUT, set the L-register to point at the location in RAM where the result of the conversion is to be placed and set the C-register to point to another location in RAM where intermediate steps are to be calculated. Then do a call to the INPUT routine that does the appropriate conversion (see Table 2).
The resulting floating-point number has three 8-bit words of mantissa and a fourth word that contains 6 bits of exponent, 1 bit for mantissa sign, and 1 bit for exponent sign (see Fig. 1). Negative mantissa are indicated only by the sign bit as the mantissa itself is in sign-magnitude form. But the negative exponents are in twos complement form.

If an addition (LADD) is wanted, place the pointer to one addend in the L-register, the pointer to the other addend in the B-register, and a pointer in the C-register. The C-register points to a four-word scratch area used during the addition process. The result is pointed to by the L-register (see Table 3).

Table 2. Program for using INPUT routine. The scratch area is 17 (octal) bytes long but the converted number is only 4 bytes long.

Program	Comments
MVI H, SCRPG	;Set H to match scratch page (RAM).
MVI L, STWD	;Store floating-point number starting
;	;at STWD.
MVI C, SCR	;Scratch area.
CALL INPUT	

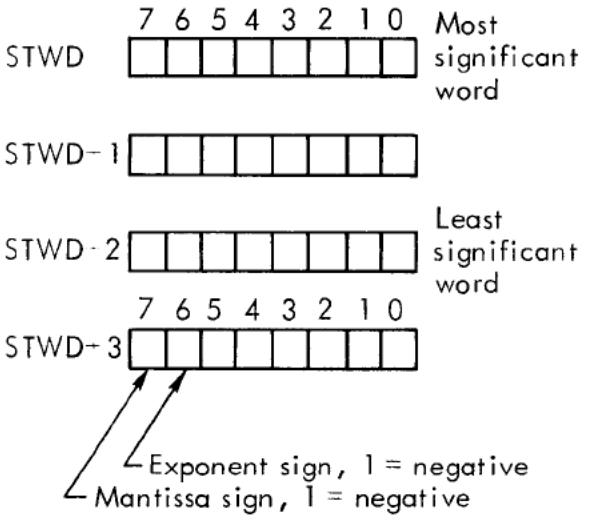


Fig. 1. Floating-point word format. This format allows representation of numbers from $\pm 6.46235 * 10^{-27}$ to $\pm 4.61168 * 10^{18}$.

The subtraction (LSUB) routine is very similar to the addition routine. The L-register holds the pointer to the minuend and the B-register holds the pointer to the subtrahend. The C-register once again is used as a four-word scratch area, and the result is placed in the area pointed to by the L-register, destroying the previous data residing there (see Table 4).

If a multiplication (LMUL) is wanted, again use the L-, B-, and C-registers. The pointer for the multiplicand resides in the L-register, the pointer for the multiplier in the B-register and the pointer to the result in the C-register (see Table 5).

Table 3. Assembly language setup for addition.

Program	Comments
MVI H, SCRPG	;Set H to scratch page (RAM).
MVI L, ADD1	;Pointer four-word addend and final
;	;result.
MVI B, ADD2	;Pointer 2nd four-word addend.
MVI C, SLR	;Four-word scratch area.
Call LADD	;Turn control over to addition
;	;routines.

Table 4. Assembly language setup for subtraction.

Program	Comments
MVI H, SCRPG	;Set H to match scratch page (RAM).
MVI L, SUB1	;Pointer to four-word minuend and
;	;final result.
MVI B, SUB2	;Pointer to four-word subtrahend.
MVI C, SCR	;Four-word scratch area.
Call LSUB	;Turn control over to subtraction
;	;routines.

Table 5. Assembly language setup for multiplication.

Program	Comments
MVI H, SCRPG	;Scratch page pointer (RAM).
MVI L, MLCAN	;Pointer to multplicant.
MVI B, Mlplr	;Pointer to multiplier.
MVI C, Rslt	;Pointer to result.
CALL LMUL	;Turn control over to multiply
;	;routine.

Division (LDIV) like multiplication uses the C-register to hold the pointer to the result (quotient). The L-register pointer refers to dividend and the B-register pointer refers to the divisor (see Table 6).

The square root routine (DSQRT) uses the L-register to point to the number to be converted, the B-register to point to the final converted number, and the C-register to point to a 16 octal-word scratch area (see Table 7).

The final routine is the output routine (CVRT). This routine converts the binary floating-point number pointed to in the L-register to its ASCII equivalent and types it out on the teletypewriter. This routine uses a 17 octal-word scratch area pointed to by the C-register (see Table 8). The final data is printed in scientific notation. The output routine like the INPUT routine is easily modified to output its data to an internal (memory) register for display on an LED display.

Table 9 gives a simple program that allows the user to check out the various routines and examine the various binary floating-point numbers.

Table 6. Assembly language setup for division.

Program	Comments
MVI H, SCRPG	;Scratch page pointer (RAM).
MVI L, dvdnd	;Pointer to dividend.
MVI B, dvsr	;Pointer to divisor.
CALL LDIV	;Turn control over to divide routine.

Table 7. Assembly language setup for square root.

Program	Comments
MVI H, SCRPG	;Scratch page pointer (RAM).
MVI L, NUM	;Number to be converted.
MVI B. CUTNM	;Converted number.
MVI C, SCR	;16 Octal-word scratch area.
CALL DSQRT	;Turn control over to square root
;	;routine.

Table 8. Assembly language to set up OUTPUT routine for its proper execution.

Program	Comments
MVI H, SCRPG	;Scratch page (RAM).
MVI L, OUTNM	;Number to be converted from floating
;	;to decimal and printed in scientific
;	;notation on teletypewriter.
MVI C, SCR	;17 octal-word scratch area.
CALL CVRT	;Turn control over to convert routine.

Acknowledgments

This package was based on a package purchased from David Mead of Recognition System. Major modifications were made by Hal Brand to allow ASCII I/O and a triple-precision mantissa. Overflow-underflow problems were resolved by Frank Olken. A hardy thanks is given to Eugene Fisher for foreseeing the need for such a package.

Table 9. Sample program that takes two operands from the teletypewriter, divides them, and outputs the result to the teletypewriter. This routine can be useful in becoming familiar with the different routines in the floating-point package.

Program	Comments
ORG 4500Q	;Program starts at location 100
;	;(octal) page 1.
SCRPG EQU 11Q	;Scratch page is page 11 (octal).
OP1 EQU OQ	;Starting location of operand 1.
OP2 EQU OP1 + 4	;Starting location of operand 2.
RSULT EQU OP2 + 4	;Starting location of result.
SCR EQU RSULT + 4	;Starting location of scratch area.
MVI H, SCRPG	;Set H register to RAM scratch page.
MVI L, OP1	;Pointer to operand 1.
MVI C, SCR	;Scratch area.
CALL Input	;Input operand 1 from teletypewriter.
MVI L, OP2	;Pointer to operand 2.
MVI C, SCR	;Scratch.
CALL INPUT	;Input operand 2 from teletypewriter.
MVI L, OP1	;Operand-1 pointer in L-register.
MVI B, OP2	;Operand-2 pointer in B-register.
MVI C, RSULT	;Result to C-register pointer.
CALL LDIV	;Divide OP1 by OP2 and place the
;	;result in RSULT.
MVI L, RSULT	;L-pointer now RSULT.
MVI C, SLR	;Scratch area.
CALL CVRT	;Output number starting in location
;	;RSULT to teletypewriter.
HALT	;End.

Appendix. Source Listing of Floating-Point Package

```
8080 MACRO ASSEMBLER, VER 2.2  ERRORS = 0 PAGE 1

                        ;       ////FLOATING POINT PACKAGE FOR THE MCS8
                        ;       ////BY DAVID MEAD
                        ;       ////MODIFIED BY HAL BRAND 9/6/74
                        ;       ////MODIFIED FOR 24 BIT MANTISSAS**********
                        ;       ////PLUS ADDED I/O CONVERSION ROUTINES
                        ;       ////NEW ROUTINE COMMENTS
                        ;       ////ARE PRECEEDED BY /
                        ;       ////OTHER CHANGES ARE NOTED BY **
                       ;        ////MODIFIED BY FRANK OLKEN 6/28/75
                        ;
                        ;
004400                          ORG 4400Q
                        ;
                        ;
000060                 OUTR EQU 60Q             ;/SET TO ODT'S TTY ROUTINE
000333                 INP  EQU 333Q            ;/SET READ TO ODT'S INPUT
000300                MINCH    EQU     300Q          ;MINIMUM CHARACTERISTIC WITH SIGN EXTENDED
000077                MAXCH    EQU     077Q          ;MAXIMUM CHARACTERISTIC WITH SIGN EXTENDED
                        ;
                        ;
                        ;*****************************************************
                        ;       //// DIVIDE SUBROUTINE
                        ;*****************************************************
                        ;
                        ;
004400   315 151 014 LDIV:    CALL    CSIGN           ;COMPUTE SIGN OF RESULT
004403   315 332 012          CALL    ZCHK            ;CHECK IF DIVIDEND = ZERO
004406   302 022 011          JNZ     DTST2           ;IF DIVIDEND .NE. 0 CHECK DIVISOR
004411   315 342 012          CALL    BCHK            ;CHECK FOR ZERO/ZERO
004414   312 250 013          JZ      INDFC           ;ZERO/ZERO = INDEFINITE
004417   303 257 013          JMP     WZERC           ;ZERO/NONZERO = ZERO
004422   315 342 012 DTST2:   CALL    BCHK            ;COME HERE IF DIVIDEND .NE. 0
004425   312 133 014          JZ      OFLWC           ;NONZERO/ZERO = OVERFLOW
                                                     ;IF WE GET HERE, THINGS LOOK OKAY
004430   135                  MOV     E,L             ;SAVE BASE IN E
004431   151                  MOV L,C                 ;BASE 6 TO L
004432   315 035 013          CALL DCLR               ;CLEAR QUOTIENT MANTISSA SLOT
004435   153                  MOV L,E                 ;RESTORE BASE IN L
004436   315 020 014          CALL ENT1               ;DO FIRST CYCLE
004441   151                  MOV L,C                 ;BASE  6 TO L
004442   315 351 012          CALL DLST               ;MOVE QUOTIENT OVER ONE PLACE
004445   026 027              MVI D,23                ;NUMBER OF ITERATIONS TO D
004447   153           REP3:  MOV L,E
004450   315 012 014          CALL ENT2
004453   025                  DCR D                   ;DEC D
004454   312 073 011          JZ  GOON
004457   175                  MOV A,L
004460   151                  MOV L,C                 ;BASE 6 TO L
004461   117                  MOV C,A
004462   315 351 012          CALL DLST               ;MOVE QUOTIENT MANT OVER
004465   175                  MOV A,L                 ;CPTR TO A
```

```
004466  131                     MOV E,C             ;LPTR TO E
004467  117                     MOV C,A             ;CPTR TO C
004470  303 047 011             JMP REP3
                        ;
004473  315 341 013 GOON:       CALL AORS             ;CHECK IF RESULT IS NORMALIZED
004476  372 115 011             JM  CRIN
004501  175                     MOV A,L             ;LPTR TO A
004502  151                     MOV L,C             ;CPTR TO L
004503  117                     MOV C,A             ;LPTR TO C
004504  315 351 012             CALL DLST           ;SHIFT QUOTIENT LEFT
004507  115                     MOV C,L
004510  153                     MOV L,E
004511  315 071 014             CALL    LDCP          ;COMPUTE THE CHARACTERISTIC OF RESULT
004514  311                     RET
                        ;
004515  315 114 013 CRIN:       CALL    CFCHE         ;GET A=CHAR(H,L), E=CHAR(H,B)
004520  223                     SUB     E             ;NEW CHAR = CHAR(DIVIDEND) - CHAR(DVISIOR)
004521  376 177                 CPI     177Q          ;CHECK MAX POSITIVE NUMBER
004523  312 133 014             JZ      OFLWC         ;JUMP ON OVERFLOW
004526  306 001                 ADI     1             ;ADD 1 SINCE WE DID NOT LEFTSHIFT
004530  315 104 014             CALL    CCHK          ;CHECK AND STORE CHARACTERISTIC
004533  311                     RET                   ;RETURN
                        ;
                        ;
                        ;
                        ;*****************************************************
                        ;       //// ADDITION SUBROUTINE
                        ;*****************************************************
                        ;
                        ;
004534  257             LADD:  XRA A                 ;/***SET UP TO ADD
004535  303 142 011             JMP LADS              ;/NOW DO IT
                        ;
                        ;
                        ;*****************************************************
                        ;       //// SUBTRACTION SUBROUTINE
                        ;*****************************************************
                        ;
                        ;
004540  076 200          LSUB:  MVI A,200Q            ;/****SET UP TO SUBTRACT
                        ;                       SUBROUTINE LADS
                        ;                       FLOATING POINT ADD OR SUB
                        ;                       A 128 ON ENTRY SUB
                        ;                       A 0 ON ENTRY ADD
                        ;                       F-S F,FIRST OPER DESTROYED
                        ;                       BASE  11 USED FOR SCRATCH
004542  315 357 013      LADS:  CALL ACPR             ;SAVE ENTRY PNT AT BASE  6
004545  315 342 012             CALL    BCHK          ;CHECK ADDEND/SUBTRAHEND = ZERO
004550  310                     RZ                    ;IF SO, RESULT=ARG SO RETURN
                                                      ;THIS WILL PREVENT UNDERFLOW INDICATION ON
                                                      ;ZERO + OR - ZERO
```

```
004551  315 133 013          CALL CCMP
004554  312 234 011          JZ  EQ02          ;IF EQUAL, GO ON
004557  127                  MOV D,A           ;SAVE LPTR CHAR IN D
004560  332 177 011          JC  LLTB
004563  223                  SUB E             ;L.GT.B IF HERE
004564  346 177              ANI 127
004566  127                  MOV D,A           ;DIFFERENCE TO D
004567  135                  MOV E,L           ;SAVE BASE IN E
004570  151                  MOV L,C           ;C PTR TO L
004571  054                  INR L             ;C PTR 1 TO L
004572  163                  MOV M,E           ;SAVE BASE IN C PTR 1
004573  150                  MOV L,B           ;B PTR TO L
004574  303 204 011          JMP NCHK
004577  173          LLTB:   MOV A,E           ;L.LT.B IF HERE,BPTR TO A
004600  222                  SUB D             ;SUBTRACT LPTR CHAR FROM BPTR CHAR
004601  346 177              ANI 127
004603  127                  MOV D,A           ;DIFFERENCE TO D
004604  076 030      NCHK:   MVI A,24
004606  272                  CMP D
004607  322 214 011          JNC SH10
004612  026 030              MVI D,24
004614  267          SH10:   ORA A
004615  315 370 012          CALL DRST
004620  025                  DCR D
004621  302 214 011          JNZ SH10
004624  175          EQUL:   MOV A,L
004625  270                  CMP B
004626  302 234 011          JNZ EQ02          ;F.GT.S IF L.NE.B
004631  151                  MOV L,C           ;C PTR TO L
004632  054                  INR L             ;C PTR 1 TO L
004633  156                  MOV L,M           ;RESTORE L
004634  315 002 012  EQ02:   CALL LASD          ;CHECK WHAT TO
004637  315 357 013          CALL ACPR         ;SAVE ANSWER
004642  376 002              CPI 2             ;TEST FOR ZERO ANSWER
004644  302 252 011          JNZ NOT0
004647  303 215 013          JMP     WZER        ;WRITE FLOATING ZERO AND RETURN
                     ;
004652  026 001      NOT0:   MVI D,1           ;WILL TEST FOR SUB
004654  242                  ANA D
004655  312 326 011          JZ  ADDZ          ;LSB 1 INPLIES SUB
004660  315 347 013          CALL TSTR         ;CHECK NORMAL/REVERSE
004663  312 271 011          JZ  SUBZ          ;IF NORMAL,GO SUBZ
004666  175                  MOV A,L           ;OTHERWISE REVERSE
004667  150                  MOV L,B           ;ROLES
004670  107                  MOV B,A           ;OF L AND B
                     ;
004671  315 046 013 SUBZ:     CALL    DSUB        ;SUBTRACT SMALLER FROM BIGGER
004674  315 357 011          CALL    MANT        ;SET UP SIGN OF RESULT
004677  315 347 013          CALL    TSTR        ;SEE IF WE NEED TO INTERCHANGE
                                                 ;BPTR AND LPTR
004702  312 255 012          JZ      NORM        ;NO INTERCHANGE NECESSARY, SO NORMALIZE
```

```
                                                    ;AND RETURN
004705   175                    MOV A,L             ;INTERCHANGE
004706   150                    MOV L,B             ;L
004707   107                    MOV B,A             ;AND B
004710   171                    MOV A,C             ;CPTR  TO A
004711   110                    MOV C,B             ;BPTR TO C
004712   135                    MOV E,L             ;LPTR TO E
004713   107                    MOV B,A             ;CPTR TO B
004714   315 044 014            CALL LXFR           ;MOVE BPTR> TO  LPTR>
004717   170                    MOV A,B
004720   101                    MOV B,C
004721   117                    MOV C,A
004722   153                    MOV L,E
004723   303 255 012            JMP     NORM        ;NORMALIZE RESULT AND RETURN
                        ;
                        ;   COPY THE LARGER CHARACTERISTIC TO THE RESULT
                        ;
004726   315 133 013 ADDZ:     CALL    CCMP        ;COMPARE THE CHARACTERISTICS
004731   322 337 011            JNC     ADD2        ;IF CHAR(H,L) .GE. CHAR(H,B) CONTINUE
004734   315 215 014            CALL    BCTL        ;IF CHAR(H,L) .LT. CHAR(H,B) THE COPY
                                                    ;CHAR(H,B) TO CHAR(H,L)
004737   315 357 011 ADD2:      CALL    MANT        ;COMPUTE SIGN OF RESULT
004742   315 006 013            CALL    DADD        ;ADD MANTISSAS
004745   322 322 013             JNC    SCCFG            ;IF THER IS NO OVFLW - DONE
004750   315 370 012            CALL    DRST        ;IF OVERFLOW SHIFT RIGHT
004753   315 266 013            CALL    INCR        ;AND INCREMENT CHARACTERISTIC
004756   311                    RET                 ;ALL DONE, SO RETURN
                        ;
                        ;   THIS ROUTINE STORES THE MANTISSA SIGN IN THE RESULT
                        ;   THE SIGN HAS PREVIOUSLY BEEN COMPUTED BY LASD.
                        ;
004757   135             MANT:  MOV E,L             ;SAVE L PTR
004760   151                    MOV L,C             ;C PTR TO L
004761   176                    MOV A,M             ;LOAD INDEX WORD
004762   346 200                ANI 128             ;SCARF SIGN
004764   153                    MOV L,E             ;RESTORE L PTR
004765   054                    INR L               ;L PTR 2
004766   054                    INR L
004767   054                    INR L               ;TO L
004770   137                    MOV E,A             ;SAVE SIGN IN E
004771   176                    MOV A,M
004772   346 177                ANI 127             ;SCARF CHAR
004774   203                    ADD E               ;ADD SIGN
004775   167                    MOV M,A             ;STORE IT
004776   055                    DCR L               ;RESTORE
004777   055                    DCR L
005000   055                    DCR L               ;L PTR
005001   311                    RET
                        ;
                        ;
                         ;                      SUBROUTINE LASD
```

```
                        ;                               UTILITY ROUTINE FOR LADS
                        ;                               CALCULATES TRUE OPER AND SGN
                        ;                               RETURNS ANSWER IN
005002  315 171 014  LASD:   CALL MSFH                    ;FETCH MANT SIGNS, F IN A,D
005005  273                  CMP E                        ;COMPARE SIGNS
005006  332 064 012          JC  ABCH                     ;F ,S- MEANS GO TO A BRANCH
005011  302 075 012          JNZ BBCH                     ;F- S  MEANS GO TO B BRANCH
005014  203                  ADD E                        ;SAME SIGN IF HERE, ADD SIGNS
005015  332 042 012          JC  BMIN                     ;IF BOTH MINUS, WILL OVERFLOW
005020  315 341 013          CALL AORS                    ;BOTH POS IF HERE
005023  362 106 012          JP  L000                     ;IF AN ADD, LOAD 0
005026  315 364 013  COM1:   CALL DCMP                    ;COMPARE F WITH S
005031  332 124 012          JC  L131                     ;S.GT.F,SO LOAD 131
005034  302 110 012          JNZ L001                     ;F.GT.S,SO LOAD 1
005037  076 002        L002: MVI A,2                      ;ERROR CONDITION, ZERO ANSWER
005041  311                  RET
005042  315 341 013  BMIN:   CALL AORS                    ;CHECK FOR ADD OR SUB
005045  362 116 012          JP  L128                     ;ADD, SO LOAD 128
005050  315 364 013  COM2:   CALL DCMP                    ;COMPARE F WITH S
005053  332 113 012          JC  L003                     ;S.GT.F,SO LOAD 3
005056  302 121 012          JNZ L129                     ;FGT.S.SO LOAD 129
005061  303 037 012          JMP L002                     ;ERROR
005064  315 341 013  ABCH:   CALL AORS                    ;FT,S- SO TEST FOR A/S
005067  372 106 012          JM  L000                     ;SUBTRACT, SO LOAD 0
005072  303 026 012          JMP COM1                     ;ADD, SO GO TO DCMP
005075  315 341 013  BBCH:   CALL AORS                    ;F-,S ,SO TEST FOR A/S
005100  372 116 012          JM  L128                     ;SUB
005103  303 050 012          JMP COM2                     ;ADD
005106  257          L000:   XRA A
005107  311                  RET
005110  076 001        L001: MVI A,1
005112  311                  RET
005113  076 003        L003: MVI A,3
005115  311                  RET
005116  076 200        L128: MVI A,128
005120  311                  RET
005121  076 201        L129: MVI A,129
005123  311                  RET
005124  076 203        L131: MVI A,131
005126  311                  RET
                        ;
                        ;                               SUBROUTINE LMCM
                        ;                               COMPARES THE MAGNITUDE OF
                        ;                               TWO FLOATING PNT NUMBERS
                        ;                               Z 1 IF  ,C 1 IF F.LT.S.
005127  315 133 013  LMCM:   CALL CCMP                    ;CHECK CHARS
005132  300                  RNZ                          ;RETURN IF NOT EQUAL
005133  315 364 013          CALL DCMP                    ;IF EQUAL, CHECK MANTS
005136  311                  RET
                        ;
                        ;
```

```
                        ;
                        ;*************************************************
                        ;       //// MULTIPLY SUBROUTINE
                        ;*************************************************
                        ;
                        ;                       SUBROUTINE LMUL
                        ;                       FLOATING POINT MULTIPLY
                        ;                       L PTR X B PTR TO C PTR
                        ;
005137  315 151 014 LMUL:   CALL    CSIGN       ;COMPUTE SIGN OF RESULT AND STORE IT
005142  315 332 012         CALL    ZCHK        ;CHECK FIRST OPERAND FOR ZERO
005145  312 257 013         JZ      WZERC       ;ZERO * ANYTHING = ZERO
005150  315 342 012         CALL    BCHK        ;CHECK SECOND OPERAND FOR ZERO
005153  312 257 013         JZ      WZERC       ;ANYTHING * ZERO = ZERO
005156  135                 MOV E,L             ;SAVE L PTR
005157  151                 MOV L,C             ;C PTR TO L
005160  315 035 013         CALL DCLR           ;CLR PRODUCT MANT LOCS
005163  153                 MOV L,E             ;L PTR TO L
005164  026 030             MVI D,24            ;LOAD NUMBER ITERATIONS
005166  315 370 012 KPGO:   CALL DRST           ;SHIFT L PTR RIGHT
005171  332 244 012         JC  MADD            ;WILL ADD B PTR IF C 1
005174  175                 MOV A,L             ;INTERCHANGE
005175  151                 MOV L,C             ;L AND
005176  117                 MOV C,A             ;C PTRS
005177  315 370 012 INTR:   CALL DRST           ;SHIFT PRODUCT OVER
005202  175                 MOV A,L             ;INTERCHANGE
005203  151                 MOV L,C             ;L AND C PTRS BACK TO
005204  117                 MOV C,A             ;ORIGINAL>
005205  025                 DCR D
005206  302 166 012         JNZ KPGO            ;MORE CYCLES IF Z 0
005211  315 341 013         CALL    AORS        ;TEST IF RESULT IS NORMALIZED
005214  372 100 014         JM      LMCP        ;IF NORMALIZED GO COMPUTE CHAR
005217  135                 MOV     E,L         ;SAVE LPTR IN E
005220  151                 MOV     L,C         ;SET L=CPTR
005221  315 351 012         CALL    DLST        ;LEFT SHIFT RESULT TO NORMALIZE
005224  153                 MOV     L,E         ;RESTORE LPTR
005225  315 114 013         CALL    CFCHE       ;OTHERWISE SET A=CHAR(H,L), E=CHAR(H,B)
005230  203                 ADD     E           ;CHAR(RESULT) = CHAR(H,L) + CHAR(H,B)
005231  376 200             CPI     200Q        ;CHECK FOR SMALLEST NEGATIVE NUMBER
005233  312 142 014         JZ      UFLWC       ;IF SO THEN UNDERFLOW
005236  326 001             SUI     1           ;SUBTRACT 1 TO COMPENSATE FOR NORMALIZE
005240  315 104 014         CALL    CCHK        ;CHECK CHARACTERISTIC AND STORE IT
005243  311                 RET                 ;RETURN
                        ;
005244  175           MADD: MOV A,L             ;INTERCHANGE
005245  151                 MOV L,C             ;L AND
005246  117                 MOV C,A             ;C PTRS
005247  315 006 013         CALL DADD           ;ACCUMULATE PRODUCT
005252  303 177 012         JMP INTR
                        ;
                        ;   SUBROUTINE NORM
```

```
                        ;
                        ;       THIS SUBROUTINE WILL NORMALIZE A FLOATING POINT
                        ;       NUMBER, PRESERVING ITS ORIGINAL SIGN.
                        ;       WE CHECK FOR UNDERFLOW AND SET THE CONDITION
                        ;       FLAG APPROPRIATELY.  (SEE ERROR RETURNS).
                        ;       THER IS AN ENTRY POINT TO FLOAT A SIGNED INTEGER
                        ;       (FLOAT) AND AN ENTRY POINT TO FLOAT AN UNSIGNED
                        ;       INTEGER.
                        ;
                        ;   ENTRY POINTS:
                        ;
                        ;       NORM  - NORMALIZE FLOATING PT NUMBER AT (H,L)
                        ;       FLOAT - FLOAT TRIPLE PRECISION INTEGER AT (H,L)
                        ;               PRESERVING SIGN BIT IN (H,L)+3
                        ;       DFXL  - FLOAT UNSIGNED (POSITIVE) TRIPLE PRECISION
                        ;               AT (H,L)
                        ;
                        ;REGISTERS ON EXIT:
                        ;
                        ;       A = CONDITION FLAG (SEE ERROR RETURNS)
                        ;       D,E = GARBAGE
                        ;       B,C,H,L = SAME AS ON ENTRY
                        ;
005255  135             NORM:   MOV     E,L             ;SAVE L IN E
005256  315 101 013     NORM1:  CALL    GCHAR           ;GET CHAR(H,L) IN A WITH SIGN EXTENDED
005261  127                     MOV     D,A             ;SAVE CHAR IN D
005262  153             FXL1:   MOV     L,E             ;RESTORE L
005263  315 332 012     FXL2:   CALL    ZMCHK           ;CHECK FOR ZERO MANTISSA
005266  312 215 013             JZ      WZER            ;IF ZERO MANTISSA THEN ZERO RESULT
005271  176             REP6:   MOV     A,M             ;GET MOST SIGNIFICANT BYTE OF
                                                        ;MANTISSA
005272  267                     ORA     A               ;SET FLAGS
005273  372 313 012             JM      SCHAR           ;IF MOST SIGNFICANT BIT = 1 THEN
                                                        ;NUMBER IS NORMALIZED AND WE GO TO
                                                        ;STORE THE CHARACTERISTIC
005276  172                     MOV     A,D             ;OTHERWISE CHECK FOR UNDERFLOW
005277  376 300                 CPI     MINCH           ;COMPARE WITH MINIMUM CHAR
005301  312 143 013             JZ      WUND            ;IF EQUAL THEN UNDERFLOW
005304  315 351 012             CALL    DLST            ;SHIFT MANTISSA LEFT
005307  025                     DCR     D               ;DECREMENT CHARACTERSTIC
005310  303 271 012             JMP     REP6            ;LOOP AN TEST NEXT BIT
005313  303 303 013 SCHAR:   JMP     INCR3           ;STORE THE CHARACTERISTIC USING
                                                        ;THE SAME CODE AS THE INCREMENT
                        ;
005316  135             DFXL:   MOV     E,L             ;ENTER HERE TO FLOAT UNSIGNED
                                                        ;INTEGER
                                                        ;FIRT SAVE L IN E
005317  054                     INR     L               ;MAKE (H,L) POINT TO CHAR
005320  054                     INR     L               ;MAKE (H,L) POINT TO CHAR
005321  054                     INR     L               ;MAKE (H,L) POINT TO CHAR
005322  257                     XRA     A               ;ZERO ACCUMULATOR
```

```
8080 MACRO ASSEMBLER, VER 2.2  ERRORS = 0 PAGE 8

  005323   167                    MOV     M,A          ;STORE A PLUS (+) SIGN
  005324   153                    MOV     L,E          ;RESTORE L
  005325   026 030       FLOAT:   MVI     D,24         ;ENTER HERE TO FLOAT INTEGER
                                                       ;PRESERVING ORIGINAL SIGN IN (H,L)+3
                                                       ;SET UP CHARACTERISTIC
  005327   303 263 012            JMP     FXL2         ;GO FLOAT THE NUMBER
                         ;
                         ;
                         ;
                         ;    SUBROUTINE ZCHK
                         ;
                         ;        THIS ROUTINE SETS THE ZERO FLAG IF IT DETECTS
                         ;        A FLOATING ZERO AT (H,L).
                         ;
                         ;    SUBROUTINE ZMCHK
                         ;
                         ;        THIS ROUTINE SETS THE ZERO FLAG IF IT DETECTS A
                         ;        ZERO MANTISSA AT (H,L)
                         ;
  005332                 ZCHK:
  005332   054           ZMCHK:    INR     L            ;SET L TO POINT LAST BYTE OF MANTISSA
  005333   054                     INR     L            ;SET L TO POINT TO LAST BYTE OF MANTISSA
  005334   176                     MOV     A,M          ;LOAD LEAST SIGNIFICANT BYTE
  005335   055                     DCR     L            ;L POINTS TO MIDDLE BYTE
  005336   266                     ORA     M            ;OR WITH LEAST SIGNFICANT BYTE
  005337   055                     DCR     L            ;L POINTS TO MOST SIGNFICANT BYTE
                                                       ;OF MANTISSA (ORIGINAL VALUE)
  005340   266                     ORA     M            ;OR IN MOST SIGNFICANT BYTE
  005341   311                     RET                  ;RETURNS WITH ZERO FLAG SET APPROPRIATELY
                         ;
                         ;  SUBROUTINE BCHK
                         ;
                         ;        THIS ROUTINE CHECKS (H,B) FOR FLOATING PT ZERO
                         ;
  005342   135           BCHK:     MOV     E,L          ;SAVE LPTR IN E
  005343   150                     MOV     L,B          ;SET L=BPTR
  005344   315 332 012             CALL    ZCHK         ;CHECK FOR ZERO
  005347   153                     MOV     L,E          ;RESTORE L=LPTR
  005350   311                     RET                  ;RETURN
                         ;
                         ;
                         ;                                SUBROUTINE DLST
                         ;                                SHIFTS DBL WORD ONE PLACE LF
  005351   054           DLST:     INR L
  005352   054                     INR L                  ;/** TP
  005353   176                     MOV A,M                ;LOAD IT
  005354   267                     ORA A                  ;KILL CARRY
  005355   027                     RAL                    ;SHIFT IT LEFT
  005356   167                     MOV M,A                ;STORE IT
  005357   055                     DCR L

8080 MACRO ASSEMBLER, VER 2.2  ERRORS = 0 PAGE 9

  005360   176                     MOV A,M                ;LOAD IT
```

```
005361  027             RAL             ;SHIFT IT LEFT
                        ;               IF CARRY SET BY FIRST SHIFT
                        ;               IT WILL BE IN LSB OF SECOND
005362  167             MOV M,A
005363  055             DCR L           ;/***TP EXTENSION
005364  176             MOV A,M
005365  027             RAL
005366  167             MOV M,A         ;/***ALL DONE TP
005367  311             RET
                        ;               SUBROUTINE DRST
                        ;               SHIFTS DOUBLE WORD ONE PLACE
                        ;               TO THE RIGHT
                        ;               DOES NOT AFFECT D
005370  135     DRST:   MOV E,L         ;/***TP MODIFIED RIGHT SHIFT TP
005371  176             MOV A,M         ;LOAD FIRST WORD
005372  037             RAR             ;ROTATE IT RIGHT
005373  167             MOV M,A         ;STORE IT
005374  054             INR L           ;/*** TP
005375  176             MOV A,M         ;LOAD SECOND WORD
005376  037             RAR             ;SHIFT IT RIGHT
005377  167             MOV M,A         ;STORE IT
005400  054             INR L           ;/*** TP EXTENSION
005401  176             MOV A,M
005402  037             RAR
005403  167             MOV M,A
005404  153             MOV L,E         ;/***TP - ALL DONE TP
005405  311             RET
                        ;               SUBROUTINE DADD
                        ;               ADDS TWO DOUBLE PRECISION
                        ;               WORDS, C 1 IF THERE IS OVRFLW
005406  135     DADD:   MOV E,L         ;SAVE BASE IN E
005407  150             MOV L,B         ;BASE  3 TO L
005410  054             INR L           ;BASE  4 TO L
005411  054             INR L           ;/***TP
005412  176             MOV A,M         ;LOAD S MANTB
005413  153             MOV L,E         ;BASE TO L
005414  054             INR L           ;BASE  1 TO L
005415  054             INR L           ;/***TP
005416  206             ADD M           ;ADD TWO MANTB S
005417  167             MOV M,A         ;STORE ANSWER
005420  150             MOV L,B         ;/***TP EXTENSION
005421  054             INR L
005422  176             MOV A,M
005423  153             MOV L,E
005424  054             INR L
005425  216             ADC M
005426  167             MOV M,A         ;/***TP - ALL DONE
005427  150             MOV L,B         ;BASE  3 TO L
005430  176             MOV A,M         ;MANTA OF S TO A
005431  153             MOV L,E         ;BASE TO L
```

```
005432  216                     ADC M               ;ADD WITH CARRY
005433  167                     MOV M,A             ;STORE ANSWER
005434  311                     RET
                        ;                           SUBROUTINE DCLR
                        ;                           CLEARS TWO SUCCESSIVE
                        ;                           LOCATIONS OF MEMORY
005435  257             DCLR:   XRA A
005436  167                     MOV M,A
005437  054                     INR L
005440  167                     MOV M,A
005441  054                     INR L               ;/***TP EXTENSION
005442  167                     MOV M,A             ;/***TP ZERO 3
005443  055                     DCR L               ;/***TP - ALL DONE
005444  055                     DCR L
005445  311                     RET
                        ;                           /*****ALL NEW DSUB - SHORTER***
                        ;                           SUBROUTINE DSUB
                        ;                           DOUBLE PRECISION SUBTRACT
005446  135             DSUB:   MOV E,L             ;SAVE BASE IN E
005447  054                     INR L               ;/***TP EXTENSION
005450  054                     INR L               ;/START WITH LOWS
005451  176                     MOV A,M             ;/GET ARG
005452  150                     MOV L,B             ;/NOW SET UP TO SUB
005453  054                     INR L
005454  054                     INR L
005455  226                     SUB M               ;/NOW DO IT
005456  153                     MOV L,E             ;/NOW MUST PUT IT BACK
005457  054                     INR L
005460  054                     INR L
005461  167                     MOV M,A             ;/PUT BACK
005462  055                     DCR L               ;/***TP - ALL DONE
005463  176                     MOV A,M             ;/GET LOW OF LOP
005464  150                     MOV L,B             ;/SET TO BOP
005465  054                     INR L               ;/SET TO BOP LOW
005466  236                     SBB M               ;/GET DIFF. OF LOWS
005467  153                     MOV L,E             ;/SAVE IN LOP LOW
005470  054                     INR L               ;/TO LOP LOW
005471  167                     MOV M,A             ;/INTO RAM
005472  055                     DCR L               ;/BACK UP TO LOP HIGH
005473  176                     MOV A,M             ;/GET LOP HIGH
005474  150                     MOV L,B             ;/SET TO BOP HIGH
005475  236                     SBB M               ;/SUB. WITH CARRY
005476  153                     MOV L,E             ;/SAVE IN LOP HIGH
005477  167                     MOV M,A             ;/INTO RAM
005500  311                     RET                 ;/ALL DONE - MUCH SHORTER
                        ;
                        ;   SUBROUTINE GCHAR
                        ;
                        ;       THIS SUBROUTINE RETURNS THE CHARACTERISTIC OF
                        ;       THE FLOATING POINT NUMBER POINTED TO BY (H,L)
                        ;       IN THE A REGISTER WITH ITS SIGN EXTENDED INTO THE
```

```
                        ;       LEFTMOST BIT.
                        ;
                        ;   REGISTERS ON EXIT:
                        ;
                        ;       A = CHARACTERISTIC OF (H,L) WITH SIGN EXTENDED
                        ;       L = (ORIGINAL L) + 3
                        ;       B,C,D,E,H = SAME AS ON ENTRY
                        ;
005501  054             GCHAR:  INR     L               ;MAKE (H,L) POINT TO CHAR
005502  054                     INR     L               ;MAKE (H,L) POINT TO CHAR
005503  054                     INR     L               ;MAKE (H,L) POINT TO CHAR
005504  176                     MOV     A,M             ;SET A=CHAR + MANTISSA SIGN
005505  346 177                 ANI     177Q            ;GET RID OF MANTISSA SIGN BIT
005507  306 100                 ADI     100Q            ;PROPAGATE CHAR SIGN INTO LEFTMOST BIT
005511  356 100                 XRI     100Q            ;RESTORE ORIGINAL CHAR SIGN BIT
005513  311                     RET                     ;RETURN WITH (H,L) POINTING TO THE
                                                        ;CHAR = ORIGINAL (H,L)+3
                                                        ;SOMEONE ELSE WILL CLEAN UP
                        ;
                        ;
                        ;   SUBROUTINE CFCHE
                        ;
                        ;       THIS SUBROUTINE RETURNS THE CHARACTERISTICS OF THE
                        ;       FLOATING POINT NUMBERS POINTED TO BY (H,L) AND
                        ;       (H,B) IN THE A AND E REGISTERS RESPECTIVELY,
                        ;       WITH THEIR SIGNS EXTENDED INTO THE LEFTMOST BIT.
                        ;
                        ;   REGISTERS ON EXIT:
                        ;
                        ;       A = CHARACTERISTIC OF (H,L) WITH SIGN EXTENDED
                        ;       E = CHARACTERISTIC OF (H,B) WITH SIGN EXTENDED
                        ;       B,C,H,L = SAME AS ON ENTRY
                        ;       D = A
                        ;
005514  135             CFCHE:  MOV     E,L             ;SAVE LPTR IN E
005515  150                     MOV     L,B             ;SET L = BPTR
005516  315 101 013             CALL    GCHAR           ;GET CHAR(H,B) WITH SIGN EXTENDED IN A
005521  153                     MOV     L,E             ;RESTORE L = LPTR
005522  137                     MOV     E,A             ;SET E=CHAR(H,B) WITH SIGN EXTENDED
005523  315 101 013             CALL    GCHAR           ;SET A=CHAR(H,L) WITH SIGN EXTENDED
005526  055                     DCR     L               ;RESTORE L = LPTR
005527  055                     DCR     L               ;RESTORE L = LPTR
005530  055                     DCR     L               ;RESTORE L = LPTR
005531  127                     MOV     D,A             ;SET D=A=CHAR(H,L) WITH SIGN EXTENDED
005532  311                     RET
                        ;
                        ;
                        ;   SUBROUTINE CCMP
                        ;
                        ;       THIS SUBROUTINE COMPARES THE CHARACTERISTICS OF
                        ;       FLOATING POINT NUMBERS POINTED TO BY (H,L) AND (H,B).
```

```
                        ;       THE ZERO FLIP-FLOP IS SET IF CHAR(H,L) EQUALS
                        ;       CHAR(H,B).  IF CHAR(H,L) IS LESS THAN CHAR(H,B) THEN
                        ;       THE CARRY BIT WILL BE SET.
                        ;
                        ;   REGISTERS ON EXIT:
                        ;
                        ;       A = CHARACTERISTIC OF (H,L) WITH SIGN EXTENDED
                        ;       E = CHARACTERISTIC OF (H,B) WITH SIGN EXTENDED
                        ;       D = A
                        ;       B,C,H,L = SAME AS ON ENTRY
                        ;
005533   315 114 013 CCMP:    CALL    CFCHE           ;FETCH CHARACTERTISTICS WITH SIGN EXTENDED
                                                     ;INTO A (CHAR(H,L)) AND E (CHAR(H,B)) REGIS
005536   127                  MOV     D,A             ;SAVE CHAR (H,L)
005537   223                  SUB     E               ;SUBTRACT E (CHAR(H,B))
005540   027                  RAL                     ;ROTATE SIGN BIT INTO CARRY BIT
005541   172                  MOV     A,D             ;RESTORE A=CHAR(H,L)
005542   311                  RET                     ;RETURN
                        ;
                        ;   ERROR RETURNS
                        ;
                        ;       THE FOLLOWING CODE IS USED TO RETURN VARIOUS
                        ;       ERROR CONDITIONS.  IN EACH CASE A FLOATING POINT
                        ;       NUMBER IS STORED IN  THE 4 WORDS POINTED TO BY (H,L)
                        ;       AND A FLAG IS STORED IN THE ACCUMULATOR.
                        ;
                        ;       CONDITION    FLAG    RESULT (+)          RESULT (-)
                        ;
                        ;       UNDERFLOW     377    000 000 000 100     000 000 000 300
                        ;       OVERFLOW      177    377 377 377 077     377 377 377 277
                        ;       INDEFINITE    077    377 377 377 077     377 377 377 277
                        ;       NORMAL        000    XXX XXX XXX XXX     XXX XXX XXX XXX
                        ;       NORMAL ZERO   000    000 000 000 100     (ALWAYS RETURNS +0)
                        ;
                        ;   ENTRY POINTS:
                        ;
                        ;       WUND - WRITE UNDERFLOW
                        ;       WOVR - WRITE OVERFLOW
                        ;       WIND - WRITE INDEFINITE
                        ;       WZER - WRITE NORMAL ZERO
                        ;
1                       WFLT    MACRO   VMANT,VCHAR,VFLAG,LABEL  ;WRITE FLOATING NUMBER
1
1                               MVI     D,VCHAR         ;LOAD CHARACTERISTIC INTO D REGISTER
1                               CALL    WCHAR           ;WRITE CHARACTERISTIC
1                       LABEL:: MVI     A,VMANT         ;LOAD MANTISSA VALUE
1                                                       ;WE ASSUME HERE THAT ALL BYTES OF MANTISSA
1                                                       ;ARE THE SAME
1                               CALL    WMANT           ;WRITE THE MANTISSA
1                               MVI     A,VFLAG         ;SET ACCUMULATOR TO FLAG
1                               ORA     A               ;SET FLAGS PROPERLY
```

```
        1                       RET                 ;RETURN (WMANT RESTORED (H,L))
                                ENDM
                        ;
005543  1               +WUND:  WFLT    0,'00Q,377Q,UFLWI  ;WRITE UNDERFLOW
        1               +
005543  1 026 100       +       MVI     D,00040H    ;LOAD CHARACTERISTIC INTO D REGISTER
005545  1 315 237 013+         CALL    WCHAR       ;WRITE CHARACTERISTIC
005550  1 076 000       +UFLWI::MVI     A,00000H    ;LOAD MANTISSA VALUE
        1               +                           ;WE ASSUME HERE THAT ALL BYTES OF MANTISSA
        1               +                           ;ARE THE SAME
005552  1 315 230 013+         CALL    WMANT       ;WRITE THE MANTISSA
005555  1 076 377       +       MVI     A,000FFH    ;SET ACCUMULATOR TO FLAG
005557  1 267           +       ORA     A           ;SET FLAGS PROPERLY
005560  1 311           +       RET                 ;RETURN (WMANT RESTORED (H,L))
005561  1               +WOVR:  WFLT    377Q,77Q,177Q,OFLWI  ;WRITE OVERFLOW
        1               +
005561  1 026 077       +       MVI     D,0003FH    ;LOAD CHARACTERISTIC INTO D REGISTER
005563  1 315 237 013+         CALL    WCHAR       ;WRITE CHARACTERISTIC
005566  1 076 377       +OFLWI::MVI     A,000FFH    ;LOAD MANTISSA VALUE
        1               +                           ;WE ASSUME HERE THAT ALL BYTES OF MANTISSA
        1               +                           ;ARE THE SAME
005570  1 315 230 013+         CALL    WMANT       ;WRITE THE MANTISSA
005573  1 076 177       +       MVI     A,0007FH    ;SET ACCUMULATOR TO FLAG
005575  1 267           +       ORA     A           ;SET FLAGS PROPERLY
005576  1 311           +       RET                 ;RETURN (WMANT RESTORED (H,L))
005577  1               +WIND:  WFLT    377Q,77Q,77Q,INDFI  ;WRITE INDEFINITE
        1               +
005577  1 026 077       +       MVI     D,0003FH    ;LOAD CHARACTERISTIC INTO D REGISTER
005601  1 315 237 013+         CALL    WCHAR       ;WRITE CHARACTERISTIC
005604  1 076 377       +INDFI::MVI     A,000FFH    ;LOAD MANTISSA VALUE
        1               +                           ;WE ASSUME HERE THAT ALL BYTES OF MANTISSA
        1               +                           ;ARE THE SAME
005606  1 315 230 013+         CALL    WMANT       ;WRITE THE MANTISSA
005611  1 076 077       +       MVI     A,0003FH    ;SET ACCUMULATOR TO FLAG
005613  1 267           +       ORA     A           ;SET FLAGS PROPERLY
005614  1 311           +       RET                 ;RETURN (WMANT RESTORED (H,L))
                        ;
005615  054             WZER:   INR     L           ;WRITE NORMAL ZERO
005616  054                     INR     L           ;
005617  054                     INR     L           ;
005620  066 100                 MVI     M,100Q      ;STORE CHARACTERISTIC FOR ZERO
005622  257                     XRA     A           ;ZERO ACCUMULATOR
005623  315 230 013             CALL    WMANT       ;STORE ZERO MANTISSA
005626  267                     ORA     A           ;SET FLAGS PROPERLY
005627  311                     RET                 ;RETURN
                        ;
                        ; ROUTINE TO WRITE MANTISSA FOR ERROR RETURNS
                        ;
005630  055             WMANT: DCR     L           ;POINT LEAST SIGNIFICANT BYTE
                                                    ;OF MANTISSA
005631  167                     MOV     M,A         ;STORE LSBYTE OF MANTISSA
```

```
005632  055                 DCR     L           ;POINT TO NEXT LEAST SIGNIFICANT BYTE
                                                ;OF MANTISSA
005633  167                 MOV     M,A         ;STORE NLSBYTE OF MANTISSA
005634  055                 DCR     L           ;POINT TO MOST SIGNIFICANT BYTE
                                                ;OF MANTISSA
005635  167                 MOV     M,A         ;STORE MSBYTE OF MANTISSA
005636  311                 RET                 ;RETURN (H,L) POINTS TO BEGINNING OF
                                                ;FLOATING POINT RESULT
                    ;
                    ; ROUTINE TO WRITE CHARACTERTIC FOR ERROR RETURNS
                    ; NOTE:  WE PRESERVE ORIGINAL MANTISSA SIGN
                    ; ON ENTRY D CONTAINS NEW CHARACTERTISTIC TO BE STORED.
                    ;
005637  054         WCHAR:  INR     L           ;SET (H,L) TO POINT TO CHARACTERISTIC
005640  054                 INR     L           ;PART OF ABOVE
005641  054                 INR     L           ;PART OF ABOVE
005642  176                 MOV     A,M         ;LOAD CHARACTERISTIC A
                                                ;AND MANTISSA SIGN
005643  346 200             ANI     200Q        ;JUST KEEP MANTISSA SIGN
005645  262                 ORA     D           ;OR IN NEW CHARACTERISTIC
005646  167                 MOV     M,A         ;STORE IT BACK
005647  311                 RET                 ;RETURN WITH (H,L) POINT TO CHARACTERISTIC
                                                ;OF RESULT
                                                ;SOMEONE ELSE WILL FIX UP (H,L)
                    ;
                    ;   SUBROUTINE INDFC
                    ;
                    ;       THIS ROUTINE WRITES A FLOATING INDEFINITE, SETS
                    ;       THIS WRITES WRITES A FLOATING POINT INDEFINITE
                    ;       AT (H,C), SETS THE CONDITION FLAG AND RETURNS
                    ;
                    ;
005650  135         INDFC:  MOV     E,L         ;SAVE LPTR IN E
005651  151                 MOV     L,C         ;SET L=CPTR SO (H,L)=ADDR OF RESULT
005652  315 177 013         CALL    WIND        ;WRITE INDEFINITE
005655  153                 MOV     L,E         ;RESTORE L=LPTR
005656  311                 RET                 ;RETURN
                    ;
                    ;
                    ;   SUBROUTINE WZERC
                    ;
                    ;       THIS ROUTINE WRITES A NORMAL FLAOTING POINT ZERO
                    ;       AT (H,C), SETS THE CONDITION FLAG AND RETURNS
                    ;
005657  135         WZERC:  MOV     E,L         ;SAVE LPTR IN E
005660  151                 MOV     L,C         ;SETL=CPTR SO (H,L)=ADDR OF RESULT
005661  315 215 013         CALL    WZER        ;WRITE NORMAL ZERO
005664  153                 MOV     L,E         ;RESTORE L=LPTR
005665  311                 RET                 ;RETURN
                    ;
                    ;   SUBROUTINE INCR
```

```
                        ;
                        ;       THIS SUBROUTINE INCREMENTS THE CHARACTERISTIC
                        ;       OF THE FLOATING POINT NUMBER POINTED TO BY (H,L).
                        ;       WE TEST FOR OVERFLOW AND SET APPROPRIATE FLAG.
                        ;       (SEE ERRROR RETURNS).
                        ;
                        ;   REGISTERS ON EXIT:
                        ;
                        ;       A = CONDITION FLAG (SEE ERROR RETURNS)
                        ;       D = CLOBBERED
                        ;       B,C,H,L = SAME AS ON ENTRY
                        ;
005666  315 101 013 INCR:   CALL    GCHAR       ;GET CHAR WITH SIGN EXTENDED
005671  376 077             CPI     MAXCH       ;COMPARE WITH MAX CHAR PERMITTED
005673  312 166 013         JZ      OFLW1       ;INCREMENT WOULD CAUSE OVERFLOW
005676  127                 MOV D,A           ;/SAVE IT IN D
005677  024                 INR D             ;/INCREMENT IT
005700  303 306 013         JMP     INCR2       ;JUMP AROUND ALTERNATE ENTRY POINT
005703  054           INCR3:    INR     L           ;COME HERE TO STORE CHARACTERISTIC
005704  054                 INR     L           ;POINT (H,L) TO CHAR
005705  054                 INR     L           ;POINT (H,L) TO CHAR
005706  076 177       INCR2:    MVI A,177Q
005710  242                 ANA D             ;/KILL SIGN BIT
005711  127                 MOV D,A           ;/BACK TO D
005712  176                 MOV A,M           ;/NOW SIGN IT
005713  346 200             ANI 200Q          ;/GET MANTISSA SIGN
005715  262                 ORA D             ;/PUT TOGETHER
005716  167                 MOV M,A           ;/STORE IT BACK
005717  055                 DCR L             ;/NOW BACK TO BASE
005720  055                 DCR L             ;/***TP
005721  055                 DCR L
005722  257           SCCFG:     XRA     A                ;SET SUCCESS FLAG
005723  311                 RET
                        ;
                        ;   SUBROUTINE DECR
                        ;
                        ;       THIS SUBROUTINE DECREMENTS THE CHARACTERISTIC
                        ;       OF THE FLOATING POINT NUMBER POINTED TO BY (H,L).
                        ;       WE TEST FOR UNDERFLOW AND SET APPROPRIATE FLAG.
                        ;       (SEE ERRROR RETURNS).
                        ;
                        ;   REGISTERS ON EXIT:
                        ;
                        ;       A = CONDITION FLAG (SEE ERROR RETURNS)
                        ;       D = CLOBBERED
                        ;       B,C,H,L = SAME AS ON ENTRY
                        ;
005724  315 101 013 DECR:   CALL    GCHAR       ;GET CHAR WITH SIGN EXTENDED
005727  376 300             CPI     MINCH       ;COMPARE WITH MIN CHAR PERMITTED
005731  312 150 013         JZ      UFLW1       ;DECREMENT WOULD CAUSE UNDERFLOW
005734  127                 MOV     D,A         ;SAVE CHARACTERSTIC IN D
```

```
005735  025                     DCR     D               ;DECREMENT CHARACTERISTIC
005736  303 306 013             JMP     INCR2           ;GO STORE IT BACK
                        ;
                        ;                               SUBROUTINE AORS
                        ;                               RETURN S 1 IF BASE  6
                        ;                               HAS A 1 IN MSB
005741  135            AORS:   MOV E,L                   ;SAVE BASE
005742  151                    MOV L,C                   ;BASE  6 TO L
005743  176                    MOV A,M                   ;LOAD IT
005744  267                    ORA A                     ;SET FLAGS
005745  153                    MOV L,E                   ;RESTORE BASE
005746  311                    RET
                        ;                               SUBROUTINE TSTR
                        ;                               CHECKS C PTR TO SEE IF
                        ;                               NLSB 1
                        ;                               RETURNS Z 1 IF NOT
                        ;                               DESTROYS E,D
005747  135            TSTR:   MOV E,L                   ;SAVE BASE
005750  151                    MOV L,C                   ;C PTR TO L
005751  026 002                MVI D,2                   ;MASK TO D
005753  176                    MOV A,M                   ;LOAD VALUE
005754  153                    MOV L,E                   ;RESTORE BASE
005755  242                    ANA D                     ;AND VALUE WITH MASK
005756  311                    RET
                        ;                               SUBROUTINE ACPR
                        ;                               STORES A IN LOCATION OF CPTR
                        ;                               LPTR IN E
005757  135            ACPR:   MOV E,L                   ;SAVE LPTR
005760  151                    MOV L,C                   ;CPTR TO L
005761  167                    MOV M,A                   ;STORE A
005762  153                    MOV L,E                   ;RESTORE BASE
005763  311                    RET
                        ;                               SUBROUTINE DCMP
                        ;                               COMPARES TWO DOUBLE LENGTH
                        ;                               WORDS
005764  176            DCMP:   MOV A,M                   ;NUM MANTA TO A
005765  135                    MOV E,L                   ;SAVE BASE IN E
005766  150                    MOV L,B                   ;BASE 3 TO L
005767  276                    CMP M                     ;COMPARE WITH DEN MANTA
005770  153                    MOV L,E                   ;RETURN BASE TO L
005771  300                    RNZ                       ;RETURN IF NOT THE SAME
005772  054                    INR L                     ;L TO NUM MANTB
005773  176                    MOV A,M                   ;LOAD IT
005774  150                    MOV L,B                   ;DEN MANTB ADD TO L
005775  054                    INR L                     ;BASE  4 TO L
005776  276                    CMP M
005777  153                    MOV L,E
006000  300                    RNZ                       ;/***TP EXTENSION
006001  054                    INR L                     ;/NOW CHECK BYTE 3
006002  054                    INR L
006003  176                    MOV A,M                   ;/GET FOR COMPARE
```

```
006004   150                    MOV L,B
006005   054                    INR L
006006   054                    INR L              ;/BYTE 3 NOW
006007   276                    CMP M              ;/COMPARE
006010   153                    MOV L,E            ;/***TP - ALL DONE
006011   311                    RET
                        ;                         SUBROUTINE DIVC
                        ;                         PERFORMS ONE CYCLE OF DOUBLE
                        ;                         PRECISION FLOATING PT DIVIDE
                        ;                         ENTER AT ENT1 ON FIRST CYCLE
                        ;                         ENTER AT ENT2 ALL THEREAFTER
006012   315 351 012  ENT2:  CALL DLST            ;SHIFT MOVING DIVIDEND
006015   332 027 014         JC  OVER            ;IF CARRY 1,NUM.GT.D
006020   315 364 013  ENT1:  CALL DCMP            ;COMPARE NUM WITH DEN
006023   322 027 014         JNC OVER            ;IF CARRY NOT SET,NUM.GE.DEN
006026   311                 RET
006027   315 046 013  OVER:  CALL DSUB            ;CALL DOUBLE SUBTRACT
006032   135                 MOV E,L             ;SAVE BASE IN E
006033   151                 MOV L,C             ;BASE  6 TO L
006034   054                 INR L               ;BASE  7 TO L
006035   054                 INR L               ;/***TP
006036   176                 MOV A,M
006037   306 001             ADI 1               ;ADD 1
006041   167                 MOV M,A             ;PUT IT BACK
006042   153                 MOV L,E             ;RESTORE BASE TO L
006043   311                 RET
                        ;                         SUBROUTINE LXFR
                        ;                         MOVES CPTR TO EPTR
                        ;                         MOVES 3 WORDS IF ENTER AT LXFR
006044   026 004      LXFR:  MVI D,4             ;/MOVE 4 WORDS
006046   151          REP5:  MOV L,C             ;CPTR TO L
006047   176                 MOV A,M             ; CPTR> TO A
006050   153                 MOV L,E             ;EPTR TO L
006051   167                 MOV M,A
006052   014                 INR C               ;/INCREMENT C
006053   034                 INR E               ;/INCREMENT E TO NEXT
006054   025                 DCR D               ;/TEST FOR DONE
006055   302 046 014         JNZ REP5            ;/GO FOR FOR TILL D=0
006060   173                 MOV A,E             ;/NOW RESET C AND E
006061   326 004             SUI 4               ;/RESET BACK BY 4
006063   137                 MOV E,A             ;/PUT BACK IN E
006064   171                 MOV A,C             ;/NOW RESET C
006065   326 004             SUI 4               ;/BY 4
006067   117                 MOV C,A             ;/BACK TO C
006070   311                 RET                 ;/DONE
                        ;
                        ;   SUBROUTINE LDCP
                        ;
                        ;       THIS SUBROUTINE COMPUTES THE CHARACTERISTIC
                        ;       FOR THE FLOATING DIVIDE ROUTINE
                        ;
```

```
                        ;   REGISTERS ON EXIT:
                        ;
                        ;       A = CONDITION FLAG (SEE ERROR RETURNS)
                        ;       D,E = GARBAGE
                        ;       B,C,H,L = SAME AS ON ENTRY
                        ;
                        ;   REGISTERS ON ENTRY:
                        ;
                        ;       (H,B) = ADDRESS OFF DIVISOR
                        ;       (H,C) = ADDRESS OF QUOTIENT
                        ;       (H,L) = ADDRESS OF DIVIDEND
                        ;
006071  315 114 013 LDCP:    CALL    CFCHE       ;SET E=CHAR(H,B), A=CHAR(H,L)
006074  223                  SUB     E           ;SUBTRACT TO GET NEW CHARACTERISTIC
006075  303 104 014          JMP     CCHK        ;GO CHECK FOR OVER/UNDERFLOW
                                                 ;AND STORE CHARACTERTISTIC
                        ;
                        ;
                        ;   SUBROUTINE LMCP
                        ;
                        ;       THIS SUBROUTINE COMPUTES THE CHARACTERISTIC
                        ;       FOR THE FLOATING MULTIPLY ROUTINE.
                        ;
                        ;   REGISTERS ON EXIT:
                        ;
                        ;       A = CONDITION FLAG (SEE ERROR RETURNS)
                        ;       D,E = GARBAGE
                        ;       B,C,H,L = SAME AS ON ENTRY
                        ;
                        ;   REGISTERS ON ENTRY:
                        ;
                        ;       (H,B) = ADDRESS OFF MULTIPLICAND
                        ;       (H,C) = ADDRESS OF PRODUCT
                        ;       (H,L) = ADDRESS OF MULTIPLIER
                        ;
006100  315 114 013 LMCP:    CALL    CFCHE       ;SET E=CHAR(H,B), A=CHAR(H,L)
006103  203                  ADD     E           ;ADD TO GET NEW CHARACTERISTIC
                                                 ;NOW FALL INTO THE ROUTINE
                                                 ;WHICH CHECKS FOR OVER/UNDERFLOW
                                                 ;AND STORE CHARACTERTISTIC
                        ;
                        ;
                        ;   SBUROUTINE CCHK
                        ;
                        ;       THIS SUBROUTINE CHECKS A CHARACTERISTIC IN
                        ;       THE ACCUMULATOR FOR OVERFLOW OR UNDERFLOW.
                        ;       IT THEN STORES THE CHARACTERISTIC, PRESERVING
                        ;       THE PREVIOUSLY COMPUTED MANTISSA SIGN.
                        ;
                        ;  REGISTERS ON ENTRY:
                        ;
```

```
                        ;       (H,L) = ADDRESS OF ONE OPERAND
                        ;       (H,B) = ADDRESS OF OTHER OPERAND
                        ;       (H,C) = ADDRESS OF RESULT
                        ;       A     = NEW CHARACTERISTIC OF  RESULT
                        ;
                        ;   REGISTERS ON EXIT:
                        ;
                        ;       A = CONDITION FLAG (SEE ERROR RETURNS)
                        ;       D,E = GARBAGE
                        ;       B,C,H,L = SAME AS ON ENTRY
                        ;
006104                  CCHK:                           ;ENTER HERE TO CHECK CHARACTERISTIC
006104  376 100                 CPI     100Q            ;CHECK FOR 0 TO +63
006106  332 123 014             JC      STORC           ;JUMP IF OKAY
006111  376 200                 CPI     200Q            ;CHECK FOR +64 TO +127
006113  332 133 014             JC      OFLWC           ;JUMP IF OVERFLOW
006116  376 300                 CPI     300Q            ;CHECK FOR -128 TO -65
006120  332 142 014             JC      UFLWC           ;JUMP IF UNDERFLOW
006123  135             STORC:  MOV     E,L             ;SAVE L IN E
006124  151                     MOV     L,C             ;LET L POINT TO RESULT
006125  127                     MOV     D,A             ;SAVE CHARACTERISTIC IN D
006126  315 303 013             CALL    INCR3           ;STORE CHARACTERISTIC
006131  153                     MOV     L,E             ;RESTORE L
006132  311                     RET                     ;RETURN
                        ;
                        ;   SUBROUTINE OFLWC
                        ;
                        ;       THIS ROUTINE WRITES A FLOATING POINT OVERFLOW AT (H,C)
                        ;       SETS THE CONDITION FLAG, AND RETURNS.
                        ;
006133  135             OFLWC:  MOV     E,L             ;SAVE L IN E
006134  151                     MOV     L,C             ;SET L=CPTR, SO (H,L)=ADDR OF RESULT
006135  315 161 013             CALL    WOVR            ;WRITE OUT OVERFLOW
006140  153                     MOV     L,E             ;RESTORE L
006141  311                     RET                     ;RETURN
                        ;
                        ;   SUBROUTINE UFLWC
                        ;
                        ;       THIS ROUTINE WRITES A FLOATING POINT UNDERFLOW AT (H,C)
                        ;       SETS THE CONDITION FLAG, AND RETURNS.
                        ;
006142  135             UFLWC:  MOV     E,L             ;SAVE L IN E
006143  151                     MOV     L,C             ;SET L=CPTR, SO (H,L)=ADDR OF RESULT
006144  315 143 013             CALL    WUND            ;WRITE OUT UNDEFLOW
006147  153                     MOV     L,E             ;RESTORE L
006150  311                     RET                     ;RETURN
                        ;
                        ;
                        ;   SUBROUTINE CSIGN
                        ;
                        ;       THIS SUBROUTINE COMPUTES AND STORE THE MANTISSA
```

```
1
8080 MACRO ASSEMBLER, VER 2.2  ERRORS = 0 PAGE 20


                        ;       SIGN FOR THE FLOATING MULTIPLY AND DIVIDE ROUTINES
                        ;
                        ;   REGISTERS ON ENTRY:
                        ;
                        ;       (H,L) = ADDRESS OF ONE OPERAND
                        ;       (H,B) = ADDRESS OF OTHER OPERAND
                        ;       (H,C) = ADDRESS OF RESULT
                        ;
                        ;   REGISTERS ON EXIT:
                        ;
                        ;       A,D,E = GARBAGE
                        ;       B,C,H,L = SAME AS ON ENTRY
                        ;
                        ;
  006151   315 171 014 CSIGN:   CALL    MSFH          ;SET A=SIGN(H,L), E=SIGN(H,B)
  006154   253                  XRA     E             ;EXCLUSIVE OR SIGNS TO GET NEW SIGN
  006155   315 161 014          CALL    CSTR          ;STORE SIGN INTO RESULT
  006160   311                  RET                   ;RETURN
                        ;
                        ;
                         ;                          SUBROUTINE CSTR
                         ;                          STORES VALUE IN A IN
                         ;                          CPTR 2
                         ;                          PUTS LPTR IN E
  006161   135          CSTR:   MOV E,L                 ;SAVE LPTR IN E
  006162   151                  MOV L,C                 ;CPTR TO L
  006163   054                  INR L                   ;CPTR 2
  006164   054                  INR L                   ;TO L
  006165   054                  INR L                   ;/***TP
  006166   167                  MOV M,A                 ;STORE ANSWER
  006167   153                  MOV L,E                 ;LPTR BACK TO L
  006170   311                  RET
                        ;
                        ;   SUBROUTINE MSFH
                        ;
                        ;       THIS SUBROUTINE FETCHES THE SIGNS OF THE MANTISSAS
                        ;       OF THE FLOATING POINT NUMBERS POINTED TO BY (H,L)
                        ;       AND (H,B) INTO THE A AND E REGISTERS RESPECTIVELY.
                        ;
                        ;   REGISTERS ON EXIT:
                        ;
                        ;       A = SIGN  OF MANTISSA OF (H,L)
                        ;       E = SIGN OF MANTISSA OF (H,B)
                        ;       B,C,D,H,L = SAME AS ON ENTRY
                        ;
  006171   135          MSFH:   MOV E,L               ;SAVE LPTR
  006172   150                  MOV L,B               ;BPTR TO L
  006173   054                  INR L                 ;BPTR 2
  006174   054                  INR L                 ;/***TP
  006175   054                  INR L                 ;TO L
  006176   176                  MOV A,M               ; BPTR 2>TO A
1
8080 MACRO ASSEMBLER, VER 2.2  ERRORS = 0 PAGE 21


  006177   346 200              ANI 128               ;SAVE MANT SIGN
```

```
006201  153                     MOV L,E             ;LPTR BACK TO L
006202  137                     MOV E,A             ;STORE BPTR MANT SIGN
006203  054                     INR L               ;LPTR 2
006204  054                     INR L               ;/***TP
006205  054                     INR L               ;TO L
006206  176                     MOV A,M             ; LPTR 2>TO A
006207  346 200                 ANI 128             ;SAVE LPTR MANT SIGN
006211  055                     DCR L               ;LPTR BACK
006212  055                     DCR L               ;TO L
006213  055                     DCR L               ;/***TP
006214  311                     RET
                        ;                           SUBROUTINE BCTL
                        ;                           MOVES BPTR CHAR TO LPTR CHAR
                        ;                           DESTROYSE
006215  135             BCTL:   MOV E,L               ;LPTR TO E
006216  150                     MOV L,B             ;BPTR TO L
006217  054                     INR L               ;BPTR  2
006220  054                     INR L               ;/***TP
006221  054                     INR L               ;TO L
006222  176                     MOV A,M             ;BPTR CHAR TO A
006223  153                     MOV L,E             ;LPTR TO L
006224  054                     INR L               ;LPTR  2
006225  054                     INR L               ;TO L
006226  054                     INR L               ;/***TP
006227  167                     MOV M,A             ;STORE BPTR CHAR IN LPTR CHAR
006230  153                     MOV L,E             ;LPTR TO L
006231  311                     RET
                        ;*****************************************************
                        ;       ///SQUARE ROOT
                        ;          **********
                        ;       THE L REG PTS TO THE    TO BE
                        ;       OPERATED ON.
                        ;       THE B REG PTS TO THE LOC WHERE
                        ;       THE RESULT IS TO BE STORED
                        ;       THE C REG PTS TO 17(10) SCRATCH
                        ;       AREA.
                        ;       WHERE:
                        ;       C = ITERATION COUNT
                        ;                           C+1 = L REG
                        ;       C+2 = B REG
                        ;       C+3 TO C+6 = INTRL REG 1
                        ;       C+7 TO C+10 = INTRL REG 2
                        ;       C+11 TO C+14 = INTRL REG3
                        ;       C+15 =
                        ;*****************************************************
006232  175             DSQRT:  MOV A,L             ;STORE L IN
006233  151                     MOV L,C             ;2ND WRD SCRTCH
006234  066 000                 MVI M,0             ;INITIALIZE ITER COUNT
006236  054                     INR L
006237  167                     MOV M,A
```

8080 MACRO ASSEMBLER, VER 2.2 ERRORS = 0 PAGE 22

```
006240  054                     INR L               ;STR B IN 3RD
006241  160                     MOV M,B             ;WRD OF SCRTCH
006242  054                     INR L               ;SET C TO INTRL
006243  115                     MOV C,L             ;REG 1
006244  157                     MOV L,A             ;SET L PTR AT
006245  174                     MOV A,H             ;SET REGS FOR COPY
```

```
006246   315 210 016          CALL COPY          ;CPY   TO INTRL REG1
006251   315 046 016          CALL GCHR          ;PUT CHR IN A
006254   107                  MOV B,A            ;MAKE COPY
006255   346 200              ANI 200Q           ;CK NEG
006257   302 031 015          JNZ ERSQ
006262   170                  MOV A,B
006263   346 100              ANI 100Q           ;CK NEG EXP
006265   170                  MOV A,B
006266   312 302 014          JZ  EPOS
006271   037                  RAR                ;DIV BY 2
006272   346 177              ANI 177Q
006274   366 100              ORI 100Q           ;SET SIGN BIT
006276   167                  MOV M,A            ;SAVE 1ST APPROX
006277   303 306 014          JMP AGN4
006302   037           EPOS:  RAR                ;DIV BY 2
006303   346 177              ANI 177Q
006305   167                  MOV M,A            ;SAVE 1ST APPROX
006306   151           AGN4:  MOV L,C            ;SET REGS
006307   171                  MOV A,C            ;TO COPY 1ST
006310   306 004              ADI 4              ;APPROX
006312   117                  MOV C,A            ;INTO INTRL REG 2
006313   174                  MOV A,H            ;FRM INTRL REG1
006314   315 210 016          CALL COPY
006317   171                  MOV A,C
006320   326 004              SUI 4              ;MULTIPLY INTRL REG 1
006322   157                  MOV L,A
006323   101                  MOV B,C            ;TIMES INTRL REG2
006324   306 010              ADI 10Q            ;PLACE RESULT IN
006326   117                  MOV C,A            ;INTRL REG 3
006327   315 137 012          CALL LMUL
006332   171                  MOV A,C
006333   326 010              SUI 10Q            ;COPY ORG   INTO
006335   117                  MOV C,A            ;INTRL REG 1
006336   326 002              SUI 2
006340   157                  MOV L,A
006341   156                  MOV L,M
006342   174                  MOV A,H
006343   315 210 016          CALL COPY
006346   171                  MOV A,C
006347   306 010              ADI 10Q            ;ADD INTRL
006351   157                  MOV L,A            ;REG3 TO
006352   101                  MOV B,C            ;INTRL REG1
006353   306 004              ADI 4              ;ANS TO INTRL
006355   117                  MOV C,A            ;REG3
006356   315 134 011          CALL LADD
```

```
006361  175                     MOV A,L
006362  326 004                 SUI 4               ;DIV INTRL REG 3
006364  107                     MOV B,A             ;BY INTRL REG 2
006365  326 004                 SUI 4               ;PUT ANSW IN INTRL
006367  117                     MOV C,A             ;REG1
006370  315 000 011             CALL LDIV
006373  315 046 016             CALL GCHR
006376  326 001                 SUI 1
006400  346 177                 ANI 177Q
006402  167                     MOV M,A
006403  171                     MOV A,C
006404  326 003                 SUI 3               ;C PTS TO INTRL REG 1
006406  157                     MOV L,A             ;GET ITER
006407  106                     MOV B,M             ;COUNT NOW INCR
006410  004                     INR B
006411  160                     MOV M,B
006412  170                     MOV A,B
006413  376 005                 CPI 5               ;IF = 5 RTN ANS
006415  302 306 014             JNZ AGN4            ;OTHERWISE CONT
006420  151                     MOV L,C
006421  055             ALDN:   DCR L               ;COPY ANS INTO
006422  116                     MOV C,M             ;LOC REQUESTED
006423  054                     INR L
006424  174                     MOV A,H
006425  315 210 016             CALL COPY
006430  311                     RET
006431  151             ERSQ:   MOV L,C
006432  315 215 013             CALL    WZER          ;WRITE A FLOATING ZERO
006435  303 021 015             JMP ALDN
                        ;                           C+1 = L REG
                      ;
                       ;
                       ;
                       ;*****************************************************
                       ;       //// 5 DIGIT FLOATING PT. OUTPUT
                       ;*****************************************************
                       ;
                       ;
                       ;
                       ;
                       ;       *******ROUTINE TO CONVERT FLOATING PT.
                       ;       ***NUMBERS TO ASCII AND OUTPUT THEM VIA A SUBROUTINE
                       ;       ***CALLED OUTR  -  NOTE: THIS IS CURRENTLY SET
                       ;       ***TO ODT'S OUTPUT ROUTINE
                       ;
                       ;
006440  315 332 012 CVRT:     CALL    ZCHK            ;CHECK FOR NEW ZERO
006443  302 070 015           JNZ     NNZRO           ;NOT ZERO
006446  014                   INR     C               ;IT WAS, OFFSET C BY 2
006447  014                   INR     C
006450  151                   MOV     L,C
```

```
006451   315 215 013            CALL   WZER              ;WRITE ZERO
006454   315 031 016           CALL SIGN             ;SEND SPACE ON POS ZERO
                        ;
                        ;
006457   054                    INR    L                 ;PNT TO DECIMAL EXPONENT
006460   054                    INR    L
006461   054                    INR    L
006462   054                    INR    L
006463   257                    XRA    A                 ;SET IT TO ZERO
006464   167                    MOV    M,A
006465   303 227 015            JMP    MDSKP             ;OUTPUT IT
006470   126           NNZRO:   MOV    D,M               ;/GET THE NUMBER TO CONVERT
006471   054                   INR L
006472   106                   MOV B,M
006473   054                   INR L
006474   136                   MOV E,M
006475   054                   INR L              ;/4 WORD***TP
006476   176                   MOV A,M            ;/***TP
006477   014                   INR C              ;/OFFSET SCRATCH POINTER BY 2
006500   014                   INR C
006501   151                   MOV L,C            ;/L NOT NEEDED ANY MORE
006502   162                   MOV M,D            ;/SAVE NUMBER IN SCRATCH
006503   054                   INR L
006504   160                   MOV M,B
006505   054                   INR L
006506   163                   MOV M,E            ;/***TP
006507   054                   INR L              ;/***TP
006510   107                   MOV B,A            ;/SAVE COPY OF CHAR & SIGN
006511   346 177               ANI 177Q           ;GET ONLY CHAR.
006513   167                   MOV M,A            ;/SAVE ABS(NUMBER)
006514   376 100               CPI 100Q           ;CK FOR ZERO
006516   312 125 015           JZ  NZRO
006521   326 001               SUI 1              ;/GET SIGN OF DEC. EXP
006523   346 100               ANI 100Q           ;/GET SIGN OF CHAR.
006525   007           NZRO:   RLC                ;MOVE IT TO SIGN POSITION
006526   054                   INR L              ;/MOVE TO DECIMAL EXP.
006527   167                   MOV M,A            ;/SAVE SIGN OF EXP.
006530   170                   MOV A,B            ;/GET MANT. SIGH BACK
006531   315 031 016           CALL SIGN          ;/OUTPUT SIGN
006534   056 235               MVI L,(TEN5 AND 377Q)  ;/TRY MULT. OR DIV. BY 100000 FIRST
006536   315 172 016           CALL COPT          ;/MAKE A COPY IN RAM
006541   315 046 016  TST8:    CALL GCHR          ;/GET CHAR. OF NUMBER
006544   107                   MOV B,A            ;/SAVE A COPY
006545   346 100               ANI 100Q           ;/GET ABSOLUTE VALUE OF CHAR
006547   170                   MOV A,B            ;/INCASE PLUS
006550   312 156 015           JZ  GOTV           ;/ALREADY PLUS
006553   076 200               MVI A,200Q         ;/MAKE MINUS INTO PLUS
006555   220                   SUB B              ;/PLUS=200B-CHAR
006556   376 022       GOTV:   CPI 22Q            ;/TEST FOR USE OF 100000
006560   372 174 015           JM  TRY1           ;/WONT GO
006563   315 054 016           CALL MORD          ;/WILL GO SO DO IT
```

```
006566  306 005             ADI 5               ;/INCREMENT DEC. EXPONENT BY 5
006570  167                 MOV M,A             ;/UPDATE MEM
006571  303 141 015         JMP TST8            ;/GO TRY AGAIN
006574  056 241      TRY1:  MVI L,(TEN AND 377Q)  ;/NOW USE JUST TEN
006576  315 172 016         CALL COPT           ;/PUT IT IN RAM
006601  315 046 016  TST1:  CALL GCHR           ;/GET CHARACTERISTIC
006604  376 001             CPI 1               ;/MUST GET IN RANGE 1 TO 6
006606  362 222 015         JP  OK1             ;/ATLEAST ITS 1 OR BIGGER
006611  315 054 016  MDGN:  CALL MORD           ;/MUST MUL OF DIV BY 10
006614  306 001             ADI 1               ;/INCREMENT DECIMAL EXP.
006616  167                 MOV M,A             ;/UPDATE MEM
006617  303 201 015         JMP TST1            ;/NOW TRY AGAIN
006622  376 007       OK1:  CPI 7               ;/TEST FOR LESS THAN 7
006624  362 211 015         JP  MDGN            ;/NOPE - 7 OR GREATER
006627  151          MDSKP:  MOV    L,C                ;/SET UP DIGIT COUNT
006530  055                 DCR L
006631  055                 DCR L               ;/IN 1ST WORD OF SCRATCH
006632  066 005             MVI M,5             ;/5 DIGITS
006634  137                 MOV E,A             ;/SAVE CHAR. AS LEFT SHIFT COUNT
006635  315 377 015         CALL LSFT           ;/SHIFT LEFT PROPER NUMBER
006640  376 012             CPI 12Q             ;/TEST FOR 2 DIGITS HERE
006642  362 122 016         JP  TWOD            ;/JMP IF 2 DIGITS TO OUTPUT
006645  315 303 015         CALL DIGO           ;/OUTPUT FIRST DIGIT
006650  315 327 015  POPD:  CALL MULTT          ;/MULTIPLY THE NUMBER BY 10
006653  315 303 015  INPOP: CALL DIGO           ;/PRINT DIGIT IN A
006656  302 250 015         JNZ POPD            ;/MORE DIGITS?
006661  076 305             MVI A,305Q          ;/NO SO PRINT E
006663  315 060 000         CALL OUTR           ;/BASIC CALL TO OUTPUT
006666  315 107 016         CALL GETEX          ;/GET DECIMAL EXP
006671  107                 MOV B,A             ;/SAVE A COPY
006672  315 031 016         CALL SIGN           ;/OUTPUT SIGN
006675  170                 MOV A,B             ;/GET EXP BACK
006676  346 077             ANI 77Q             ;/GET GOOD BITS
006700  315 151 016         CALL CTWO           ;/GO CONVERT 2 DIGITS
006703  306 260       DIGO: ADI 260Q            ;/MAKE A INTO ASCII
006705  315 060 000         CALL OUTR           ;/OUTPUT DIGIT
006710  151                 MOV L,C             ;/GET DIGIT COUNT
006711  055                 DCR L               ;/BACK UP TO DIGIT COUNT
006712  055                 DCR L
006713  176                 MOV A,M             ;/TEST FOR DECIMAL PT
006714  376 005             CPI 5               ;/PRINT . AFTER 1ST DIGIT
006716  076 256             MVI A,256Q          ;/JUST IN CASE
006720  314 060 000         CZ  OUTR            ;/OUTPUT . IF 1ST DIGIT
006723  126                 MOV D,M             ;/NOW DECREMENT DIGIT COUNT
006724  025                 DCR D
006725  162                 MOV M,D             ;/UPDATE MEM AND LEAVE FLOPS SET
006726  311                 RET                 ;/SERVES AS TERM FOR DIGO & CVRT
006727  036 001       MULTT: MVI E,1            ;/MULT. BY 10 (START WITH X2)
006731  315 377 015         CALL LSFT           ;/LEFT SHIFT 1 = X2
006734  151                 MOV L,C             ;/SAVE X2 IN ¤RESULT¤
006735  055                 DCR L               ;/SET TO TOP OF NUMBER
```

```
006736  171                     MOV A,C         ;/SET C TO RESULT
006737  306 011                 ADI 11Q
006741  117                     MOV C,A         ;/NOW C SET RIGHT
006742  174                     MOV A,H         ;/SHOW RAM TO RAM TRANSFER
006743  315 210 016             CALL COPY       ;/SAVE X2 FINALLY
006746  171                     MOV A,C         ;/MUST RESET C
006747  326 011                 SUI 11Q         ;/BACK TO NORMAL
006751  117                     MOV C,A
006752  036 002                 MVI E,2         ;/NOW GET (X2)X4=X8
006754  151                     MOV L,C         ;/BUT MUST SAVE OVERFLOW
006755  055                     DCR L
006756  315 003 016             CALL TLP2       ;/GET X8
006761  151                     MOV L,C         ;/SET UP TO CALL DADD
006762  171                     MOV A,C         ;/SET B TO X2
006763  306 012                 ADI 12Q         ;/TO X2
006765  107                     MOV B,A
006766  315 006 013             CALL DADD       ;/ADD TWO LOW WORDS
006771  055                     DCR L           ;/BACK UP TO OVERFLOW
006772  176                     MOV A,M         ;/GET IT
006773  150                     MOV L,B         ;/NOW SET TO X2 OVERFLOW
006774  055                     DCR L           ;/ITS AT B-1
006775  216                     ADC M           ;/ADD WITH CARRY - CARRY WAS PRESERVED
006776  311                     RET             ;/ALL DONE, RETURN OVERFLOW IN A
006777  151             LSFT:   MOV L,C         ;/SET PTR FOR LEFT SHIFT OF NUMBER
007000  055                     DCR L           ;/BACK UP TO OVERFLOW
007001  257                     XRA A           ;/OVERFLOW=0 1ST TIME
007002  167             TLOOP:  MOV M,A         ;/SAVE OVERFLOW
007003  035             TLP2:   DCR E           ;/TEST FOR DONE
007004  370                     RM              ;/DONE WHEN E MINUS
007005  054                     INR L           ;/MOVE TO LOW
007006  054                     INR L
007007  054                     INR L           ;/***TP EXTENSION
007010  176                     MOV A,M         ;/SHIFT LEFT 4 BYTES
007011  027                     RAL
007012  167                     MOV M,A         ;/PUT BACK
007013  055                     DCR L           ;/***TP - ALL DONE
007014  176                     MOV A,M         ;/GET LOW
007015  027                     RAL             ;/SHIFT LEFT 1
007016  167                     MOV M,A         ;/RESTORE IT
007017  055                     DCR L           ;/BACK UP TO HIGH
007020  176                     MOV A,M         ;/GET HIGH
007021  027                     RAL             ;/SHIFT IT LEFT WITH CARRY
007022  167                     MOV M,A         ;/PUT IT BACK
007023  055                     DCR L           ;/BACK UP TO OVERFLOW
007024  176                     MOV A,M         ;/GET OVERFLOW
007025  027                     RAL             ;/SHIFT IT LEFT
007026  303 002 016             JMP TLOOP       ;/GO FOR MORE
007031  346 200         SIGN:   ANI 200Q        ;/GET SIGN BIT
007033  076 240                 MVI A,240Q      ;/SPACE INSTEAD OF PLUS
007035  312 042 016             JZ  PLSV        ;/TEST FOR +
007040  076 255                 MVI A,255Q      ;/NEGATIVE
```

```
007042  315 060 000  PLSV:   CALL OUTR          ;/OUTPUT SIGN
007045  311                  RET
007046  151           GCHR:   MOV L,C           ;/GET CHARCTERISTIC
007047  054           GETA:   INR L             ;/MOVE TO IT
007050  054                   INR L
007051  054                   INR L             ;/***TP
007052  176                   MOV A,M           ;/FETCH INTO A
007053  311                   RET               ;/DONE
007054  315 107 016  MORD:   CALL GETEX         ;/MUL OR DIV DEPENDING ON EXP
007057  137                   MOV E,A           ;/SAVE DECIMAL EXP
007060  105                   MOV B,L           ;/SET UP TO MULT OR DIV
007061  004                   INR B             ;/NOW BOP POINTER SET
007062  151                   MOV L,C           ;/L POINTS TO NUMBER TO CONVERT
007063  171                   MOV A,C           ;/POINT C AT ¤RESULT¤ AREA
007064  306 011               ADI 11Q           ;/IN SCRATCH
007066  117                   MOV C,A           ;/NOW C SET RIGHT
007067  173                   MOV A,E           ;/NOW TEST FOR MUL
007070  346 200               ANI 200Q          ;/TEST NEGATIVE DEC. EXP.
007072  312 114 016           JZ  DIVIT         ;/IF EXP IS + THEN DIVIDE
007075  315 137 012           CALL LMUL         ;/MULT.
007100  171           FINUP:  MOV A,C           ;/SAVE LOC. OF RESULT
007101  115                   MOV C,L           ;/C=LOC OF NUMBER (IT WAS DESTROYED)
007102  157                   MOV L,A           ;/SET L TO LOC. OF RESUTL
007103  174                   MOV A,H           ;/SHOW RAM TO RAM TRANSFER
007104  315 210 016           CALL COPY         ;/MOVE RESULT TO NUMBER
007107  151           GETEX:  MOV L,C           ;/NOW GET DECIMAL EXP
007110  054                   INR L
007111  303 047 016           JMP GETA          ;/USE PART OF GCHR
007114  315 000 011  DIVIT:   CALL LDIV         ;/DIVIDE
007117  303 100 016           JMP FINUP
007122  315 151 016  TWOD:    CALL CTWO         ;/CONVERT TO 2 DIGITS
007125  107                   MOV B,A           ;/SAVE ONES DIGIT
007126  315 107 016           CALL GETEX        ;/GET DECIMAL EXP
007131  137                   MOV E,A           ;/SAVE A COPY
007132  346 200               ANI 200Q          ;/TEST FOR NEGATIVE
007134  312 145 016           JZ  ADDI          ;/BUMP EXP BY 1 SINCE 2 DIGITS
007137  035                   DCR E             ;/DECREMENT NEGATIVE EXP SINCE 2 DIGITS
007140  163           FINIT:  MOV M,E           ;/RESTORE EXP WITH NEW VALUE
007141  170                   MOV A,B           ;/NOW DO 2ND DIGIT
007142  303 253 015           JMP INPOP         ;/GO OUT 2ND AND REST FO DIGITS
007145  034           ADDI:   INR E             ;/COMPENSATE FOR 2 DIGITS
007146  303 140 016           JMP FINIT
007151  036 377       CTWO:   MVI E,377Q        ;/CONVERT 2 DIGIT BIN TO BCD
007153  034           LOOP:   INR E             ;/ADD UP TENS DIGIT
007154  326 012               SUI 12Q           ;/SUBTRACT 10
007156  362 153 016           JP  LOOP          ;/TILL NEGATIVE RESULT
007161  306 012               ADI 12Q           ;/RESTORE ONES DIGIT
007163  107                   MOV B,A           ;/SAVE ONES DIGIT
007164  173                   MOV A,E           ;/GET TENS DIGIT
007165  315 303 015           CALL DIGO         ;/OUTPUT IT
007170  170                   MOV A,B           ;/SET A TO 2ND DIGIT
```

```
007171  311                     RET
007172  171             COPT:   MOV A,C             ;/COPY FROM 10 N TO RAM
007173  306 005                 ADI 5
007175  117                     MOV C,A             ;/SET C TO PLACE TO PUT
007176  076 016                 MVI A,(TEN5/256)
007200  315 210 016             CALL COPY           ;/COPY IT
007203  171                     MOV A,C             ;/NOW RESET C
007204  326 005                 SUI 5
007206  117                     MOV C,A             ;/ITS RESET
007207  311                     PET
007210  104             COPY:   MOV B,H             ;/SAVE RAM H
007211  147                     MOV H,A             ;/SET TO SOURCE H
007212  176                     MOV A,M             ;/GET 4 WORDS INTO THE REGS.
007213  054                     INR L
007214  126                     MOV D,M
007215  054                     INR L
007216  136                     MOV E,M
007217  054                     INR L
007220  156                     MOV L,M             ;/LAST ONE ERASES L
007221  140                     MOV H,B             ;/SET TO DESTINATION RAM
007222  105                     MOV B,L             ;/SAVE 4TH WORD IN B
007223  151                     MOV L,C             ;/SET TO DESTINATION
007224  167                     MOV M,A             ;/SAVE FIRST WORD
007225  054                     INR L
007226  176                     MOV A,M             ;/SAVE THIS WORD IN A (INPUT SAVES C HERE
007227  162                     MOV M,D             ;/NOW PUT 2ND WORD
007230  054                     INR L
007231  163                     MOV M,E
007232  054                     INR L
007233  160                     MOV M,B             ;/ALL 4  COPIED NOW
007234  311                     RET                 ;/ALL DONE
                        ;
                        ;
007235  303 120 000     TEN5:   DB 303Q,120Q,0Q,21Q  ;/303240(8) = 100000.
007241  240 000 000     TEN:    DB 240Q,0Q,0Q,4Q  ;/12(8) = 10
                        ;
                        ;       SCRATCH MAP FOR I/O CONVERSION ROUTINES
                        ;
                        ;       RELATIVE TO (C+2)USE
                        ;       C-2             DIGIT COUNT
                        ;       C-1             OVERFLOW
                        ;       C               HIGH NUMBER - MANTISSA
                        ;       C+1             LOW NUMBER
                        ;       C+2             CHARACTERISTIC
                        ;       C+3             DECIMAL EXPONEXT (SIGN & MAG.)
                        ;       C+4             TEN**N
                        ;       C+5             TEN**N
                        ;       C+6             TEN**N
                        ;       C+7             RESULT OF MULT & DIV
                        ;       C+8             AND TEMP FOR X2
                        ;       C+9             ¤       ¤
```

```
                        ;       C+10            L FOR NUMBER TO GO INTO (INPUT ONLY)
                        ;       C+11            DIGIT JUST INPUT (INPUT ONLY)
                        ;
                        ;
                        ;                       /*****BEGIN INPUT*************
                        ;
007245  076 277         ERR:    MVI A,277Q        ;ERROR IN INPUT
007247  315 060 000             CALL OUTR         ;/SEND A ?(SPACE)
007252  076 240                 MVI A,240Q
007254  315 060 000             CALL OUTR         ;/OUTPUT SPACE
007257  303 272 016             JMP PRMT          ;/GO PROMPT USER AND RESTART
                        ;
                        ;*****************************************************
                        ;       //// 4 1/2 DIGIT INPUT ROUTINE
                        ;*****************************************************
                        ;
                        ;
                        ;                       /L POINTS TO WHERE TO PUT INPUT NUMBER
                        ;                       /C POINTS TO 13(10) WORDS OF SCRATCH
                        ;
007262  105             INPUT:  MOV B,L           ;/SAVE ADDRESS WHERE DATA IS TO GO
007263  171                     MOV A,C           ;/IN SCRATCH
007264  306 017                 ADI 17Q           ;/COMPUTE LOC. IN SCRATCH
007266  157                     MOV L,A
007267  160                     MOV M,B           ;/PUT IT
007270  014                     INR C             ;/OFFSET SCRATCH POINTER
007271  014                     INR C             ;/BY 2
007272  076 272         PRMT:   MVI A,272Q        ;/PROMPT USER WITH :
007274  315 060 000             CALL OUTR         ;/OUTPUT :
007277  315 305 017             CALL ZROIT        ;/ZERO NUMBER
007302  054                     INR L             ;/AND ZERO
007303  167                     MOV M,A           ;/DECIMAL EXPONENT
007304  315 142 017             CALL GNUM         ;/GET INTEGER PART OF NUM
007307  376 376                 CPI 376Q          ;/TERM=.?
007311  312 034 017             JZ  DECPT         ;/YES
007314  376 025         TSTEX:  CPI 25Q           ;/TEST FOR E
007316  312 061 017             JZ  INEXP         ;/YES - HANDLE EXP
007321  376 360                 CPI 360Q          ;/TEST FOR SPACE TERM (240B-260B)
007323  302 245 016             JNZ ERR           ;/NOT LEGAL TERM
007326  315 253 017             CALL FLTSGN       ;/FLOAT   AND SIGN IT
007331  315 107 016  SCALE:     CALL GETEX        ;/GET DECIMAL EXP
007334  346 177                 ANI 177Q          ;/GET GOOD BITS
007336  137                     MOV E,A           ;/SAVE COPY
007337  346 100                 ANI 100Q          ;/GET SIGN OF EXP
007341  007                     RLC               ;/INTO SIGN BIT
007342  267                     ORA A             ;/SET FLOPS
007343  107                     MOV B,A           ;/SAVE SIGN
007344  173                     MOV A,E           ;/GET EXP BACK
007345  312 353 016             JZ  APLS          ;/JMP IS +
007350  076 200                 MVI A,200Q        ;/MAKE MINUS +
```

```
007352  223                     SUB E               ;/NOW ITS +
007353  200             APLS:   ADD B               ;/SIGN NUMBER
007354  167                     MOV M,A             ;/SAVE EXP (SIGN & MAG.)
007355  056 235                 MVI L,(TEN5 AND 377Q)  ;/TRY MORD WITH 10**5 FIRST
007357  315 172 016             CALL COPT           ;/TRANSFER TO RAM
007362  315 107 016             CALL GETEX          ;/GET DECIMAL EXP
007365  346 077         INT5:   ANI 77Q             ;/GET MAG. OF EXP
007367  376 005                 CPI 5Q              ;/TEST FOR USE OF 10**5
007371  372 005 017             JM  TRYTN           ;/WONT GO - TRY 10
007374  315 054 016             CALL MORD           ;/WILL GO SO DO IT
007377  326 005                 SUI 5Q              ;/MAG = MAG -5
007401  167                     MOV M,A             ;/UPDATE DEC. EXP IN MEM
007402  303 365 016             JMP INT5            ;/GO TRY AGAIN
007405  056 241       TRYTN:    MVI L,(TEN AND 377Q)  ;/PUT TEN IN RAM
007407  315 172 016             CALL COPT
007412  315 107 016             CALL GETEX          ;/SET UP FOR LOOP
007415  346 077         INT1:   ANI 77Q             ;/GET MAGNITUDE
007417  267                     ORA A               ;/TEST FOR 0
007420  312 257 017             JZ  SAVEN           ;/DONE, MOVE NUM OUT AND GET OUT
007423  315 054 016             CALL MORD           ;/NOT DONE - DO 10
007426  326 001                 SUI 1Q              ;/EXP = EXP -1
007430  167                     MOV M,A             ;/UPDATE MEM
007431  303 015 017             JMP INT1            ;/TRY AGAIN
007434  151             DECPT:  MOV L,C             ;/ZERO DIGIT COUNT
007435  055                     DCR L               ;/SINCE ITS NECESSARY
007436  055                     DCR L               ;/TO COMPUTE EXP.
007437  066 000                 MVI M,0             ;/ZEROED
007441  315 245 017             CALL EP1            ;/GNUM IN MIDDLE
007444  137                     MOV E,A             ;/SAVE TERMINATOR
007445  151                     MOV L,C             ;/MOVE DIGIT COUNT TO EXP
007446  055                     DCR L               ;/BACK UP TO DIGIT COUNT
007447  055                     DCR L
007450  106                     MOV B,M             ;/GOT DIGIT COUNT
007451  315 107 016             CALL GETEX          ;/SET L TO DEC. EXP
007454  160                     MOV M,B             ;/PUT EXP
007455  173                     MOV A,E             ;/TERM BACK TO A
007456  303 314 016             JMP TSTEX           ;/TEST FOR E+OR-XX
007461  315 253 017   INEXP:    CALL FLTSGN         ;/FLOAT AND SIGN NUMBER
007464  315 257 017             CALL SAVEN          ;/SAVE NUMBER IN (L) TEMP
007467  315 305 017             CALL ZROIT          ;/ZERO OUT NUM. FOR INPUTTING EXP
007472  315 142 017             CALL GNUM           ;/NOW INPUT EXPONENT
007475  376 360                 CPI 360Q            ;/TEST FOR SPACE TERM.
007477  302 245 016             JNZ ERR             ;/NOT LEGAL - TRY AGAIN
007502  151                     MOV L,C             ;/GET EXP OUT OF MEM
007503  054                     INR L               ;/***TP
007504  054                     INR L               ;/EXP LIMITED TO 5 BITS
007505  176                     MOV A,M             ;/GET LOWEST 8 BITS
007506  346 037                 ANI 37Q             ;/GET GOOD BITS
007510  107                     MOV B,A             ;/SAVE THEM
007511  054                     INR L               ;/GET SIGN OF EXP
007512  176                     MOV A,M             ;/INTO A
```

```
007513  267                     ORA A           ;/SET FLOPS
007514  170                     MOV A,B         ;/INCASE NOTHING TO DO
007515  372 123 017             JM  USEIT       ;/IF NEG. USE AS +
007520  076 000                 MVI A,0Q        ;/IF + MAKE -
007522  220                     SUB B           ;/0-X = -X
007523  054             USEIT:  INR L           ;/POINT AT EXP
007524  206                     ADD M           ;/GET REAL DEC. EXP
007525  167                     MOV M,A         ;/PUT IN MEM
007526  171                     MOV A,C         ;/NOW GET NUMBER BACK
007527  306 015                 ADI 15Q         ;/GET ADD OF L
007531  157                     MOV L,A         ;/L POINTS TO L OF NUMBER
007532  156                     MOV L,M         ;/NOW L POINTS TO NUMBER
007533  174                     MOV A,H         ;/RAM TO RAM COPY
007534  315 210 016             CALL COPY       ;/COPY IT BACK
007537  303 331 016             JMP SCALE       ;/NOW ADJUST FOR EXP
007542  315 333 000  GNUM:     CALL INP        ;/GET A CHAR
007545  376 240                 CPI 240Q        ;/IGNORE LEADING SPACES
007547  312 142 017             JZ  GNUM
007552  376 255                 CPI 255Q        ;/TEST FOR -
007554  302 170 017             JNZ TRYP        ;/NOT MINUS
007557  151                     MOV L,C         ;/MINUS SO SET SIGN
007560  054                     INR L           ;/IN CHAR LOC.
007561  054                     INR L           ;/***TP
007562  054                     INR L
007563  066 200                 MVI M,200Q      ;/SET - SIGN
007565  303 142 017             JMP GNUM
007570  376 253         TRYP:   CPI 253Q        ;/IGNORE +
007572  312 142 017             JZ  GNUM
007575  326 260         TSTN:   SUI 260Q        ;/STRIP ASCII
007577  370                     RM              ;/RETURN IF TERM
007600  376 012                 CPI 12Q         ;/TEST FOR NUMBER
007602  360                     RP              ;/ILLEGAL
007603  137                     MOV E,A         ;/SAVE DIGIT
007604  315 277 017             CALL GETN       ;/LOC. OF DIGIT STORAGE TO L
007607  163                     MOV M,E         ;/SAVE DIGIT
007610  315 327 015             CALL MULTT      ;/MULT NUMBER BY 10
007613  267                     ORA A           ;/TEST FOR TOO MANY DIGITS
007614  300                     RNZ             ;/TOO MANY DIGITS
007615  315 277 017             CALL GETN       ;/GET DIGIT
007620  151                     MOV L,C         ;/SET L TO NUMBER
007621  054                     INR L
007622  054                     INR L           ;/***TP
007623  206                     ADD M           ;/ADD IN THE DIGIT
007624  167                     MOV M,A         ;/PUT RESULT BACK
007625  055                     DCR L           ;/NOW DO HIGH
007626  176                     MOV A,M         ;/GET HIGH TO ADD IN CARRY
007627  316 000                 ACI 0Q          ;/ADD IN CARRY
007631  167                     MOV M,A         ;/UPDATE HIGH
007632  055                     DCR L           ;/***TP EXTENSION
007633  176                     MOV A,M
007634  316 000                 ACI 0Q          ;/ADD IN CARRY
```

```
007636  167                     MOV M,A             ;/***TP ALL DONE
007637  330                     RC                  ;/OVERFLOW ERROR
007640  055                     DCR L               ;/BUMP DIGIT COUNT NOW
007641  055                     DCR L
007642  106                     MOV B,M             ;/GET DIGIT COUNT
007643  004                     INR B               ;/BUMP DIGIT COUNT
007644  160                     MOV M,B             ;/UPDATE DIGIT COUNT
007645  315 333 000  EPI:       CALL INP            ;/GET NEXT CHAR
007650  303 175 017             JMP TSTN            ;/MUST BE NUM. OR TERM
007653  151          FLTSGN:    MOV     L,C           ;POINT L AT NUMBER TO FLOAT
007654  303 325 012               JMP     FLOAT       ;GO FLOAT IT
007657  171            SAVEN:   MOV A,C             ;/PUT NUMBER IN (L)
007660  306 015                   ADI 15Q             ;/GET ADD OF L
007662  157                     MOV L,A
007663  136                     MOV E,M             ;/GET L OF RESULT
007664  153                     MOV L,E             ;/POINT L AT (L)
007665  054                     INR L               ;/SET TO 2ND WORD TO SAVE C
007666  161                     MOV M,C             ;/SAVE C IN (L) +1 SINCE IT WILL BE DESTROYED
007667  151                     MOV L,C             ;/SET UP TO CALL COPY
007670  113                     MOV C,E             ;/NOW L&C SET
007671  174                     MOV A,H             ;/RAM TO RAM COPY
007672  315 210 016             CALL COPY           ;/COPY TO L
007675  117                     MOV C,A             ;/(L)+1 RETURNED HERE SO SET AS C
007676  311                     RET                 ;/NOW EVERYTHING HUNKY-DORRY
007677  171            GETN:    MOV A,C             ;/GET DIGIT
007700  306 016                   ADI 16Q             ;/LAST LOC. IN SCRATCH
007702  157                     MOV L,A             ;/PUT IN L
007703  176                     MOV A,M             ;/GET DIGIT
007704  311                     RET
007705  151            ZROIT:   MOV L,C             ;/ZERO NUMBER
007706  257                     XRA A
007707  167                     MOV M,A             ;/***TP
007710  054                     INR L               ;/***TP
007711  167                     MOV M,A
007712  054                     INR L
007713  167                     MOV M,A
007714  054                     INR L               ;/NOW SET SIGN TO +
007715  167                     MOV M,A
007716  311                     RET                 ;/DONE
                                END
NO PROGRAM ERRORS
```

SYMBOL TABLE

* 01

A	000007	ABCH	005064	ACPR	005757	ADDI	007145
ADD2	004737	ADDZ	004726	AGN4	006306	ALDN	006421
AORS	005741	APLS	007353	B	000000	BBCH	005075
BCHK	005342	BCTL	006215	BMIN	005042	C	000001
CCHK	006104	CCMP	005533	CFCHE	005514	COMI	005026
COM2	005050	COPT	007172	COPY	007210	CRIN	004515
CSIGN	006151	CSTR	006161	CTWO	007151	CVRT	006440 *
D	000002	DADD	005406	DCLR	005435	DCMP	005764
DECPT	007434	DECR	005724 *	DFXL	005316 *	DIGO	006703
DIVIT	007114	DLST	005351	DRST	005370	DSQRT	006232 *
DSUB	005446	DTST2	004422	E	000003	ENTI	006020
ENT2	006012	EPI	007645	EPOS	006302	EQ02	004634
EQUL	004624 *	ERR	007245	ERSQ	006431	FINIT	007140
FINUP	007100	FLOAT	005325	FLTSG	007653	FXLI	005262 *
FXL2	005263	GCHAR	005501	GCHR	007046	GETA	007047
GETEX	007107	GETN	007677	GNUM	007542	GOON	004473
GOTV	006556	H	000004	INCR	005666	INCR2	005706
INCR3	005703	INDFI	005604 *	INDFC	005650	INEXP	007461
INP	000333	INPOP	006653	INPUT	007262 *	INTI	007415
INT5	007365	INTR	005177	KPGO	005166	L	000005
L000	005106	L001	005110	L002	005037	L003	005113
L128	005116	L129	005121	L131	005124	LADD	004534
LAOS	004542	LASD	005002	LDCP	006071	LDIV	004400
LLTB	004577	LMCM	005127 *	LMCP	006100	LMUL	005137
LOOP	007153	LSFT	006777	LSUB	004540 *	LXFR	006044
M	000006	MADD	005244	MANT	004757	MAXCH	000077
MDGN	006611	MDSKP	006627	MINCH	000300	MORD	007054
MSFH	006171	MULTT	006727	NCHK	004604	NNZRO	006470
NORM	005255	NORMI	005256 *	NOTO	004652	NZRO	006525
OFLWI	005566	OFLWC	006133	OKI	006622	OUTR	000060
OVER	006027	PLSV	007042	POPD	006650	PRMT	007272
PSW	000006	REP3	004447	REP5	006046	REP6	005271
SAVEN	007657	SCALE	007331	SCCFG	005722	SCHAR	005313
SHIO	004614	SIGN	007031	SP	000006	STORC	006123
SUBZ	004671	TEN	007241	TEN5	007235	TLOOP	007002
TLP2	007003	TRYI	006574	TRYP	007570	TRYTN	007405
TSTI	006601	TST8	006541	TSTEX	007314	TSTN	007575
TSTR	005747	TWOD	007122	UFLWI	005550	UFLWC	006142
USEIT	007523	WCHAR	005637	WFLT	001607	WIND	005577
WMANT	005630	WOVR	005561	WUND	005543	WZER	005615
WZERC	005657	ZCHK	005332	ZMCHK	005332	ZROIT	007705

* 02

* 03

* 04

RAC/gw

RERUN: 12/30/75
80 Jane Levdar, T-46
120 Gene Fisher, T-101

RERUN: 8/26/76
Jane Levdar 125

RERUN: 12/28/76
Jane Levdar 150
RERUN: 7/12/77
Jane Levdar 100

Technical Information Department
LAWRENCE LIVERMORE LABORATORY
University of California | Livermore, California | 94550

171 286