```
SOURCE ACCESS NAME=      PPC2.P359.SRC.CNS
OBJECT ACCESS NAME=      PPC2.P359.OBJ.CNS359
LISTING ACCESS NAME=     PPC2.P359.LST.CNS359
ERROR ACCESS NAME=
OPTIONS=                 XREF
MACRO LIBRARY PATHNAME=


LINE      KEY   NAME

 0002      A    VERSION
                    =>PPC2.P359.SRC.P359
```

```
0031                    IDT  'CNS'
0032          ********************************************************
0033          *                                                      *
0034          *                CCCCC    N     N   SSSSS             *
0035          *               C     C   NN    N   S     S            *
0036          *               C         N N   N   S                  *
0037          *               C         N  N  N    SSSSS            *
0038          *               C         N   N N         S            *
0039          *               C     C   N    NN   S     S            *
0040          *                CCCCC    N     N   SSSSS             *
0041          *                                                      *
0042          *           PPPP      3333     555555     9999        *
0043          *           P   P    3    3    5         9    9       *
0044          *           P   P         3    5         9    9       *
0045          *           PPPP       3333    55555      99999      *
0046          *           P             3         5         9       *
0047          *           P         3   3    5    5    9    9       *
0048          *           P          3333     5555      9999        *
0049          *                                                      *
0050          ********************************************************
```

```
0055                    *
0056                    *       CONVERT THE NUMBER IN THE FAC TO A STRING.
0057                    *
0058                    *       CALL:    FAC  NUMBER
0059                    *                R0   0 FOR FREE FORMAT(R1 & R2 ARE IGNORED)
0060                    *                     BIT 0 ON FOR FIXED FORMAT
0061                    *                     BIT 1 ON FOR AN EXPLICIT SIGN
0062                    *                     BIT 2 ON TO OUTPUT THE SIGN OF A POS.
0063                    *                     NO. AS A PLUS SIGN('+') INSTEAD OF A
0064                    *                       SPACE (BIT 1 MUST ALSO BE ON)
0065                    *                     BIT 3 ON FOR E-NOTATION OUTPUT
0066                    *                      : BIT 4 ALSO ON FOR EXTENDED E-NOTATION
0067                    *                R1 AND R2 SPECIFY THE FIELD SIZE:
0068                    *                R1   NUMBER OF PLACES IN THE FIELD TO THE
0069                    *                     LEFT OF THE DECIMAL POINT INCLUDING AN
0070                    *                     EXPLICIT SIGN AND EXCLUDING THE DECIMAL
0071                    *                     POINT.
0072                    *                R2   NUMBER OF PLACES IN THE FIELD TO THE
0073                    *                     RIGHT OF THE DECIMAL POINT INCLUDING THE
0074                    *                     DECIMAL POINT.
0075                    *                R1 AND R2 EXCLUDE THE 4 POSITIONS FOR THE
0076                    *                     EXPONENT IF BIT 3 IS ON.
0077                    *           ERRORS: THE FIELD HAS MORE THAN 14 SIGNIFICANT DIGITS
0078                    *               IF THE NUMBER IS TOO BIG TO FIT IN THE FIELD, THE
0079                    *              FIELD IS FILLED WITH ASTERISKS.
0080                    *               THE ORIGINAL CONTENTS OF THE FAC ARE LOST.
0081                    *
0082                    *
0083                            DEF  CNS
0084                    *
0085                            REF  ROLOUT,ROLIN,FAC,FAC1,FAC8,FAC11,FAC12
0086                            REF  FAC15,FAC33,SIGN
0087                            REF  ROLB,R1LB,R4LB,R6LB,R12LB
0088                            REF  CBHA,ROUNUP,PAD,EXP
0089                            REF  WSM,WSM2,WSM4,WSM6,WSM8
0090                    *
0091                            ASMIF VERS=DX10
0092                    LWCNS   DATA  >0002
0093                            ASMELS
0094                            REF  LWCNS
0095                            ASMEND
0096 0000 0004          LWCNP   DATA  >0004
0097 0002 0008          LWCNE   DATA  >0008
0098 0004 0010          LWCNF   DATA  >0010
0099                    *
0100                    *   INTEGER POWER OF TEN TABLE
0101 0006 2710          CNSITT  DATA  10000
0102 0008 03E8                  DATA  1000
0103 000A   00          LW100   BYTE  0
0104 000B   64          LB100   BYTE  100
0105 000C   00          LW10    BYTE  0
0106 000D   0A          LB10    BYTE  10
0107 000E 0001                  DATA  1
0108 0010   20          LBSPC   BYTE  ' '
0109 0011   2A          LBAST   BYTE  '*'
0110 0012   2E          LBPER   BYTE  '.'
0111 0013   45          LBE     BYTE  'E'
0112 0014   30          LBZER   BYTE  '0'
0113 0016                   EVEN
```

```
0115 0016 C28B  CNS     MOV  R11,R10          IN ROLOUT:USE R10 TO RETURN
0116 0018 06A0          BL   @ROLOUT
     001A 0000
0117                *-----------------CONDITIONAL ASSEMBLY---------------------*
0118                    ASMIF VERS=DX10
0119
0120                    ASMELS
0121 001C
0122 001C 05C9          INCT R9
0123 001E C64D          MOV  R13,*R9
0124                    ASMEND
0125                *-----------------END OF CONDITIONAL ASSEMBLY-------------*
0126 0020 0206          LI   R6,FAC11         Optimize for space and speed
     0022 0000
0127 0024 D036          MOVB *R6+,R0          @FAC11=0 if free format output
0128 0026 0980          SRL  R0,8             Put in LSByte
0129 0028 D076          MOVB *R6+,R1          @FAC12 places to left of dec
0130 002A 0981          SRL  R1,8             Put in LSByte
0131 002C D0B6          MOVB *R6+,R2          @FAC13 places to right of dec
0132 002E 0982          SRL  R2,8             Put in LSByte
0133 0030 DDA0          MOVB @LBSPC,*R6+      Put extra space at beginning
     0032 0010'
0134                *                             for CNSCHK
0135 0034 0203          LI   R3,'-'*256       ASSUME NUMBER IS NEGATIVE
     0036 2D00
0136 0038 0760          ABS  @FAC             IS NUMBER NEGATIVE?
     003A 0000
0137 003C 1107          JLT  CNS01            YES-ITS SIGN IS KNOWN
0138 003E 0203          LI   R3,' '*256       NO-ASSUME A SPACE WILL BE USED
     0040 2000
0139 0042 2420          CZC  @LWCNP,R0        DO POSITIVE NUMBERS GET A PLUS
     0044 0000'
0140                *                         SIGN?
0141 0046 1302          JEQ  CNS01            NO-USE A SPACE
0142 0048 0203          LI   R3,'+'*256       YES-GET A PLUS SIGN
     004A 2B00
0143 004C DD83  CNS01   MOVB R3,*R6+          PUT SIGN IN BUFFER
0144 004E C800          MOV  R0,@WSM          IS FREE FORMAT OUTPUT SPECIFI
     0050 0000
0145 0052 1675          JNE  CNSX             NO-USE FIX FORMAT OUTPUT
```

```
0147                    *    FREE FORMAT FLOATING OUTPUT
0148 0054 C120               MOV  @FAC,R4          IS IT 0?
     0056 003A'
0149 0058 1611               JNE  CNSF1            NO
0150 005A 0606               DEC  R6
0151 005C 0204               LI   R4,'0'           YES-CONVERT TO A '0' AND QUIT
     005E 2030
0152 0060 DD84               MOVB R4,*R6+
0153 0062 DDA0               MOVB @R4LB,*R6+
     0064 0000
0154 0066 04C4               CLR  R4               PUT 0 AT END OF STRING
0155 0068 D584               MOVB R4,*R6
0156 006A 0204               LI   R4,>5902         PUT THE BEGINNING OF STRING
     006C 5902
0157                    *                          IN FAC+11,LENGTH IN FAC+12
0158                    *                          FAC15=59,LENGTH=2
0159 006E D804               MOVB R4,@FAC11
     0070 0022'
0160 0072 D820               MOVB @R4LB,@FAC12
     0074 0064'
     0076 0000
0161 0078 0460               B    @ROLIN           RT IN ROLIN
     007A 0000
0162 007C 06A0  CNSF1        BL   @CNSTEN          GET BASE TEN EXPONENT,IS NO.
     007E 02CA'
0163                    *                          LESS THAN ONE?
0164 0080 1112               JLT  CNSF02           YES-IT CANNOT BE PRINTED AS AN
0165                    *                          INTEGER
0166 0082 028D               CI   R13,9            NO-IS NUMBER TOO BIG TO PRINT
     0084 0009
0167                    *                          AS AN INTEGER?
0168 0086 150F               JGT  CNSF02           YES-ROUND NO. FOR E-NOTATION
0169                    *                          OUTPUT
0170 0088 D820               MOVB @FAC,@ROLB       NO-CHECK IF THE NUMBER IS AN
     008A 0056'
     008C 0000
0171                    *                          INTEGER,GET EXPONENT,HIGH BYTE
0172                    *                          IS STILL ZERO
0173 008E 0220               AI   R0,PAD           R0=PAD+FAC+2-64
     0090 0000
0174 0092 0220               AI   R0,>C            GET POINTER TO FIRST
     0094 000C
0175                    *                          FRACTIONAL BYTE
0176 0096 04C1  CNSF01       CLR  R1
0177 0098 D070               MOVB *R0+,R1          IS NEXT BYTE OF FRACTION ZERO?
0178 009A 1605               JNE  CNSF02           NO-PRINT NO. IN FIXED POINT
0179                    *                          FORMAT
0180 009C 0280               CI   R0,FAC8          YES-REACHED END OF NUMBER?
     009E 0000
0181 00A0 1AFA               JL   CNSF01           NO-CONTINUE LOOKING AT
0182                    *                          FRACTIONAL BYTES
0183 00A2 04CA               CLR  R10              YES-NUMBER IS AN INTEGER,
0184                    *                          SET INTEGER FLAG
0185 00A4 1011               JMP  CNSF05           GO PRINT THE NUMBER,NO ROUND-
0186                    *                          ING IS NECESSARY
0187 00A6 0201  CNSF02       LI   R1,5             ASSUME ROUNDING FOR E-NOTATION
     00A8 0005
0188 00AA 028D               CI   R13,9            IS NO. TOO BIG FOR FIXED POINT
     00AC 0009
0189                    *                          OUTPUT?
```

```
0190 00AE 1509          JGT   CNSF04          YES-ROUND FOR E-NOTATION
0191 00B0 028D          CI    R13,-4           NO-IS NUMBER TOO SMALL FOR
     00B2 FFFC
0192                *                          FIXED POINT OUTPUT?
0193 00B4 1106          JLT   CNSF04          YES-ROUND FOR ENOTATION OUTPU
0194 00B6 8C71          C     *R1+,*R1+        Force R1 to = 9
0195 00B8 028D          CI    R13,-2           NO-WILL NO. BE PRINTED WITH
     00BA FFFE
0196                *                          MAXIMUM NUMBER OF FIXED FORMA
0197                *                          SIGNIFICANT DIGITS?
0198 00BC 1502          JGT   CNSF04          YES-ROUND ACCORDINGLY
0199 00BE 0581          INC   R1              NO-ROUND NUMBER FOR MAXIMUM
0200                *                          SIGNIFICANT DIGITS (R1 = 10)
0201 00C0 A04D          A     R13,R1           THAT CAN BE PRINTED FOR THIS
0202                *                          NUMBER
0203 00C2
0204 00C2 06A0  CNSF04  BL    @CNSRND          ROUND NO. ACCORDINGLY-ROUNDIN
     00C4 0246'
0205                *                          CAN CHANGE THE EXPONENT AND
0206                *                          SO THE PRINT FORMAT TO BE USE
0207 00C6 070A          SETO  R10              SET NON-INTEGER FLAG
0208 00C8 028D  CNSF05  CI    R13,9            DECIDE WHICH PRINT FORMAT TO
     00CA 0009
0209 00CC 152B          JGT   CNSG             USE, TOO BIG FOR FIXED FORMAT
0210 00CE 028D          CI    R13,-6           USE E-NOTATION NUMBER IN RANG
     00D0 FFFA
0211                *                          FOR MAX.FIXED POINT DIGITS?
0212 00D2 1516          JGT   CNSF08          YES-USE FIXED FORMAT OUTPUT
0213 00D4 028D          CI    R13,-10          NO-NO. TOO SMALL FOR FIXED
     00D6 FFF6
0214                *                          FORMAT?
0215 00D8 1125          JLT   CNSG             YES-USE E-NOTATION OUTPUT
0216                *                          NO-THE NO. OF SIGNIFICANT
0217                *                          DIGITS WILL DETERMINE FIXED
0218                *                          FORMAT OUTPUT OR NOT
0219 00DA 0200          LI    R0,FAC8          GET POINTER TO LAST BYTE
     00DC 009E'
0220                *                          OF FAC+1
0221 00DE 04C1          CLR   R1               CLEAR LOW BYTE OF LEAST
0222                *                          SIGNIFICANT BYTE REGISTER
0223 00E0 0203          LI R3,4                4=15-11 GET NO. OF DIGITS+2-
     00E2 0004
0224                *                          EXPONENT SCALE FACTOR
0225 00E4 A0C7          A     R7,R3            TAKE INTO ACCOUNT A LEADING
0226                *                          ZERO IN FAC+1
0227 00E6 0643  CNSF06  DECT  R3               DECREMENT SIG DIGIT COUNT FOR
0228                *                          LAST ZERO BYTE
0229 00E8 0600          DEC   R0               POINT TO NEXT HIGHER BYTE OF
0230                *                          FAC
0231 00EA D050          MOVB  *R0,R1           IS NEXT BYTE ALL ZERO?
0232 00EC 13FC          JEQ   CNSF06           YES-CONTINUE LOOKING FOR LEAS
0233                *                          SIGNIFICANT BYTE
0234                *                          NO-FOUND THE LEAST SIGNIFICA
0235                *                          BYTE,THIS LOOP WILL ALWAYS
0236                *                          TERMINATE SINCE FAC+1 NEVER
0237 00EE 04C0          CLR   R0               TAKE INTO ACCOUNT IF THE LEAS
0238                *                          SIGNIFICANT BYTE IS DIVISIBLE
0239                *                          BY TEN
0240 00F0 06C1          SWPB  R1               IS DIVISIBLE BY TEN
0241 00F2 3C20          DIV   @LW10,R0         DIVIDE LEAST SIGNIFICANT BYTE
```

```
      00F4 000C'
0242                *
0243 00F6 C041              MOV   R1,R1           BY TEN
0244 00F8 1601              JNE   CNSF07          IS THE REMAINDER ZERO?
0245                *                             NO-SIGNIFICANT DIGIT COUNT IS
0246 00FA 0603              DEC   R3              CORRECT
0247                *                             YES-LEAST SIGNIFICANT BYTE HAS
0248 00FC 8343    CNSF07    C     R3,R13          A TRAILING ZERO
0249                *                             TOO MANY SIGNIFICANT DIGITS
0250 00FE 1512              JGT   CNSG            FOR FIXED FORMAT?
                                                  YES-USE E-NOTATION
```

```
0252                    *     FREE FORMAT FIXED POINT AND INTEGER FLOATING OUTPUT
0253 0100 6347  CNSF08  S     R7,R13            MAKE THE EXPONENT EVEN
0254 0102 110A          JLT   CNSF12            ARE THERE DIGITS TO LEFT OF
0255                    *                       DECIMAL POINT? JUMP IF NOT
0256                    *                       YES-PRINT DECIMAL POINT WITH
0257                    *                       THE NUMBER
0258 0104 0204          LI    R4,3              FIGURE OUT WHERE THE DECIMAL
     0106 0003
0259                    *                       POINT GOES IN
0260 0108 A10D          A     R13,R4            THE NUMBER'S DIGITS
0261 010A 0203  CNSF10  LI    R3,12             CONVERT THE MAXIMUM NUMBER OF
     010C 000C
0262                    *                       DECIMAL DIGITS, LEADING AND
0263                    *                       TRAILING ZEROS ARE SUPPRESSED
0264                    *                       LATER
0265 010E 06A0          BL    @CNSDIG           CONVERT NUMBER TO DECIMAL
     0110 02E6'
0266                    *                       DIGITS
0267 0112 06A0          BL    @CNSUTR           REMOVE TRAILING ZEROS
     0114 0408'
0268 0116 1011          JMP   CNSG01            SUPPRESS LEADING ZEROS AND
0269 0118
0270 0118 0700  CNSF12  SETO  R0                FIGURE OUT HOW MANY ZEROS
0271                    *                       THERE ARE
0272 011A 600D          S     R13,R0            BETWEEN DECIMAL POINT AND
0273                    *                       FIRST DIGIT
0274 011C 06A0          BL    @CNSPER           PUT DECIMAL POINT AND ZEROS
     011E 03B2'
0275                    *                       IN BUFFER
0276 0120 04C4          CLR   R4                DON'T PRINT ANOTHER DECIMAL
0277                    *                       POINT IN THE NUMBER
0278 0122 10F3          JMP   CNSF10            CONVERT NO. TO DECIMAL DIGITS
0279                    *                       FINISH UP
```

```
0281                    *    FREE FORMAT E-NOTATION FLOATING OUTPUT
0282 0124 0203  CNSG    LI   R3,8              GET MAXIMUM NO. OF DIGITS TO
     0126 0008
0283                    *                       PRINT
0284 0128 0204          LI   R4,3              FIGURE OUT WHERE TO PUT
     012A 0003
0285                    *                       DECIMAL POINT
0286 012C 6107          S    R7,R4             TAKE A LEADING ZERO INTO
0287                    *                       ACCOUNT
0288 012E 06A0          BL   @CNSDIG           CONVERT NO. TO DECIMAL DIGITS
     0130 02E6'
0289 0132 06A0          BL   @CNSUTR           SUPPRESS TRAILING ZEROS
     0134 0408'
0290 0136 06A0          BL   @CNSEXP           PUT EXPONENT INTO BUFFER
     0138 0330'
0291 013A 0460  CNSG01  B    @CNSMLS           SUPPRESS LEADING ZEROS AND
     013C 03C4'
0292                    *                       FINISH UP
```

```
0294                    *     FIXED FORMAT OUTPUT
0295                    *     WSM    R0 FORMAT SPECIFICATION
0296                    *     WSM2   R1 FORMAT SPECIFICATION
0297                    *     WSM4   R2 FORMAT SPECIFICATION
0298                    *     WSM6   NUMBER OF DIGIT PLACES TO LEFT OF DECIMAL POIN
0299                    *     WSM8   NUMBER OF DIGIT PLACES TO RIGHT OF DECIMAL POI
0300                    *
0301 013E C801  CNSX    MOV   R1,@WSM2      SAVE R1 FORMAT SPECIFICATION
     0140 0000
0302 0142 C802          MOV   R2,@WSM4      SAVE R2 FORMAT SPECIFICATION
     0144 0000
0303 0146 2420          CZC   @LWCNE,R0     IS E-NOTATION TO BE USED?
     0148 0002'
0304 014A 1606          JNE   CNSX01        YES-REMOVE PLACE FOR SIGN FROM LE
0305                    *                   OF DP COUNT
0306 014C 0283          CI    R3,'-'*256    NO-IS NUMBER NEGATIVE?
     014E 2D00
0307 0150 1303          JEQ   CNSX01        YES-REMOVE SIGN FROM DIGIT COUNT
0308 0152 2420          CZC   @LWCNS,R0     NO-IS AN EXPLICIT SIGN SPECIFIED?
     0154 0000
0309 0156 1306          JEQ   CNSX02        NO-DIGIT COUNT IS CORRECT AS IS
0310 0158 0601  CNSX01  DEC   R1            REMOVE PLACE FOR SIGN FROM LEFT O
0311                    *                   DP DIGIT COUNT
0312 015A 1504          JGT   CNSX02        ANY PLACES FOR DIGITS LEFT?
0313 015C 0283          CI    R3,'-'*256    NO-IS NUMBER NEGATIVE?
     015E 2D00
0314 0160 1301          JEQ   CNSX02        YES-CANNOT DO ANYTHING ABOUT IT
0315 0162 04C1          CLR   R1            NO-SEE IF NO DIGITS TO LEFT OF D
0316                    *                   WILL WORK
0317 0164 C801  CNSX02  MOV   R1,@WSM6      SAVE NUMBER OF DIGITS TO LEFT DP
     0166 0000
0318 0168 1110          JLT   CNSJ04        FIELD TOO SMALL IF THERE ARE
0319                    *                   NEGATIVE PLACES
0320 016A 0602          DEC   R2            TAKE DECIMAL POINT FROM RIGHT OF
0321                    *                   DP COUNT
0322 016C 1501          JGT   CNSX03        ARE THERE STILL PLACES LEFT?
0323 016E 04C2          CLR   R2            NO-DON'T PRINT ANY DIGITS THERE
0324 0170 C802  CNSX03  MOV   R2,@WSM8      SAVE RIGHT OF DP DIGIT COUNT
     0172 0000
0325 0174 C101          MOV   R1,R4         COMPUTE HOW MANY SIGNIFICANT DIGI
0326                    *                   ARE TO BE PRINTED
0327 0176 A102          A     R2,R4
0328 0178 1308          JEQ   CNSJ04        NONE-ERROR
0329                    *           FALL INTO NO-TO FIXED FORMAT FLOATING OUTPUT
```

```
0331                    *     FIXED FORMAT FLOATING OUTPUT
0332 017A 06A0                  BL    @CNSTEN       GET BASE TEN EXPONENT OF THE FAC
     017C 02CA'
0333 017E 2420                  CZC   @LWCNE,R0     IS E-FORMAT CALLED FOR?
     0180 0002'
0334 0182 1645                  JNE   CNSK          YES-GO DO IT
0335                    *     FIXED FORMAT FLOATING F-FORMAT OUTPUT
0336 0184 880D                  C     R13,@WSM6     ARE THERE TOO MANY DIGITS IN THE
     0186 0166'
0337                    *                           NUMBER FOR THE FIELD SIZE?
0338 0188 1102                  JLT   CNSJ00        NO - OK
0339 018A 0460  CNSJ04  B       @CNSAST
     018C 0440'
0340 018E C04D  CNSJ00  MOV     R13,R1              NO-GET EXPONENT
0341 0190 A042                  A     R2,R1         COMPUTE WHERE ROUNDING SHOULD TAKE
0342                    *                           PLACE
0343 0192 0281                  CI    R1,-1         IS THE NO. TOO SMALL FOR THE FIELD?
     0194 FFFF
0344 0196 112A                  JLT   CNSVZR        YES-RESULT IS ZERO
0345 0198 06A0                  BL    @CNSRND       NO-ROUND NO. TO THE PROPER PLACE
     019A 0246'
0346 019C 6347                  S     R7,R13        CONVERT EXPONENT TO AN EVEN NUMBER
0347 019E 110D                  JLT   CNSJ01        ANY DIGITS TO LEFT OF DP?
0348 01A0 0700                  SETO  R0            YES-COMPUTE HOW MANY ZERO ARE
0349                    *                           NEEDED BEFORE THE NUMBER TO FILL
0350                    *                           OUT THE FIELD  TO THE PROPER SIZE
0351 01A2 A020                  A     @WSM6,R0
     01A4 0186'
0352 01A6 600D                  S     R13,R0
0353 01A8 06A0                  BL    @CNSZER       PUT ZEROS IN THE BUFFER IF NEEDED
     01AA 03BC'
0354 01AC 0203                  LI    R3,3          COMPUTE THE NUMBER OF DIGITS TO
     01AE 0003
0355                    *                           CONVERT
0356 01B0 A0CD                  A     R13,R3        TAKE INTO ACCOUNT THE NUMBER'S SIZE
0357 01B2 C103                  MOV   R3,R4         YES-COMPUTE WHERE THE DP WILL GO
0358 01B4 A0E0                  A     @WSM8,R3      TAKE INTO ACCOUNT THE NO. OF
     01B6 0172'
0359                    *                           DECIMAL PLACES
0360 01B8 1011                  JMP   CNSJ02        GO CONVERT THE NUMBER
0361 01BA
0362 01BA C0E0  CNSJ01  MOV     @WSM8,R3            NUMBER IS LESS THAN ONE
     01BC 01B6'
0363 01BE 1316                  JEQ   CNSVZR        NO DECIMAL PLACES-PRINT ZERO
0364 01C0 C020                  MOV   @WSM6,R0      GET SIZE OF FIELD TO RIGHT OF DP
     01C2 01A4'
0365 01C4 0580                  INC   R0            ADD  ONE FOR CNSZER
0366 01C6 06A0                  BL    @CNSZER       FILL FIELD WITH ZEROS-THEY WILL BE
     01C8 03BC'
0367                    *                           SUPPRESSED
0368 01CA C306                  MOV   R6,R12        SAVE POINTER TO DP
0369 01CC 0700                  SETO  R0            COMPUTE NO. OF ZEROS AFTER DP
0370 01CE 600D                  S     R13,R0        AND BEFORE THE NUMBER
0371 01D0 06A0                  BL    @CNSPER       PUT THEM AND A DP INTO THE BUFFER
     01D2 03B2'
0372 01D4 A0CD                  A     R13,R3        FIGURE OUT HOW MANY DIGITS TO
0373                    *                           CONVERT
0374 01D6 0223                  AI    R3,3          SCALE ACCORDINGLY
     01D8 0003
0375 01DA 04C4                  CLR   R4            DONNOT PRINT A DECIMAL POINT
```

```
0376 01DC 06A0  CNSJ02 BL    @CNSDIG       CONVERT THE NO. TO DECIMAL DEGITS
     01DE 02E6'
0377 01E0 C020         MOV   @WSM4,R0      IS A DECIMAL POINT REQUIRED?
     01E2 0144'
0378 01E4 1601         JNE   CNSJ03        YES-IT IS ALREADY THERE
0379 01E6 D700         MOVB  R0,*R12       NO-OVERWRITE IT WITH ZERO
0380 01E8 0460  CNSJ03 B     @CNSCHK       GO FINISH UP
     01EA 041A'
```

```
0382                    *     FIXED FORMAT OUTPUT OF ZERO
0383 01EC C020  CNSVZR  MOV   @WSM6,R0      GET LEFT OF DP FIELD SIZE
     01EE 01C2'
0384 01F0 0580          INC   R0            ADJUST IT FOR CNSZER
0385 01F2 06A0          BL    @CNSZER       PUT IN THE CORRECT AMOUNT OF ZEROS
     01F4 03BC'
0386 01F6 C306          MOV   R6,R12        SAVE POINTER TO WHERE DP WILL GO
0387 01F8 C020          MOV   @WSM4,R0      IS A DP CALLED FOR?
     01FA 01E2'
0388 01FC 1302          JEQ   CNSV01        NO-DON'T PRINT ONE
0389 01FE 06A0          BL    @CNSPER       YES-PRINT IT AND SOME ZEROS AFTER
     0200 03B2'
0390                    *                   IT IF NEEDED
0391 0202 C020  CNSV01  MOV   @WSM,R0       GET R0 FORMAT SPECIFICATION
     0204 0050'
0392 0206 2420          CZC   @LWCNE,R0     IS E-FORMAT CALLED FOR?
     0208 0002'
0393 020A 13EE          JEQ   CNSJ03        NO-FINISH UP
0394 020C 1019          JMP   CNSK01        YES-PRINT AN EXPONENT
```

```
0396                    *     FIXED FORMAT FLOATING E-FORMAT OUTPUT
0397 020E C160  CNSK    MOV   @FAC,R5       IS IT ZERO?
     0210 008A'
0398 0212 1603          JNE   CNSK1         NO-GO TO CNSK1
0399 0214 04C7          CLR   R7            YES-DO IT DIFFERENTLY:R7,R13 SET T
0400 0216 04CD          CLR   R13           BE 0 AND JUMP TO CNSVZR
0401 0218 10E9          JMP   CNSVZR
0402 021A A042  CNSK1   A     R2,R1         GET TOTAL NUMBER OF DIGITS TO PRIN
0403 021C 0601          DEC   R1            COMPUTE WHERE ROUNDING SHOULD OCCU
0404 021E 06A0          BL    @CNSRND       ROUND NUMBER FOR E-FORMAT OUTPUT
     0220 0246'
0405 0222 C0E0          MOV   @WSM6,R3      GET NUMBER OF DIGITS TO LEFT OF DP
     0224 01EE'
0406 0226 6343          S     R3,R13        COMPUTE WHAT EXPONENT SHOULD BE
0407                    *                   PRINTED
0408 0228 058D          INC   R13           SCALE PROPERLY
0409 022A 60C7          S     R7,R3         CONSIDER ONLY EVEN EXPONENTS
0410 022C 05C3          INCT  R3            COMPUTE NUMBER OF DIGITS TO PRINT
0411                    *                   & WHERE TO PUT THE DECIMAL POINT.
0412 022E C103          MOV   R3,R4
0413 0230 A0E0          A     @WSM8,R3      TAKE DIGITS TO RIGHT OF DP INTO
     0232 01BC'
0414                    *                   ACCOUNT
0415 0234 06A0          BL    @CNSDIG       CONVERT NUMBER TO DECIMAL DIGITS
     0236 02E6'
0416 0238 C020          MOV   @WSM4,R0      IS A DECIMAL POINT NEEDED?
     023A 01FA'
0417 023C 1601          JNE   CNSK01        YES-LEAVE IT ALONE
0418 023E 0606          DEC   R6            NO-OVERWRITE IT WITH THE EXPONENT
0419 0240 06A0  CNSK01  BL    @CNSEXP       PUT THE EXPONENT INTO THE BUFFER
     0242 0330'
0420 0244 10D1          JMP   CNSJ03        FINISH UP AND ZERO SUPPRESS
```

```
0422                    *     ROUND THE NUMBER IN THE FAC
0423                    *     CALL   R1              NUMBER OF DECIMAL DIGITS TO RIGHT
0424                    *                            OF MOST SIGNIFICANT DIGIT TO ROUND
0425                    *                            TO
0426                    *            R13             BASE TEN EXPONENT
0427                    *            R7              0 IF R13 IS EVEN, 1 IF R13 IS ODD
0428                    *            BL              CNSRND
0429                    *            STATUS          BITS REFLECT EXPONENT
0430                    *            R13             BASE TEN EXPONENT OF ROUNDED RESULT
0431                    *            R7              0 IF R13 IS EVEN, 1 IF R13 IS ODD
0432                    *         DESTROYS: R0-R3,R12,R10
0433                    *         ASSUMES R12 GE -1
0434 0246 05C9  CNSRND  INCT R9                      SAVE RETURN ADDRESS
0435 0248 C64B          MOV  R11,*R9
0436 024A 6341          S    R1,R13                  COMPUTE BASE TEN EXPONENT OF PLACE
0437                    *                            TO ROUND TO
0438 024C 6047          S    R7,R1                   TAKE POSITION OF FIRST DIGIT INTO
0439                    *                            ACCOUNT
0440 024E 0811          SRA  R1,1                    COMPUTE ADDRESS IN FAC OF BYTE TO
0441                    *                            BE LOOKED AT
0442 0250 05C1          INCT R1                      TO DETERMINE IF ROUNDING OCCURS
0443 0252 0203          LI   R3,49*256               ASSUME 50 WILL BE ADDED TO THAT BYTE
     0254 3100
0444 0256 081D          SRA  R13,1                   ROUNDING TO AN EVEN TEN'S PLACE?
0445 0258 1702          JNC  CNSR01                  YES-ASSUMPTION WAS CORRECT
0446 025A 0203          LI   R3,4*256                NO-ADD 5 TO BYTE TO BE LOOKED AT
     025C 0400
0447 025E 0281  CNSR01  CI   R1,7                    IS ALL OF FAC SIGNIFICANT?
     0260 0007
0448 0262 1531          JGT  CNSR05                  YES-NO NEED TO ROUND
0449 0264 0207          LI   R7,FAC                  NO-GET POINTER INTO FAC
     0266 0210'
0450 0268 04CC          CLR  R12                     THE NUMBER IS POSITIVE
0451 026A D357          MOVB *R7,R13                 GET CURRENT FAC EXPONENT
0452 026C D28D          MOVB R13,R10                 SAVE IT TO SEE IF IT WILL CHANGE
0453 026E 098D          SRL  R13,8                   PUT EXPONENT IN THE LOW BYTE
0454 0270 A1C1          A    R1,R7                   GET ADDRESS OF BYTE TO LOOK AT
0455 0272 B5C3          AB   R3,*R7                  ADD NO. TO ADD TO ROUND-1 INTO
0456                    *                            CORRECT BYTE
0457 0274 C2C3          MOV R3,R11                   IN ROUNUP: CHANGE R3 VALUE
0458 0276 C10A          MOV R10,R4                   IN ROUNUP: USE R10 TO RETURN
0459 0278 020A          LI   R10,CNSROV
     027A 0290'
0460 027C D160          MOVB @FAC,R5                 IN ROUNUP: GET THE EXPONENT VALUE
     027E 0266'
0461                    *                            FROM EXP AND EXP+1, SO NOW PROVIDE
0462 0280 0985          SRL  R5,8
0463 0282 C805          MOV  R5,@EXP
     0284 0000
0464 0286 D805          MOVB R5,@SIGN                CLEAR SIGN WHICH IS USED IN ROUNUP
     0288 0000
0465 028A C149          MOV  R9,R5                   IN ROUNUP: R9 VALUE MAY BE CHANGED
0466 028C 0460          B    @ROUNUP                 PROPIGATE CARRY UPWARDS IN FAC
     028E 0000
0467 0290 C284  CNSROV  MOV  R4,R10
0468 0292
0469 0292 C0CB          MOV  R11,R3
0470 0294 C245          MOV  R5,R9
0471 0296 04C1          CLR  R1                      LABEL PREVENTS GETTING AN OVERFLOW
0472                    *                            WARNING
```

```
0473 0298 0287          CI    R7,FAC1       DID ROUNDING OCCUR AT FIRST BYTE
     029A 0000
0474                 *                      FAC?
0475 029C 1603          JNE   CNSR02        NO-GO CLEAR REST OF FAC
0476 029E 92A0          CB    @FAC,R10      YES-DID EXPONENT CHANGE?
     02A0 027E'
0477 02A2 160C          JNE   CNSR03        YES-FAC IS CORRECTLY ZEROED AS I
0478 02A4 0283  CNSR02  CI    R3,4*256      DID ROUNDING OCCUR ON A BYTE
     02A6 0400
0479                 *                      BOUNDARY?
0480 02A8 160A          JNE   CNSR04        YES-CLEAR REST OF BYTES IN FAC
0481 02AA 04C0          CLR   R0            NO-MAKE THIS DIGIT DIVISIBLE BY T
0482 02AC D817          MOVB  *R7,@R1LB     GET BYTE WHERE ROUNDING OCCURED
     02AE 0000
0483 02B0 3C20          DIV   @LW10,R0      DIVIDE BY TEN TO GET QUOTIENT
     02B2 000C'
0484 02B4 3820          MPY   @LW10,R0      PACK QUOTIENT BACK IN, IGNORE
     02B6 000C'
0485                 *                      REMAINDER
0486 02B8 D5E0          MOVB  @R1LB,*R7     PUT THE BYTE BACK INTO THE FAC
     02BA 02AE'
0487 02BC 0587  CNSR03  INC   R7            POINT TO NEXT BYTE OF FAC
0488 02BE DDC1  CNSR04  MOVB  R1,*R7+       ZERO NEXT BYTE OF FAC
0489 02C0 0287          CI    R7,FAC8       DONE ZEROING THE REST OF THE FAC?
     02C2 00DC'
0490 02C4 1AFC          JL    CNSR04        NO-CONTINUE TO DO IT
0491 02C6 C2D9  CNSR05  MOV   *R9,R11       YES-RESTORE RETURN ADDRESS
0492 02C8 0649          DECT  R9
0493                 *                      GET NEW BASE TEN EXPONENT OF FAC
```

```
0495                *     GET BASE TEN EXPONENT OF THE NUMBER IN THE FAC
0496                * CALL   BL      CNSTEN
0497                *       STATUS  STATUS BITS REFLECT EXPONENT
0498                *       R13     BASE TEN EXPONENT
0499                *       R7      0 IF R13 IS EVEN, 1 IF R13 IS ODD
0500                * DESTROYS: NONE
0501 02CA 020D     CNSTEN LI    R13,->4000    NEGATIVE BIAS
     02CC C000
0502 02CE B360           AB    @FAC,R13      GET BASE 1 HUNDRED EXPONENT OF FAC
     02D0 02A0'
0503 02D2 087D           SRA   R13,7         MULTIPLY IT BY TWO AND PUT IT IN
0504                *                        THE LOW BYTE
0505 02D4 04C7           CLR   R7            THE HIGH BIT OF FAC+1 IS ALWAYS OFF
0506 02D6 9820           CB    @FAC1,@CBHA   IS FIRST DIGIT OF FAC ONE DECIMAL
     02D8 029A'
     02DA 0000
0507                *                        DIGIT?
0508 02DC 1102           JLT   CNSTO1        YES-BASE TEN EXPONENT IS EVEN
0509 02DE 058D           INC   R13           NO-TAKE THIS INTO ACCOUNT IN BASE
0510                *                        TEN EXPONENT
0511 02E0 0587           INC   R7           THIS MAKES THE BASE TEN EXPONENT ODD
0512 02E2 C34D     CNSTO1 MOV   R13,R13      SET STATUS BITS TO REFLECT BASE TEN
0513                *                        EXPONENT
0514 02E4 045B           RT
```

```
0516                    *      CONVERT FRACTION OF FLOATING NUMBER IN THE FAC TO
0517                    *      ASCII DIGITS
0518                    * CALL R3   NUMBER OF DECIMAL DIGITS+1 TO CONVERT
0519                    *      R4   NUMBER OF THE DIGIT THE DECIMAL POINT IS TO
0520                    *           THE LEFT OF
0521                    *      R6   TEXT POINTER TO WHERE TO PUT RESULT
0522                    * BL CNSDIG
0523                    *      R3(MB) 0
0524                    *      R6   UPDATED TO POINT TO END OF DIGITS
0525                    *      R12  POINTER TO DECIMAL POINT
0526                    *  DESTROYS: R0-R2,R4
0527 02E6 05C9  CNSDIG INCT R9              SAVE RETURN ADDRESS
0528 02E8 C64B         MOV  R11,*R9
0529 02EA 04E0         CLR  @FAC8           CLEAR GUARD DIGITS IN CASE THEY ARE
     02EC 02C2'
0530                    *                   PRINTED
0531 02EE 04C1         CLR  R1              CLEAR HIGH BYTE OF CURRENT BYTE OF
0532                    *                   FAC REGISTER
0533 02F0 0202         LI   R2,FAC1         GET POINTER TO FIRST BYTE OF FAC
     02F2 02D8'
0534 02F4 06A0         BL   @CNSD03         CHECK FOR A LEADING DECIMAL POINT
     02F6 0314'
0535 02F8 04C0  CNSD01 CLR  R0              CLEAR HIGH WORD OF THIS BYTE OF FAC
0536                    *                   FOR DIV
0537 02FA D832         MOVB *R2+,@R1LB      GET NEXT BYTE OF FAC
     02FC 02BA'
0538 02FE 3C20         DIV  @LW10,R0        SEPARATE THE TWO DECIMAL DIGITS
     0300 000C'
0539 0302 06A0         BL   @CNSD02         PUT THE FIRST ONE IN THE BUFFER
     0304 030C'
0540 0306 C001         MOV  R1,R0           GET THE ONE'S PLACE DIGIT
0541 0308 020B         LI   R11,CNSD01      SET UP RETURN ADDRESS TO LOOP AND
     030A 02F8'
0542                    *                   GET THE NEXT BYTE OF THE FAC AFTER
0543                    *                   THIS DIGIT IS PRINTED
0544 030C 0220  CNSD02 AI   R0,'0'          CONVERT THIS DECIMAL DIGIT TO ASCII
     030E 0030
0545 0310 DDA0         MOVB @R0LB,*R6+      PUT THIS ASCII DIGIT INTO BUFFER
     0312 008C'
0546 0314 0604  CNSD03 DEC  R4              IS IT TIME FOR THE DECIMAL POINT?
0547 0316 1603         JNE  CNSD04          NO-CHECK FOR END OF NUMBER
0548 0318 C306         MOV  R6,R12          YES-SAVE POINTER TO DECIMAL POINT
0549 031A DDA0         MOVB @LBPER,*R6+     PUT DECIMAL POINT IN BUFFER
     031C 0012'
0550                    *      VSPTR (VALUE STACK PTR) AT CPU >6E, MAKE SURE NOT TO
0551                    *      DESTROY IT HERE
0552 031E 0286  CNSD04 CI   R6,FAC33        FIELD OVERFLOW ?
     0320 0000
0553 0322 1402         JHE  CNSD06          YES-PUT A ZERO BYTE AT THE END &RT
0554 0324 0603         DEC  R3              NO-HAVE ALL DIGITS BEEN PRINTED?
0555 0326 1503         JGT  CNSDRT          NO-RETURN AND PRINT THE NEXT DIGIT
0556 0328 D583  CNSD06 MOVB R3,*R6          YES-PUT A ZERO BYTE AT THE END OF
0557                    *                   THE NUMBER
0558 032A C2D9  CNSD05 MOV  *R9,R11         RESTORE RETURN ADDRESS
0559 032C 0649         DECT R9
0560 032E 045B  CNSDRT RT
```

```
0562                    *       PUT EXPONENT INTO THE BUFFER
0563                    *       CALL R6             TEXT POINTER INTO BUFFER
0564                    *            R13            EXPONENT
0565                    *            BL             CNSEXP
0566                    *            R6             UPDATED TO POINT AFTER EXPONENT
0567                    *       DESTROYS: R0,R13
0568 0330 05C9  CNSEXP  INCT R9                     SAVE RETURN ADDRESS
0569 0332 CE4B          MOV  R11,*R9+
0570 0334 C64C          MOV  R12,*R9                SAVE CONTENTS OF REGISTER
0571 0336 DDA0          MOVB @LBE,*R6+              PUT AN "E" INTO THE BUFFER
     0338 0013'
0572 033A 0200          LI   R0,'-'*256             ASSUME THE EXPONENT IS NEGATIVE
     033C 2D00
0573 033E 074D          ABS  R13                    IS EXPONENT NEGATIVE?
0574 0340 1102          JLT  CNSE01                 YES-SIGN IS CORRECT
0575 0342 0200          LI   R0,'+'*256             NO-GET SIGN FOR POSITIVE EXPONENT
     0344 2B00
0576 0346 DD80  CNSE01  MOVB R0,*R6+                PUT THE EXPONENT'S SIGN INTO BUFFER
0577 0348 028D          CI   R13,100                IS THE EXPONENT TOO BIG?
     034A 0064
0578 034C 110B          JLT  CNSE02                 NO-CONVERT IT TO ASCII
0579 034E C020          MOV  @WSM,R0                IS FREE FORMAT OUTPUT?
     0350 0204'
0580 0352 1303          JEQ  CNSE04                 YES-GET THE  ASTERISK
0581 0354 2420          CZC  @LWCNF,R0              NO-IS EXTENDED EXPONENT SPECIFIED?
     0356 0004'
0582 0358 1605          JNE  CNSE02                 YES-CONVERT IT TO ASCII
0583 035A 0200  CNSE04  LI   R0,'*'*256             NO-GET AN ASTERISK
     035C 2A00
0584 035E DD80          MOVB R0,*R6+                PUT TWO ASTERISKS IN THE BUFFER FOR
0585                    *                           THE EXPONENT
0586 0360 DD80          MOVB R0,*R6+                BECAUSE IT IS TOO BIG
0587 0362 1015          JMP  CNSE03                 GO FINISH UP
0588 0364 06A0  CNSE02  BL   @CNSINT                CONVERT THE EXPONENT TO ASCII DIGIT
     0366 0398'
0589 0368 0226          AI   R6,-5                  POINT BACK TO START OF EXPONENT
     036A FFFB
0590 036C C020          MOV  @WSM,R0                IS FREE FORMAT OUTPUT?
     036E 0350'
0591 0370 130A          JEQ  CNSE05                 YES
0592 0372 2420          CZC  @LWCNF,R0              NO-IS EXTENDED EXPONENT SPECIFIED?
     0374 0004'
0593 0376 1307          JEQ  CNSE05                 NO
0594 0378 DDA6          MOVB @2(R6),*R6+            YES-MOVE 3(INSTEAD OF 2)SIGNIFICANT
     037A 0002
0595 037C DDA6          MOVB @2(R6),*R6+            DIGITS OF EXPONENT UP PASS THE
     037E 0002
0596 0380 DDA6          MOVB @2(R6),*R6+            LEADING ZEROS  FROM CNSINT
     0382 0002
0597 0384 1004          JMP  CNSE03
0598 0386 DDA6  CNSE05  MOVB @3(R6),*R6+            MOVE SIGNIFICANT DIGITS OF EXPONENT
     0388 0003
0599                    *                           UP PASS THE LEADING ZEROS FROM
0600 038A DDA6          MOVB @3(R6),*R6+            CNSINT
     038C 0003
0601 038E D5A0  CNSE03  MOVB @LW10,*R6              PUT A ZERO BYTE AT THE END OF THE
     0390 000C'
0602                    *                           NUMBER
0603 0392 C319          MOV  *R9,R12                RESTORE ORIGINAL CONTENTS OF R12
0604 0394 0649          DECT R9
```

```
0605 0396 10C9          JMP   CNSD05        POP ADDR AND RETURN
```

```
0607                    *CONVERT AN UNSIGNED INTEGER INTO A STRING OF 5 ASCII DIGITS
0608                    * CALL R6   TEXT POINTER
0609                    *      R13  INTEGER
0610                    *      BL   CNSINT
0611                    *      R6   UPDATED TO POINT AFTER NUMBER
0612                    * DESTROYS: R0,R12,R13
0613                    *
0614 0398 0200  CNSINT LI   R0,CNSITT     GET POINTER TO INTEGER POWER OF TEN
     039A 0006'
0615                    *                            TABLE
0616 039C 04CC  CNSI01 CLR  R12           CLEAR HIGH WORD OF INTEGER FOR DIV
0617 039E 3F30         DIV  *R0+,R12      DIVIDE BY NEXT POWER OF TEN
0618 03A0 022C         AI   R12,'0'       CONVERT QUOTIENT TO ASCII
     03A2 0030
0619 03A4 DDA0         MOVB @R12LB,*R6+   PUT NEXT DIGIT INTO THE BUFFER
     03A6 0000
0620 03A8 0280         CI   R0,CNSITT+10  DIVIDED BY ALL THE POWERS OF TEN?
     03AA 0010'
0621 03AC 1AF7         JL   CNSI01        NO-COMPUTE THE NEXT DIGIT OF THE NO
0622 03AE D58C         MOVB R12,*R6       YES-PUT A ZERO BYTE AT THE END OF
0623                    *                            THE NUMBER
0624 03B0 045B         RT
```

```
0626                    *   PUT SOME ZEROS IN THE BUFFER AND MAYBE A DECIMAL POINT
0627                    *   CALL    R0              NUMBER OF ZEROS+1
0628                    *           R6              TEXT POINTER INTO BUFFER
0629                    *           BL  CNSPER:     TO PUT IN A DECIMAL POINT BEFORE
0630                    *                           ZEROS
0631                    *           BL  CNSZER:     TO JUST PUT IN ZEROS
0632                    *           R6              UPDATED TO POINT AFTER THE ZEROS
0633                    *   DESTROYS:  R0
0634 03B2 DDA0  CNSPER MOVB @LBPER,*R6+  PUT A DECIMAL POINT IN THE BUFFER
     03B4 0012'
0635 03B6 1002         JMP  CNSZER       THEN SOME ZEROS
0636 03B8 DDA0  CNSZ01 MOVB @LBZER,*R6+  PUT A ZERO IN THE BUFFER
     03BA 0014'
0637 03BC 0600  CNSZER DEC  R0           ARE THERE MORE ZEROS TO PUT IN?
0638 03BE 15FC         JGT  CNSZ01       YES-GO PUT IN ANOTHER ZERO
0639 03C0 D580         MOVB R0,*R6       NO-PUT A NULL BYTE AFTER THE ZEROS
0640 03C2 045B         RT
```

```
0642                    *  SUPPRESS LEADING ZEROS AND FLOAT THE SIGN
0643                    *  CALL      JMP  CNSMLS:  ENTRY TO FINISH UP AFTER ZERO
0644                    *                          SUPPRESSING
0645                    *            BL   CNSLEA:  ENTRY TO RETURN AFTERWARDS
0646                    *            R1            ASCII SIGN IN HIGH BYTE
0647                    *            R6            POINTER TO START OF NUMBER
0648                    *    DESTROYS:  R0-R1
0649 03C4 020B  CNSMLS LI   R11,CNSSTR   ENTRY TO FINISH UP NUMBER AFTERWARD
     03C6 046A'
0650 03C8 0206  CNSLEA LI   R6,FAC15     GET POINTER TO SIGN
     03CA 0000
0651 03CC D056         MOVB *R6,R1       GET SIGN
0652 03CE DDA0  CNSL01 MOVB @LBSPC,*R6+  PUT A SPACE WHERE THE ZERO OR SIGN
     03D0 0010'
0653                    *                 WAS
0654 03D2 9816         CB   *R6,@LBZER   IS THE NEXT BYTE ZERO?
     03D4 0014'
0655 03D6 13FB         JEQ  CNSL01       YES-SUPPRESS IT
0656 03D8 D016         MOVB *R6,R0       NO-IS THIS THE END OF THE NUMBER?
0657 03DA 130F         JEQ  CNSL02       YES-PUT THE ZERO BACK, THE NUMBER IS
0658                    *                 ZERO
0659 03DC 9800         CB   R0,@LBE      NO-IS THIS THE START OF THE
     03DE 0013'
0660                    *                 EXPONENT?
0661 03E0 130C         JEQ  CNSL02       YES-PUT THE ZERO BACK IN, NO. IS 0
0662 03E2 9800         CB   R0,@LBPER    NO-IS THIS THE DECIMAL POINT?
     03E4 0012'
0663 03E6 160C         JNE  CNSL03       NO-PUT THE SIGN BACK IN
0664 03E8 C020         MOV  @WSM,R0      YES-IS FREE FORMAT OUTPUT?
     03EA 036E'
0665 03EC 1609         JNE  CNSL03       NO-THEN PUT THE SIGN BACK IN FIX
0666                    *                 FORMAT OUTPUT
0667 03EE D026         MOVB @1(R6),R0    YES-ANY DIGITS TO RIGHT OF DP ?
     03F0 0001
0668 03F2 1303         JEQ  CNSL02       NO-END OF NO.-PUT ZERO BACK
0669 03F4 9800         CB   R0,@LBE      DOES EXPONENT START AFTER DP ?
     03F6 0013'
0670 03F8 1603         JNE  CNSL03       NO-PUT THE SIGN BACK
0671 03FA 0606  CNSL02 DEC  R6           YES-POINT BACK TO WHERE THE ZERO
0672                    *                 WAS
0673 03FC D5A0         MOVB @LBZER,*R6   PUT THE ZERO BACK IN, THE NO. IS 0
     03FE 0014'
0674 0400 0606  CNSL03 DEC  R6           POINT BACK TO WHERE THE SIGN WILL
0675                    *                 GO
0676 0402 D581         MOVB R1,*R6       PUT THE SIGN BACK IN THE BUFFER
0677 0404 045B         RT
```

```
0679                    *    REMOVE TRAILING ZEROS
0680                    *    CALL     R3              0
0681                    *             R6              POINTER TO ONE PAST END OF NUMBEF
0682                    *             R12             POINTER TO DECIMAL POINT
0683                    *             R10             ZERO IF AN INTEGER IS BEING PRINTI
0684                    *             BL              CNSUTR
0685                    *             R6              POINTER TO NEW END OF NUMBER
0686                    *    DESTROYS:   NONE
0687 0406 0606          CNSU01 DEC    R6              POINT BACK TO NEXT DIGIT IN THE NU
0688 0408 9826          CNSUTR CB     @-1(R6),@LBZER     IS THE LAST DIGIT IN THE NO. (
     040A FFFF
     040C 0014'
0689 040E 13FB                 JEQ    CNSU01          YES-LOOK BACK FOR A NON-ZERO DIGIT
0690 0410 C28A                 MOV    R10,R10         NO-IS AN INTEGER BEING PRINTED?
0691 0412 1601                 JNE    CNSU02          NO-PUT A NULL AT THE END OF THE NU
0692 0414 C18C                 MOV    R12,R6          YES-END OF NUMBER IS WHERE DP IS
0693                    *                              ALL DIGITS AFTER THE DECIMAL POINT
0694                    *                              SHOULD BE ZERO
0695 0416 D583          CNSU02 MOVB   R3,*R6          PUT A ZERO BYTE AT THE END OF THE
0696                    *                              NUMBER
0697 0418 045B                 RT
```

```
0699                    *  SET UP A POINTER TO THE BEGINNING OF A FIXED FORMAT FIELD
0700                    *  AND SEE IF THE FIELD IS LARGE ENOUGH AND FINISH UP
0701                    *  CALL  R12  POINTER TO DECIMAL POINT OR WHERE IT WOULD GO
0702                    *        JMP  CNSCHK
0703                    *        R6   POINTER TO BEGINNIGN OF NUMBER
0704                    *  DESTROYS: R0,R1
0705 041A 06A0  CNSCHK BL   @CNSLEA       SUPPRESS LEADING ZEROS AND FIX UP
     041C 03C8'
0706                    *                              THE SIGN
0707 041E C18C          MOV  R12,R6        POINT TO DECIMAL POINT
0708 0420 61A0          S    @WSM2,R6      POINT TO WHERE THE BEGINNING OF THE
     0422 0140'
0709                    *                              FIELD IS
0710 0424 9826          CB   @-1(R6),@LBSPC    DOES NUMBER EXTEND BEFORE THE
     0426 FFFF
     0428 0010'
0711                    *                                   FIELD BEGINNING?
0712 042A 160A          JNE  CNSAST        YES-ERROR
0713 042C C020          MOV  @WSM,R0       NO-GET R0 FORMAT SPECIFICATION
     042E 03EA'
0714 0430 2420          CZC  @LWCNS,R0     IS AN EXPLICIT SIGN REQUIRED?
     0432 0154'
0715 0434 131A          JEQ  CNSSTR        NO-FINISH UP AND RETURN
0716 0436 9816          CB   *R6,@LBSPC    YES-IS FIRST CHAR. OF NUMBER A
     0438 0010'
0717                    *                              SPACE?
0718 043A 1317          JEQ  CNSSTR        YES-FINISH UP AND RETRUN
0719 043C 9056          CB   *R6,R1        NO-IS FIRST CHAR.OF NO. THE SIGN?
0720 043E 1315          JEQ  CNSSTR        YES-FINISH UP AND RETURN
0721                    *                              NO-ERROR
```

```
0723                    *  ASTERISK FILL A FIXED FORMAT FIELD AND FINISH UP
0724                    *  CALL      JMP           CNSAST
0725                    *            R6            POINTER TO THE BEGINNING OF THE
0726                    *                          STRING
0727                    *    DESTROYS: R0, R1
0728 0440 0206  CNSAST LI    R6, WSM               Optimize for speed and space
     0442 042E'
0729 0444 C036         MOV   *R6+, R0              GET R0 FORMAT SPECIFICATION
0730 0446 C076         MOV   *R6+, R1              GET LEFT OF DECIMAL POINT SIZE
0731 0448 A076         A     *R6+, R1              COMPUTE LENGTH OF FIELD
0732 044A 2420         CZC   @LWCNE, R0            IS E-FORMAT BEING USED?
     044C 0002'
0733 044E 1305         JEQ   CNSA01                NO-FIELD LENGTH IS CORRECT
0734 0450 8C71         C     *R1+, *R1+            YES-INCREASE FIELD LENGTH FOR THE
0735                    *                          EXPONENT (Increments R1 by 4)
0736 0452 2420         CZC   @LWCNF, R0            IS EXTENDED E-FORMAT BEING USED?
     0454 0004'
0737 0456 1301         JEQ   CNSA01                NO-FIELD LENGTH IS CORRECT
0738 0458 0581         INC   R1                    YES-INCREASE FIELD LENGTH FOR THE
0739                    *                          EXPONENT (Increments R1 by 1)
0740 045A 0206  CNSA01 LI    R6, FAC15             GET POINTER TO BEGINNING OF BUFFER
     045C 03CA'
0741 045E C006         MOV   R6, R0                GET A POINTER TO PUT ASTERISKS IN
0742                    *                          THE BUFFER
0743 0460 DC20  CNSA02 MOVB  @LBAST, *R0+          PUT AN ASTERISK INTO THE BUFFER
     0462 0011'
0744 0464 0601         DEC   R1                    IS THE FIELD FILLED YET?
0745 0466 15FC         JGT   CNSA02                NO-CONTINUE ASTERISK FILLING IT
0746 0468 D401         MOVB  R1, *R0               YES-PUT A ZERO BYTE AT THE END OF
0747                    *                          STRING
0748                    *                          FINISH UP AND RETURN
```

```
0750                    *  FINISH UP  -- COMPUTE THE LENGTH OF THE STRING AND RETURN
0751                    *  CALL     R6            POINTER TO FIRST CHARACTER IN THE
0752                    *                         STRING, THE STRING ENDS WITH A ZERO
0753                    *                         BYTE
0754                    *           JMP           CNSSTR
0755                    *      DESTROYS:  R0-R1
0756                    *
0757 046A C006          CNSSTR MOV  R6,R0         GET POINTER TO BEGINNING OF THE
0758                    *                         STRING
0759 046C D070          CNSS01 MOVB *R0+,R1       LOOK FOR END OF STRING-FOUND IT?
0760 046E 16FE                 JNE  CNSS01        NO-KEEP LOOKING
0761 0470 0600                 DEC  R0            YES-POINT TO BACK TO THE ZERO BYTE
0762 0472 6006                 S    R6,R0         COMPUTE LENGTH OF STRING
0763 0474 D820                 MOVB @R0LB,@FAC12  PUT LENGTH OF STRING IN FAC+12
     0476 0312'
     0478 0076'
0764 047A 0200                 LI   R0,PAD
     047C 0090'
0765 047E 6180                 S    R0,R6         PUT BEGINNING OF STRING IN FAC+11
0766 0480 D820                 MOVB @R6LB,@FAC11
     0482 0000
     0484 0070'
0767                    *------------------CONDITIONAL ASSEMBLY--------------------*
0768                           ASMIF VERS=DX10
0769
0770                           ASMELS
0771 0486
0772 0486 C359                 MOV  *R9,R13      RESTORE GROM ADDRESS
0773 0488 0649                 DECT R9           OFF THE STACK
0774                           ASMEND
0775                    *---------------END OF CONDITIONAL ASSEMBLY----------------*
0776 048A 0460                 B    @ROLIN         IN ROLIN RETURN
     048C 007A'
0777                           END
NO ERRORS,      NO WARNINGS
```

CBHA	R	02DA'	0088	0506				
CNS	D	0016'	0115	0083				
CNSO1		004C'	0143	0137	0141			
CNSA01		045A'	0740	0733	0737			
CNSA02		0460'	0743	0745				
CNSAST		0440'	0728	0339	0712			
CNSCHK		041A'	0705	0380				
CNSD01		02F8'	0535	0541				
CNSD02		030C'	0544	0539				
CNSD03		0314'	0546	0534				
CNSD04		031E'	0552	0547				
CNSD05		032A'	0558	0605				
CNSD06		0328'	0556	0553				
CNSDIG		02E6'	0527	0265	0288	0376	0415	
CNSDRT		032E'	0560	0555				
CNSE01		0346'	0576	0574				
CNSE02		0364'	0588	0578	0582			
CNSE03		038E'	0601	0587	0597			
CNSE04		035A'	0583	0580				
CNSE05		0386'	0598	0591	0593			
CNSEXP		0330'	0568	0290	0419			
CNSF01		0096'	0176	0181				
CNSF02		00A6'	0187	0164	0168	0178		
CNSF04		00C2'	0204	0190	0193	0198		
CNSF05		00C8'	0208	0185				
CNSF06		00E6'	0227	0232				
CNSF07		00FC'	0248	0244				
CNSF08		0100'	0253	0212				
CNSF1		007C'	0162	0149				
CNSF10		010A'	0261	0278				
CNSF12		0118'	0270	0254				
CNSG		0124'	0282	0209	0215	0250		
CNSG01		013A'	0291	0268				
CNSI01		039C'	0616	0621				
CNSINT		0398'	0614	0588				
CNSITT		0006'	0101	0614	0620			
CNSJ00		018E'	0340	0338				
CNSJ01		01BA'	0362	0347				
CNSJ02		01DC'	0376	0360				
CNSJ03		01E8'	0380	0378	0393	0420		
CNSJ04		018A'	0339	0318	0328			
CNSK		020E'	0397	0334				
CNSK01		0240'	0419	0394	0417			
CNSK1		021A'	0402	0398				
CNSL01		03CE'	0652	0655				
CNSL02		03FA'	0671	0657	0661	0668		
CNSL03		0400'	0674	0663	0665	0670		
CNSLEA		03C8'	0650	0705				
CNSMLS		03C4'	0649	0291				
CNSPER		03B2'	0634	0274	0371	0389		
CNSR01		025E'	0447	0445				
CNSR02		02A4'	0478	0475				
CNSR03		02BC'	0487	0477				
CNSR04		02BE'	0488	0480	0490			
CNSR05		02C6'	0491	0448				
CNSRND		0246'	0434	0204	0345	0404		
CNSROV		0290'	0467	0459				
CNSS01		046C'	0759	0760				
CNSSTR		046A'	0757	0649	0715	0718	0720	
CNST01		02E2'	0512	0508				

```
CNSTEN      02CA'   0501  0162  0332
CNSU01      0406'   0687  0689
CNSU02      0416'   0695  0691
CNSUTR      0408'   0688  0267  0289
CNSV01      0202'   0391  0388
CNSVZR      01EC'   0383  0344  0363  0401
CNSX        013E'   0301  0145
CNSX01      0158'   0310  0304  0307
CNSX02      0164'   0317  0309  0312  0314
CNSX03      0170'   0324  0322
CNSZ01      03B8'   0636  0638
CNSZER      03BC'   0637  0353  0366  0385  0635
DX10        0001    0003  0004  0091  0118  0768
EXP       R 0284'   0088  0463
FAC       R 02D0'   0085  0136  0148  0170  0397  0449  0460  0476  0502
FAC1      R 02F2'   0085  0473  0506  0533
FAC11     R 0484'   0085  0126  0159  0766
FAC12     R 0478'   0085  0160  0763
FAC15     R 045C'   0086  0650  0740
FAC33     R 0320'   0086  0552
FAC8      R 02EC'   0085  0180  0219  0489  0529
LB10        000D'   0106
LB100       000B'   0104
LBAST       0011'   0109  0743
LBE         0013'   0111  0571  0659  0669
LBPER       0012'   0110  0549  0634  0662
LBSPC       0010'   0108  0133  0652  0710  0716
LBZER       0014'   0112  0636  0654  0673  0688
LW10        000C'   0105  0241  0483  0484  0538  0601
LW100       000A'   0103
LWCNE       0002'   0097  0303  0333  0392  0732
LWCNF       0004'   0098  0581  0592  0736
LWCNP       0000'   0096  0139
LWCNS     R 0432'   0094  0308  0714
P359        0000    0003  0003
PAD       R 047C'   0088  0173  0764
R0          0000          0127  0128  0139  0144  0173  0174  0177  0180  0219
                          0229  0231  0237  0241  0270  0272  0303  0308  0333
                          0348  0351  0352  0364  0365  0369  0370  0377  0379
                          0383  0384  0387  0391  0392  0416  0481  0483  0484
                          0535  0538  0540  0544  0572  0575  0576  0579  0581
                          0583  0584  0586  0590  0592  0614  0617  0620  0637
                          0639  0656  0659  0662  0664  0667  0669  0713  0714
                          0729  0732  0736  0741  0743  0746  0757  0759  0761
                          0762  0764  0765
ROLB      R 0476'   0087  0170  0545  0763
R1          0001          0129  0130  0176  0177  0187  0194  0194  0199  0201
                          0221  0231  0240  0243  0243  0301  0310  0315  0317
                          0325  0340  0341  0343  0402  0403  0436  0438  0440
                          0442  0447  0454  0471  0488  0531  0540  0651  0676
                          0719  0730  0731  0734  0734  0738  0744  0746  0759
R10         000A          0115  0183  0207  0452  0458  0459  0467  0476  0690
                          0690
R11         000B          0115  0435  0457  0469  0491  0528  0541  0558  0569
                          0649
R12         000C          0368  0379  0386  0450  0548  0570  0603  0616  0617
                          0618  0622  0692  0707
R12LB     R 03A6'   0087  0619
R13         000D          0123  0166  0188  0191  0195  0201  0208  0210  0213
                          0248  0253  0260  0272  0336  0340  0346  0352  0356
```

LABEL		VALUE	DEFN	REFERENCES								
				0370	0372	0400	0406	0408	0436	0444	0451	0452
				0453	0501	0502	0503	0509	0512	0512	0573	057
				0772								
R1LB	R	02FC'	0087	0482	0486	0537						
R2		0002		0131	0132	0302	0320	0323	0324	0327	0341	0402
				0533	0537							
R3		0003		0135	0138	0142	0143	0223	0225	0227	0246	0248
				0261	0282	0306	0313	0354	0356	0357	0358	0362
				0372	0374	0405	0406	0409	0410	0412	0413	0443
				0446	0455	0457	0469	0478	0554	0556	0695	
R4		0004		0148	0151	0152	0154	0155	0156	0159	0258	0260
				0276	0284	0286	0325	0327	0357	0375	0412	0458
				0467	0546							
R4LB	R	0074'	0087	0153	0160							
R5		0005		0397	0460	0462	0463	0464	0465	0470		
R6		0006		0126	0127	0129	0131	0133	0143	0150	0152	0153
				0155	0368	0386	0418	0545	0548	0549	0552	0556
				0571	0576	0584	0586	0589	0594	0594	0595	0595
				0596	0596	0598	0598	0600	0600	0601	0619	0622
				0634	0636	0639	0650	0651	0652	0654	0656	0667
				0671	0673	0674	0676	0687	0688	0692	0695	0707
				0708	0710	0716	0719	0728	0729	0730	0731	0740
				0741	0757	0762	0765					
R6LB	R	0482'	0087	0766								
R7		0007		0225	0253	0286	0346	0399	0409	0438	0449	0451
				0454	0455	0473	0482	0486	0487	0488	0489	0505
				0511								
R9		0009		0122	0123	0434	0435	0465	0470	0491	0492	05
				0528	0558	0559	0568	0569	0570	0603	0604	07
				0773								
ROLIN	R	048C'	0085	0161	0776							
ROLOUT	R	001A'	0085	0116								
ROUNUP	R	028E'	0088	0466								
SIGN	R	0288'	0086	0464								
VERMAC	M		A0001	0003								
VERS		0000	0003	0004	0091	0118	0768					
WSM	R	0442'	0089	0144	0391	0579	0590	0664	0713	0728		
WSM2	R	0422'	0089	0301	0708							
WSM4	R	023A'	0089	0302	0377	0387	0416					
WSM6	R	0224'	0089	0317	0336	0351	0364	0383	0405			
WSM8	R	0232'	0089	0324	0358	0362	0413					