```
  879  000726' 001000  000174          12740          MOV     A,H             ;SUBTRACT MIDDLE ORDER
  880  000727' 001000  000336          12760  FDIVB:  SBI     0
  881  000730' 000000  000000
  882  000731' 001000  000147          12780          MOV     H,A
  883  000732' 001000  000170          12800          MOV     A,B             ;SUBTRACT HO
  884  000733' 001000  000336          12820  FDIVA:  SBI     0
  885  000734' 001000  000000
  886  000735' 001000  000107          12840          MOV     B,A
  887  000736' 001000  000076          12860  FDIVG:  MVI     A,0             ;GET HIGHEST ORDER
  888  000737' 000000  000000
  889  000740' 001000  000336          12880          SBI     0               ;SUBTRACT THE CARRY FROM IT
  890  000741' 000000  000000
  891  000742' 001000  000077          12900          CMC                     ;SET CARRY TO CORESPOND TO NEXT QUOTIENT BIT
  892  000743' 001000  000322          12920          JNC     FDIV2           ;GET OLD NUMBER BACK IF WE SUBTRACTED TOO MUCH
  893  000744' 000000  000755'
  894  000745' 000000  000716'
  895  000746' 001000  000062          12940          STA     FDIVG+1         ;UPDATE HIGHEST ORDER
  896  000747' 000000  000737'
  897  000750' 000000  000744'
  898  000751' 001000  000361          12960          POP     PSW             ;THE SUBTRACTION WAS GOOD
  899  000752' 001000  000361          12980          POP     PSW             ;GET PREVIOUS NUMBER OFF STACK
  900  000753' 001000  000067          13000          STC                     ;NEXT BIT IN QUOTIENT IS A ONE
  901  000754' 001000  000322          13020          XNO     1000,322        ;"JNC" AROUND NEXT 2 BYTES
  902  000755' 001000  000301          13040  FDIV2:  POP     B               ;WE SUBTRACTED TOO MUCH
  903  000756' 001000  000341          13060          POP     H               ;GET OLD NUMBER BACK
  904  000757' 001000  000171          13080          MOV     A,C             ;ARE WE DONE?
  905  000760' 001000  000074          13100          INR     A               ;SET SIGN FLAG WITHOUT AFFECTING CARRY
  906  000761' 001000  000075          13120          DCR     A
  907  000762' 001000  000037          13140          RAR                     ;PUT CARRY IN MSB
  908  000763' 001000  000372          13160          JM      ROUNDB          ;WE ARE DONE
  909  000764' 000000  000234'
  910  000765' 000000  000747'
  911  000766' 001000  000027          13180          RAL                     ;WE AREN'T, GET OLD CARRY BACK
  912                                   13200  IFE     LENGTH,<
  913                                   13220          CALL    SHFTLO>         ;ROTATE EVERYTHING LEFT ONE
  914                                   13240  IFN     LENGTH,<
  915  000767' 001000  000173          13260          MOV     A,E             ;ROTATE EVERYTHING LEFT ONE
  916  000770' 001000  000027          13280          RAL                     ;ROTATE NEXT BIT OF QUOTIENT IN
  917  000771' 001000  000137          13300          MOV     E,A
  918  000772' 001000  000172          13320          MOV     A,D
  919  000773' 001000  000027          13340          RAL
  920  000774' 001000  000127          13360          MOV     D,A
  921  000775' 001000  000171          13380          MOV     A,C
  922  000776' 001000  000027          13400          RAL
  923  000777' 001000  000117          13420          MOV     C,A>
  924  001000' 001000  000051          13440          DAD     H               ;ROTATE A ZERO INTO RIGHT END OF NUMBER
  925  001001' 001000  000170          13460          MOV     A,B             ;THE HO BYTE, FINALLY!
  926  001002' 001000  000027          13480          RAL
  927  001003' 001000  000107          13500          MOV     B,A
  928  001004' 001000  000072          13520          LDA     FDIVG+1         ;ROTATE THE HIGHEST ORDER
  929  001005' 000000  000737'
  930  001006' 000000  000764'
  931  001007' 001000  000027          13540          RAL
```



```
  932  001010' 001000  000062          13560          STA     FDIVG+1
  933  001011' 000000  000737'
  934  001012' 000000  001025'
  935  001013' 001000  000171          13580          MOV     A,C             ;ADD ONE TO EXPONENT IF THE FIRST SUBTRACTION
  936  001014' 001000  000262          13600          ORA     D               ; DID NOT WORK
  937  001015' 001000  000263          13620          ORA     E
  938  001016' 001000  000302          13640          JNZ     FDIV1           ;THIS ISN'T THE CASE
  939  001017' 000000  000720'
  940  001020' 000000  001011'
  941  001021' 001000  000345          13660          PUSH    H               ;SAVE PART OF NUMBER
  942  001022' 001000  000041          13680          LXI     H,FAC           ;GET POINTER TO FAC
  943  001023' 000000  000426*
  944  001024' 000000  001017'
  945  001025' 001000  000065          13700          DCR     M               ;DECREMENT EXPONENT
  946  001026' 001000  000341          13720          POP     H               ;GET NUMBER BACK
  947  001027' 001000  000302          13740          JNZ     FDIV1           ;DIVIDE MORE IF NO OVERFLOW OCCURED
  948  001030' 000000  000720'
  949  001031' 000000  001023'
  950  001032' 001000  000303          13760          JMP     OVERR           ;OVERFLOW!!
  951  001033' 000000  000267'
  952  001034' 000000  001030'
  953
  954
  955                                   13820          ;CHECK SPECIAL CASES AND ADD EXPONENTS FOR FMULT, FDIV
  956                                   13840          ;ALTERS A,B,H,L
  957                                   13860  IFE     LENGTH-2,<
  958                                   13880  MULDVS: MVI     A,377           ;ENTRY FROM DDIV, SUBTRACT EXPONENTS
  959                                   13900          XWD     1000,056        ;"MVI  L" AROUND NEXT BYTE
  960                                   13920  MULDVA: XRA     A               ;ENTRY FROM DMULT, ADD EXPONENTS
  961                                   13940          LXI     H,ARG-1         ;GET POINTER TO SIGN AND HO OF ARG
  962                                   13960          MOV     C,M             ;GET HO AND SIGN FOR UNPACKING
  963                                   13980          INX     H               ;INCREMENT POINTER TO EXPONENT
  964                                   14000          MOV     B,M             ;GET EXPONENT FOR BELOW
  965                                   14020          MOV     L,A>            ;SAVE ADD OR SUBTRACT FLAG
  966  001035' 001000  000170          14040  MULDIV: MOV     A,B             ;IS NUMBER IN REGISTERS ZERO?
  967  001036' 001000  000267          14060          ORA     A
  968  001037' 001000  000312          14080          JZ      MULDV2          ;IT IS, ZERO FAC AND WE ARE DONE
  969  001040' 000000  001077'
  970  001041' 000000  001033'
  971  001042' 001000  000175          14100          MOV     A,L             ;GET ADD OR SUBTRACT FLAG
  972  001043' 001000  000041          14120          LXI     H,FAC           ;GET POINTER TO EXPONENT
  973  001044' 000000  001023*
  974  001045' 000000  001040'
  975  001046' 001000  000256          14140          XRA     M               ;GET EXPONENT
  976  001047' 001000  000200          14160          ADD     B               ;ADD IN REGISTER EXPONENT
  977  001050' 001000  000107          14180          MOV     B,A             ;SAVE IT
  978  001051' 001000  000037          14200          RAR                     ;CHECK FOR OVERFLOW
  979  001052' 001000  000250          14220          XRA     B               ;OVERFLOW IF SIGN IS THE SAME AS CARRY
  980  001053' 001000  000170          14240          MOV     A,B             ;GET SUM
  981  001054' 001000  000362          14260          JP      MULDV1          ;WE HAVE OVERFLOW!!
  982  001055' 000000  001076'
  983  001056' 000000  001044'
  984  001057' 001000  000306          14280          ADI     200             ;PUT EXPONENT IN EXCESS 200
```

```
  985  001060' 000000  000200
  986  001061' 001000  000167          14300            MOV     M,A             ;SAVE IT IN THE FAC
  987  001062' 001000  000316          14320            JZ      POPHRT          ;WE HAVE UNDEFLOW!! RETURN.
  988  001065' 000000  000630'
  989  001064' 000000  001055'
  990  001065' 001000  000315          14340            CALL    UNPACK          ;UNPACK THE ARGUMENTS
  991  001066' 000000  001272'
  992  001067' 000000  001063'
  993  001070' 001000  000167          14360            MOV     M,A             ;SAVE THE NEW SIGN
  994  001071' 001000  000053          14380            DCX     H               ;POINT TO EXPONENT
  995  001072' 001000  000311          14400            RET                     ;ALL DONE, LEAVE NO IN A
  996                                   14420  IFN       EXTFNC,<
  997  001073' 001000  000357          14440  MLOVEX:   FSIGN                   ;ENTRY FROM EXP, PICK UNDERFLOW IF NEGATIVE
  998  001074' 001000  000057          14460            CMA                     ;PICK OVERFLOW IF POSITIVE
  999  001075' 001000  000341          14480            POP     H>              ;DON'T SCREW UP STACK
 1000  001076' 001000  000267          14500  MULOV1:   ORA     A               ;IS ERROR OVERFLOW OR UNDEFLOW?
 1001  001077' 001000  000341          14520  MULOV2:   POP     H               ;GET OLD RETURN ADDRESS OFF STACK
 1002                                   14540  IFE       LENGTH,<
 1003                                   14560            JM      OVERR           ;OVERFLOW
 1004                                   14580                                    ;UNDERFLOW -- FALL INTO ZERO
 1005                                   14600
 1006                                   14620
 1007                                   14640            ;ZERO FAC
 1008                                   14660            ;ALTERS A ONLY
 1009                                   14680            ;EXITS WITH A=0
 1010                                   14700  ZERO:     XRA     A               ;ZERO A
 1011                                   14720            STA     FAC             ;ZERO FAC
 1012                                   14740            RET>                    ;ALL DONE
 1013
 1014
 1015                                   14800  IFN       LENGTH,<
 1016  001100' 001000  000362          14820            JP      ZERO            ;UNDERFLOW
 1017  001101' 000000  000173'
 1018  001102' 000000  001066'
 1019  001103' 001000  000303          14840            JMP     OVERR>          ;OVERFLOW
 1020  001104' 000000  000267'
 1021  001105' 000000  001101'
 1022
 1023
 1024                                   14900            ;MULTIPLY FAC BY 10
 1025                                   14920            ;ALTERS A,B,C,D,E,M,L
 1026  001106' 001000  000315          14940  MUL10:    CALL    MOVRF           ;GET NUMBER IN REGISTERS
 1027  001107' 000000  001240'
 1028  001110' 000000  001104'
 1029  001111' 001000  000170          14960            MOV     A,B             ;GET EXPONENT
 1030  001112' 001000  000267          14980            ORA     A               ;RESULT IS ZERO IF ARG IS ZERO
 1031  001113' 001000  000310          15000            RZ                      ;IT IS
 1032  001114' 001000  000306          15020            ADI     2               ;MULTIPLY BY 4 BY ADDING 2 TO EXPONENT
 1033  001115' 000000  000002
 1034  001116' 001000  000332          15040            JC      OVERR           ;OVERFLOW!!
 1035  001117' 000000  000267'
 1036  001120' 000000  001107'
 1037  001121' 001000  000107          15060            MOV     B,A             ;RESTORE EXPONENT
```



```
 1038  001122' 001000  000315          15080            CALL    FADD            ;ADD IN ORIGINAL NUMBER TO GET 5 TIMES IT
 1039  001123' 000000  000025'
 1040  001124' 000000  001117'
 1041  001125' 001000  000041          15100            LXI     H,FAC           ;ADD 1 TO EXPONENT TO MULTIPLY NUMBER BY
 1042  001126' 000000  001044*
 1043  001127' 000000  001125'
 1044  001130' 001000  000064          15120            INR     M               ; 2 TO GET 10 TIMES ORIGINAL NUMBER
 1045  001131' 001000  000300          15140            RNZ                     ;ALL DONE IF NO OVERFLOW
 1046  001132' 001000  000303          15160            JMP     OVERR           ;OVERFLOW!!
 1047  001133' 000000  000267'
 1048  001134' 000000  001126'
 1049                                   15180  PAGE
```

```
                                        15200   SUBTTL  SIGN, SGN, FLOAT, NEG AND ABS
                                        15220           ;PUT SIGN OF FAC IN A
                                        15240           ;ALTERS A ONLY
                                        15260           ;LEAVES FAC ALONE
                                        15280           ;NOTE: TO TAKE ADVANTAGE OF THE RST INSTRUCTIONS TO SAVE BYTES,
                                        15300           ;FSIGN IS DEFINED TO BE AN RST.  "FSIGN" IS EQUIVALENT TO "CALL SIGN"
                                        15320           ;THE FIRST FEW INSTRUCTIONS OF SIGN (THE ONES BEFORE SIGNC) ARE DONE
                                        15340           ;IN THE 8 BYTES AT THE RST LOCATION.
                                        15360   REPEAT  0,<                      ;FSIGN IS ALWAYS AN RST
                                        15380   SIGN:   LDA     FAC             ;CHECK IF THE NUMBER IS ZERO
                                        15400           ORA     A
                                        15420           RZ>                     ;IT IS, A IS ZERO
001135' 001000  000072                  15440   SIGNC:  LDA     FAC-1           ;GET SIGN OF FAC, IT IS NON-ZERO
001136' 777777  777777*
001137' 000000  001133'
                                        15460           XWD     1000,376        ;"CPI" AROUND NEXT BYTE
001140' 001000  000057                  15480   FCOMPS: CMA                     ;ENTRY FROM FCOMP, COMPLEMENT SIGN
001142' 001000  000027                  15500   ICOMPS: RAL                     ;ENTRY FROM ICOMP, PUT SIGN BIT IN CARRY
001143' 001000  000237                  15520   SIGNS:  SBB     A               ;A=0 IF CARRY WAS 0, A=377 IF CARRY WAS 1
001144' 001000  000300                  15540           RNZ                     ;RETURN IF NUMBER WAS NEGATIVE
001145' 001000  000074                  15560   INRART: INR     A               ;PUT ONE IN A IF NUMBER WAS POSITIVE
001146' 001000  000311                  15580           RET                     ;ALL DONE
                                        15600
                                        15620           ;SGN FUNCTION
                                        15640           ;ALTERS A,B,C,D,E,H,L
                                        15660   IFN     LENGTH=2,<
001147' 001000  000357                  15700   SGN:    FSIGN>                  ;GET SIGN OF FAC IN A
                                        15720                                   ;FALL INTO FLOAT
                                        15740
                                        15760
                                        15780           ;FLOAT THE SIGNED INTEGER IN A
                                        15800           ;ALTERS A,B,C,D,E,H,L
001150' 001000  000006                  15820   FLOAT:  MVI     B,210           ;SET EXPONENT CORRECTLY
001151' 000000  000210
001152' 001000  000021                  15840           LXI     D,$CODE         ;ZERO D,E
001153' 000000  000601*
001154' 000000  001136'
                                        15860                                   ;FALL INTO FLOATR
                                        15880
                                        15900
                                        15920           ;FLOAT THE SIGNED NUMBER IN B,A,D,E
                                        15940           ;ALTERS A,B,C,D,E,H,L
001155' 001000  000041                  15960   FLOATR: LXI     H,FAC           ;GET POINTER TO FAC
001156' 000000  001126*
001157' 000000  001153'
001160' 001000  000117                  15980           MOV     C,A             ;PUT MO IN C
001161' 001000  000160                  16000           MOV     M,B             ;PUT EXPONENT IN THE FAC
001162' 001000  000006                  16020           MVI     B,0             ;ZERO OVERFLOW BYTE
001163' 000000  000000
001164' 001000  000043                  16040           INX     H               ;POINT TO SIGN
001165' 001000  000066                  16060           MVI     M,200           ;ASSUME A POSITIVE NUMBER
001166' 000000  000200
```



```
001167' 001000  000027                  16080           RAL                     ;PUT SIGN IN CARRY
001170' 001000  000303                  16100           JMP     FADFLT          ;GO AND FLOAT THE NUMBER
001171' 000000  000143'
001172' 000000  001156'
                                        16120
                                        16140
                                        16160           ;ABSOLUTE VALUE OF FAC
                                        16180           ;ALTERS A,H,L
001173'                                 16200   ABS:
                                        16220   IFE     LENGTH=2,<
                                        16240           CPI     2               ;IS THE ARGUMENT AN INTEGER?
                                        16260           JZ      IABS>           ;YES, USE THE INTEGER ABSOLUTE VALUE
001173' 001000  000357                  16280           FSIGN                   ;GET THE SIGN OF FAC
001174' 001000  000360                  16300           RP                      ;ALL DONE IF IT IS POSITIVE
                                        16320                                   ;FALL INTO NEG
                                        16340
                                        16360           ;NEGATE NUMBER IN THE FAC
                                        16400           ;ALTERS A,H,L
                                        16420           ;NOTE: THE NUMBER MUST BE PACKED
001175' 001000  000041                  16440   NEG:    LXI     H,FAC+1         ;GET POINTER TO SIGN
001176' 777777  777777*
001177' 000000  001171'
001200' 001000  000176                  16460           MOV     A,M             ;GET SIGN
001201' 001000  000356                  16480           XRI     200             ;COMPLEMENT SIGN BIT
001202' 000000  000200
001203' 001000  000167                  16500           MOV     M,A             ;SAVE IT
001204' 001000  000311                  16520           RET                     ;ALL DONE
                                        16540
                                        16560
                                        16580   IFE     LENGTH=2,<
                                        16600           ;NEGATE ANY TYPE VALUE IN THE FAC
                                        16620           ;ALTERS A,B,C,D,E,H,L
                                        16640   VNEG:   LDA     VALTYP          ;SEE WHAT KIND OF NUMBER WE HAVE
                                        16660           CPI     2
                                        16680           JZ      INEG            ;WE HAVE AN INTEGER, NEGATE IT THAT WAY
                                        16700           JM      TMERR           ;BLOW UP ON STRINGS
                                        16720           JMP     NEG             ;NEGATE SNG AND DBL THE SAME
                                        16740
                                        16760
                                        16780           ;SGN FUNCTION
                                        16800           ;ALTERS A,H,L
                                        16820   SGN:    CALL    VSIGN           ;GET THE SIGN OF THE FAC IN A
                                        16840           MOV     L,A             ;PUT IT IN THE LO POSITION
                                        16860           RAL                     ;EXTEND THE SIGN TO THE HO
                                        16880           SBB     A
                                        16900           MOV     H,A
                                        16920           JMP     CONISS          ;RETURN THE RESULT AND SET VALTYP
                                        16940
                                        16960
                                        16980           ;GET THE SIGN OF THE VALUE IN THE FAC IN A
                                        17000           ;ASSUMES A HAS THE VALTYP WHEN CALLED
                                        17020           ;ALTERS A,H,L
```

```
                                17040   VSIGN:  CPI     2               ;IS THE ARGUMENT AN INTEGER?
                                17060           JNZ     SIGN            ;NO, SINGLE AND DOUBLE PREC, WORK THE SAME
                                17080           LHLD    FACLO           ;GET THE INTEGER ARGUMENT
                                17100           MOV     A,H             ;GET ITS SIGN
                                17120           ORA     L               ;CHECK IF THE NUMBER IS ZERO
                                17140           RZ                      ;IT IS, WE ARE DONE
                                17160           MOV     A,H             ;IT ISN'T, SIGN IS THE SIGN OF H
                                17180           JMP     ICOMPS>         ;GO SET A CORRECTLY
                                17200   PAGE
```



```
                                17220   SUBTTL  FLOATING POINT MOVEMENT ROUTINES
                                17240           ;PUT FAC ON STACK
                                17260           ;ALTERS D,E
001205' 001000  000353          17280   PUSHF:  XCHG                    ;SAVE (HL)
001206' 001000  000052          17300           LHLD    FACLO           ;GET LO'S
001207' 000000  000555*
001210' 000000  001176'
001211' 001000  000343          17320           XTHL                    ;SWITCH LO'S AND RET ADDR
001212' 001000  000345          17340           PUSH    H               ;PUT RET ADDR BACK ON STACK
001213' 001000  000052          17360           LHLD    FAC-1           ;GET MO'S
001214' 777777  777777*
001215' 000000  001207'
001216' 001000  000343          17380           XTHL                    ;SWITCH MO'S AND RET ADDR
001217' 001000  000345          17400           PUSH    H               ;PUT RET ADDR BACK ON STACK
001220' 001000  000353          17420           XCHG                    ;GET OLD (HL) BACK
001221' 001000  000311          17440           RET                     ;ALL DONE

                                17500           ;MOVE NUMBER FROM MEMORY [(HL)] TO FAC
                                17520           ;ALTERS B,C,D,E,H,L
                                17540           ;AT EXIT NUMBER IS IN B,C,D,E
                                17560           ;AT EXIT (HL):=(HL)+4
001222' 001000  000315          17580   MOVFM:  CALL    MOVRM           ;GET NUMBER IN REGISTERS
001223' 000000  001243'
001224' 000000  001214'
                                17600                                   ;FALL INTO MOVFR AND PUT IT IN FAC


                                17660           ;MOVE REGISTERS (B,C,D,E) TO FAC
                                17680           ;ALTERS D,E
001225' 001000  000353          17700   MOVFR:  XCHG                    ;GET LO'S IN (HL)
001226' 001000  000042          17720           SHLD    FACLO           ;PUT THEM WHERE THEY BELONG
001227' 000000  001207*
001230' 000000  001223'
001231' 001000  000140          17740           MOV     H,B             ;GET MO'S IN (HL)
001232' 001000  000151          17760           MOV     L,C
001233' 001000  000042          17780           SHLD    FAC-1           ;PUT MO'S WHERE THEY BELONG
001234' 777777  777777*
001235' 000000  001227'
001236' 001000  000353          17800           XCHG                    ;GET OLD (HL) BACK
001237' 001000  000311          17820           RET                     ;ALL DONE


                                17880           ;MOVE FAC TO REGISTERS (B,C,D,E)
                                17900           ;ALTERS B,C,D,E,H,L
001240' 001000  000041          17920   MOVRF:  LXI     H,FACLO         ;GET POINTER TO FAC
001241' 000000  001227*
001242' 000000  001234'
                                17940                                   ;FALL INTO MOVRM


                                18000           ;GET NUMBER IN REGISTERS (B,C,D,E) FROM MEMORY [(HL)]
                                18020           ;ALTERS B,C,D,E,H,L
```