```
0001  * NAME BASIC-MTHPAK          DOC. 70181832000        REV. A              BS100010
0002  *                                                                        BS100020
0003  *                                                                        BS100030
0004  * DESCRIPTION BASIC-16 ARITHMETIC, CONVERSION, LIBRARY FUNCTION ROUTINES BS100040
0005  *                                                                        BS100050
0006  *                                                                        BS100060
0007  * REVISION HISTORY                                                      BS100070
0008  *    REV.       DATE          ECO NO.                                   BS100080
0009  *     A                       RELEASED                                  BS100090
0010  *                                                                       BS100100
0011  *                                                                       BS100110
0012  *                                                                       BS100120
0013  *                                                                       BS100130
0014  *                                                                       BS100140
0015  * DOCUMENTATION REFERENCES                                              BS100150
0016  *    TITLE                                DOC. NO.                      BS100160
0017  *    BASIC-16 USER'S MANUAL               70130072543                   BS100170
0018  *                                                                       BS100180
0019  *                                                                       BS100190
0020  *                                                                       BS100200
0021  *                                                                       BS100210
0022  *                                                                       BS100220
0023  *                                                                       BS100230
0024  * COPYRIGHT 1971 BY HONEYWELL INFORMATION SYSTEMS INC., COMPUTER        BS100240
0025  * SYSTEMS DIVISION, FRAMINGHAM, MASSACHUSETTS.  CONTENTS OF THIS       BS100250
0026  * PUBLICATION MAY NOT BE REPRODUCED IN ANY FORM, IN WHOLE OR IN        BS100260
0027  * PART, WITHOUT PERMISSION OF THE COPYRIGHT OWNER.  ALL RIGHTS         BS100270
0028  * RESERVED.                                                           BS100280
0029  ************************************************************************BS100290
0030       EJCT                                                             BS100300
```

```
0031                *                                                                   BSI00310
0032                *                                                                   BSI00320
0033                     SUBR   L$22        DOUBLE WORD LOAD                          BSI00330
0034                     SUBR   H$22        DOUBLE WORD STORE                         BSI00340
0035                     SUBR   N$22        TWO'S COMPLIMENT A FLOATING POINT NUMBER  BSI00350
0036                     SUBR   S$22        FLOATING POINT SUBTRACTION                BSI00360
0037                     SUBR   A$22        FLOATING POINT ADDITION                   BSI00370
0038                     SUBR   D$22        FLOATING POINT DIVISION                   BSI00380
0039                     SUBR   M$22        FLOATING POINT MULTIPLICATION             BSI00390
0040                     SUBR   E$22        FLOATING POINT EXPONENTIATION             BSI00400
0041                     SUBR   M$11        INTEGER MULTIPLY                          BSI00410
0042                     SUBR   FINT        INTEGER TO FLOATING POINT CONVERSION      BSI00420
0043                     SUBR   IFLT        FLOATING POINT TO INTEGER CONVERSION      BSI00430
0044                     SUBR   TINT        TEST FOR INTEGER                          BSI00440
0045                     SUBR   SGNF        SIGN FUNCTION                             BSI00450
0046                     SUBR   ABSF        ABSOLUTE VALUE FUNCTION                   BSI00460
0047                     SUBR   ABS,ABSF                                              BSI00470
0048                     SUBR   RNDF        RANDOM NUMBER FUNCTION                    BSI00480
0049                     SUBR   INTF        GREATEST INTEGER FUNCTION                 BSI00490
0050                     SUBR   LOGF        NATURAL LOGARITHM FUNCTION                BSI00500
0051                     SUBR   ALOG,LOGF                                            BSI00510
0052                     SUBR   EXPF        NATURAL ANTILOGARITHM FUNCTION            BSI00520
0053                     SUBR   EXP,EXPF                                             BSI00530
0054                     SUBR   SQRF        SQUARE ROOT FUNCTION                      BSI00540
0055                     SUBR   SQRT,SQRF                                            BSI00550
0056                     SUBR   COSF        COSINE FUNCTION                           BSI00560
0057                     SUBR   COS,COSF                                             BSI00570
0058                     SUBR   SINF        SINE FUNCTION                             BSI00580
0059                     SUBR   SIN,SINF                                             BSI00590
0060                     SUBR   TANF        TANGENT FUNCTION                          BSI00600
0061                     SUBR   ATNF        ARCTANGENT FUNCTION                       BSI00610
0062                     SUBR   ATAN,ATNF                                            BSI00620
0063                     ENT    PP12        LAST WORD OF BASIC-MTHPAK                 BSI00630
0064                *                                                                BSI00640
0065                *                                                                BSI00650
0066                     EXT    ERR         ERROR REPORTING ROUTINE                   BSI00660
0067                     EXT    M1                                                    BSI00670
```

```
0068                    EXI   FMI                                   BS100680
0069               *                                                BS100690
0070               *                                                BS100700
0071                    REL                                         BS100710
0072               *                                                BS100720
0073               *                                                BS100730
0074                    EJCT                                        BS100740
```

```
0075                    *                                                                       BSI00750
0076                    *                                                                       BSI00760
0077                    *          SIGN FUNCTION                                                BSI00770
0078                    *                                                                       BSI00780
0079                    *    CALLING SEQUENCE:                                                  BSI00790
0080                    *                                                                       BSI00800
0081                    *          JST    SGNF                                                  BSI00810
0082                    *          DAC    ARG          ADDRESS OF THE ARGUMENT                  BSI00820
0083                    *          ......RETURN        RESULT RETURNED IN THE A AND B REGISTERS BSI00830
0084                    *                                                                       BSI00840
0085                    *                                                                       BSI00850
0086                    *          THE ARGUMENT IS LOADED, AND IF ZERO, THE ROUTINE EXITS WITH  BSI00860
0087                    *    ZERO.  OTHERWISE THE SIGN IS STORED INTO THE C BIT, AND FLOATING   BSI00870
0088                    *    ONE IS LOADED.  IF THE C BIT IS SET, THE A REGISTER IS TWO'S COM-  BSI00880
0089                    *    PLIMENTED TO GENERATE FLOATING POINT MINUS ONE.                    BSI00890
0090                    *                                                                       BSI00900
0091                    *                                                                       BSI00910
0092 00000   0 000000   SGNF DAC    **           ENTRY                                          BSI00920
0093 00001   0 02 00000      LDA    SGNF         LOAD THE ARGUMENT                              BSI00930
0094 00002   0 10 00070      JST    LARG         X                                              BSI00940
0095 00003   101040         SNZ                 SKIP IF NON-ZERO                               BSI00950
0096 00004  -0 01 00120      JMP*   LHIS+1       IF ZERO, RETURN WITH ZERO                      BSI00960
0097 00005   140320         CSA                 STORE ARGUMENT SIGN IN C BIT                   BSI00970
0098 00006   140040         CRA                 LOAD FLOATING POINT ONE                        BSI00980
0099 00007   000201         IAB                 X                                              BSI00990
0100 00010   0 02 01144     LDA    F1           X                                              BSI01000
0101 00011   100001         SRC                 SKIP IF SIGN IS POSITIVE                       BSI01010
0102 00012   140407         ICA                 TWO'S COMPLIMENT TO GENERATE FLOATING POINT    BSI01020
0103                    *                       MINUS ONE IF SIGN IS NEGATIVE                  BSI01030
0104 00013  -0 01 00120      JMP*   LHIS+1       RETURN                                         BSI01040
0105                         EJCT                                                               BSI01050
```

```
0106                    *                                                                       BS101060
0107                    *                                                                       BS101070
0108                    *          ABSOLUTE VALUE FUNCTION                                      BS101080
0109                    *                                                                       BS101090
0110                    *    CALLING SEQUENCE:                                                  BS101100
0111                    *                                                                       BS101110
0112                    *          JSI   ABSF                                                   BS101120
0113                    *          DAC   ARG          ADDRESS OF ARGUMENT                       BS101130
0114                    *          ......RETURN       FLOATING POINT RESULT IN A AND B REGISTERS BS101140
0115                    *                                                                       BS101150
0116                    *                                                                       BS101160
0117                    *          THE ARGUMENT IS LOADED, AND ITS SIGN IS TESTED.  IF THE SIGN BS101170
0118                    *    IS MINUS, THEN THE FLOATING POINT ARGUMENT IS TWO'S COMPLIMENTED.  BS101180
0119                    *                                                                       BS101190
0120                    *                                                                       BS101200
0121 00014   0 000000   ABSF DAC   **                 ENTRY                                     BS101210
0122 00015   0 02 00014       LDA   ABSF             LOAD THE ARGUMENT                         BS101220
0123 00016   0 10 00070       JSI   LARG             X                                         BS101230
0124 00017   100400           SPL                    SKIP IF POSITIVE                          BS101240
0125 00020   0 10 00122       JSI   N$22             COMPLIMENT IF NEGATIVE                    BS101250
0126 00021  -0 01 00120       JMP*  LHIS+1           RETURN                                    BS101260
0127                         EJCT                                                             BS101270
```

```
0128                    *                                                                       BSI01280
0129                    *                                                                       BSI01290
0130                    *          RANDOM NUMBER ROUTINE                                        BSI01300
0131                    *                                                                       BSI01310
0132                    *     CALLING SEQUENCE:                                                 BSI01320
0133                    *                                                                       BSI01330
0134                    *          JSI    RNDF                                                  BSI01340
0135                    *          DAC    ARG          ADDRESS OF ARGUMENT                      BSI01350
0136                    *          ......RETURN       RANDOM NUMBER RETURNED IN A AND B REGISTERSBSI01360
0137                    *                                                                       BSI01370
0138                    *                                                                       BSI01380
0139                    *          THE ARGUMENT IS LOADED, AND IF POSITIVE, THE HIGH PART OF THEBSI01390
0140                    *     ARGUMENT IS SAVED TO INITIALIZE A SEQUENCE, AND THE ARGUMENT IS   BSI01400
0141                    *     RETURNED AS A RANDOM NUMBER.  IF THE ARGUMENT IS NEGATIVE, ITS SIGNBSI01410
0142                    *     IS SET POSITIVE.  IF ZERO, THE LAST NUMBER GENERATED IN THE       BSI01420
0143                    *     SEQUENCE IS LOADED.  THE A REGISTER IS MULTIPLIED BY 5**5, AND THEBSI01430
0144                    *     RESULT IS SAVED TO GENERATE THE NEXT NUMBER IN THE SEQUENCE.  THE BSI01440
0145                    *     EXPONENT IS INITIALIZED TO ZERO PLUS THE BIAS, AND THE RESULT IS  BSI01450
0146                    *     PACKED INTO FLOATING POINT FORMAT.                                BSI01460
0147                    *                                                                       BSI01470
0148                    *                                                                       BSI01480
0149 00022   0 000000   RNDF DAC    **                 ENTRY                                    BSI01490
0150                    *                                                                       BSI01500
0151                    *     LOAD AND TEST SIGN OF ARGUMENT                                    BSI01510
0152                    *                                                                       BSI01520
0153 00023   0 02 00022      LDA    RNDF               LOAD THE ARGUMENT                        BSI01530
0154 00024   0 10 00070      JSI    LARG                                                        BSI01540
0155 00025   0 11 01060      CAS    FO                 TEST SIGN OF ARGUMENT                    BSI01550
0156 00026   0 01 00042      JMP    RND3               POSITIVE - INITIALIZE A SEQUENCE         BSI01560
0157 00027   0 02 00045      LDA    RND1               ZERO - LOAD LAST NUMBER GENERATED        BSI01570
0158                    *                                                                       BSI01580
0159                    *     HERE IF NEGATIVE OR ZERO                                          BSI01590
0160                    *                                                                       BSI01600
0161 00030   140100           SSP                       NEGATIVE - SET SIGN POSITIVE             BSI01610
0162 00031   0 10 00507      JSI    M$11               MULTIPLY BY                              BSI01620
0163 00032   0 000044        DAC    K5E5               5**5                                     BSI01630
0164 00033   0 04 00045      STA    RND1               SAVE TO GENERATE NEXT NUMBER IN SEQUENCE BSI01640
```

```
0165 00034   0 04 00152        STA   EXP1                                          BSI01650
0166 00035   0 02 00572        LDA   C200    INITIALIZE EXPONENT TO BIAS          BSI01660
0167 00036   0 13 00152        IMA   EXP1                                          BSI01670
0168 00037   0 10 00202        JSI   NORM    FORM FLOATING POINT NUMBER           BSI01680
0169 00040   0 12 00022        IRS   RNDF    INCREMENT FOR RETURN                 BSI01690
0170 00041  -0 01 00022        JMP*  RNDF    EXIT                                 BSI01700
0171                    *                                                         BSI01710
0172                    *     HERE IF POSITIVE                                    BSI01720
0173                    *                                                         BSI01730
0174 00042   0 04 00045 RN03  STA   RND1    SAVE TO INITIALIZE A SEQUENCE        BSI01740
0175 00043  -0 01 00120        JMP*  LHIS+1  EXIT WITH ORIGINAL ARGUMENT          BSI01750
0176                    *                                                         BSI01760
0177                    *                                                         BSI01770
0178 00044   006065       K5E5  DEC   3125                                        BSI01780
0179 00045   065432       RND1  OCT   65432                                       BSI01790
0180                           EJCT                                               BSI01800
```

```
0181                    *                                                                       BSI01810
0182                    *                                                                       BSI01820
0183                    *          EFFECTIVE ADDRESS ROUTINE                                    BSI01830
0184                    *                                                                       BSI01840
0185                    *    CALLING SEQUENCE:                                                  BSI01850
0186                    *                                                                       BSI01860
0187                    *          JST    ADDR          ADDRESS OF DAC TO ARGUMENT IN A REGISTER BSI01870
0188                    *          ......RETURN        EFFECTIVE ADDRESS AND RETURN ADDRESS LEFT BSI01880
0189                    *                               IN SPECIFIED TEMPORARY STORAGE LOCATIONS BSI01890
0190                    *                                                                       BSI01900
0191                    *                                                                       BSI01910
0192                    *          THE ROUTINE LOADS THE ADDRESS OF THE ARGUMENT.  IF THE INDEX BSI01920
0193                    *    BIT IS SET THE CONTENTS OF THE X REGISTER ARE ADDED TO THE ADDRESS.BSI01930
0194                    *    THE INDIRECT BIT IS THEN TESTED, AND IF SET, THE CONTENTS OF THE  BSI01940
0195                    *    ADDRESS ARE LOADED, AND THE ROUTINE LOOPS TO TEST THE INDEX BIT.  BSI01950
0196                    *    IF THE INDIRECT BIT IS RESET, THE EFFECTIVE ADDRESS IS SAVED, AND BSI01960
0197                    *    THE RETURN IS MADE.  THE ORIGINAL CONTENTS OF THE A REGISTER ARE  BSI01970
0198                    *    INCREMENTED AND SAVED AS A RETURN ADDRESS FOR THE CALLING ROUTINE.BSI01980
0199                    *                                                                       BSI01990
0200                    *                                                                       BSI02000
0201 00046   0 000000  ADDR DAC   **              EFFECTIVE ADDRESS ROUTINE                    BSI02010
0202 00047   0 04 00117       STA   LHIS          SAVE ADDRESS                                 BSI02020
0203 00050   141206            AOA               INCREMENT AND SAVE                           BSI02030
0204 00051   0 04 00120       STA   LHIS+1        FOR RETURN ADDRESS                           BSI02040
0205 00052  -0 02 00117 AD1   LDA*  LHIS          LOAD ARGUMENT ADDRESS                        BSI02050
0206                    *                                                                       BSI02060
0207                    *    TEST THE INDEX BIT OF THE ADDRESS                                  BSI02070
0208                    *                                                                       BSI02080
0209 00053   0416 77           ALR   1             ROTATE LEFT                                  BSI02090
0210 00054   140320            CSA                 SET C BIT IF TAG                             BSI02100
0211 00055   100001            SRC                 TEST C BIT                                   BSI02110
0212 00056   0 01 00065       JMP   AD2           JUMP IF TAG                                  BSI02120
0213 00057   0406 77           ARR   1             REPOSITION                                   BSI02130
0214                    *                                                                       BSI02140
0215                    *    TEST THE INDIRECT BIT OF THE ADDRESS                               BSI02150
0216                    *                                                                       BSI02160
0217 00060   140320      AD3   CSA                 SET C BIT IF FLAG                            BSI02170
```

```
0218 00061   0 04 00117      STA   LHIS                SAVE ADDRESS                    BS102180
0219 00062   100001          SRC                       TEST C BIT                      BS102190
0220 00063   0 01 00052      JMP   AD1                 LOOP IF FLAG                    BS102200
0221 00064  -0 01 00046      JMP*  ADDR                RETURN                          BS102210
0222                 *                                                                BS102220
0223                 *    ADD THE INDEX REGISTER IF THE INDEX BIT IS SET              BS102230
0224                 *                                                                BS102240
0225 00065   0406 77    AD2   ARR   1                   REPOSITION                      BS102250
0226 00066   0 06 00000      ADD   0                   ADD INDEX REGISTER              BS102260
0227 00067   0 01 00060      JMP   AD3                 JUMP BACK TO TEST FLAG          BS102270
0228                         EJCT                                                     BS102280
```

```
0229                    *                                                                       BSI02290
0230                    *                                                                       BSI02300
0231                    *          LOAD ARGUMENT ROUTINE                                        BSI02310
0232                    *                                                                       BSI02320
0233                    *    CALLING SEQUENCE:                                                  BSI02330
0234                    *                                                                       BSI02340
0235                    *          JSI    LARG        ADDRESS OF DAC TO ARGUMENT IN A REGISTER  BSI02350
0236                    *          ......RETURN       THE ARGUMENT IS RETURNED IN THE A AND B   BSI02360
0237                    *                             REGISTERS                                 BSI02370
0238                    *                                                                       BSI02380
0239                    *                                                                       BSI02390
0240                    *          THIS ROUTINE LOADS A DOUBLE WORD ARGUMENT.  IT CALL UPON ADDRBSI02400
0241                    *    TO OBTAIN THE EFFECTIVE ADDRESS.  THE CONTENTS OF THE EFFECTIVE   BSI02410
0242                    *    ADDRESS IS LOADED IN THE A REGISTER, AND THE CONTENTS OF THE      BSI02420
0243                    *    EFFECTIVE ADDRESS PLUS ONE IS LOADED IN THE B REGISTER.           BSI02430
0244                    *                                                                       BSI02440
0245                    *                                                                       BSI02450
0246 00070   0 000000   LARG DAC    **                 LOAD ARGUMENT ROUTINE ENTRY             BSI02460
0247 00071   0 10 00046      JSI    ADDR               EFFECTIVE ADDRESS ROUTINE               BSI02470
0248 00072  -0 02 00117      LDA*   LHIS               LOAD HIGH ARGUMENT                      BSI02480
0249 00073   000201          IAB                                                               BSI02490
0250 00074   0 12 00117      IRS    LHIS               INCREMENT FOR ADDRESS OF LOW ARGUMENT   BSI02500
0251 00075  -0 02 00117      LDA*   LHIS               LOAD LOW                                BSI02510
0252 00076   000201          IAB                       REPOSITION                              BSI02520
0253 00077  -0 01 00070      JMP*   LARG               RETURN                                  BSI02530
0254                         EJCT                                                              BSI02540
```

```
0255                    *                                                                   BS102550
0256                    *                                                                   BS102560
0257                    *          DOUBLE LOAD ROUTINE                                      BS102570
0258                    *                                                                   BS102580
0259                    *     CALLING SEQUENCE:                                             BS102590
0260                    *                                                                   BS102600
0261                    *          JSI   L$22                                               BS102610
0262                    *          DAC   ARG          FLAG AND TAG MAY BE SET               BS102620
0263                    *          ......RETURN       ARGUMENT IN A AND B REGISTERS         BS102630
0264                    *                                                                   BS102640
0265                    *                                                                   BS102650
0266                    *          THIS ROUTINE LOADS THE ADDRESS OF THE DAC TO THE ARGUMENT,  BS102660
0267                    *     AND THEN CALLS LARG TO LOAD THE DOUBLE WORD ARGUMENT.  THE RETURN  BS102670
0268                    *     IS MADE THROUGH RETURN ADDRESS CALCULATED BY ADDR.            BS102680
0269                    *                                                                   BS102690
0270                    *                                                                   BS102700
0271 00100    0 000000  L$22 DAC   **               DOUBLE LOAD ENTRY                      BS102710
0272 00101    0 02 00100     LDA   L$22             LOAD DAC OF ARGUMENT ADDRESS           BS102720
0273 00102    0 10 00070     JSI   LARG             LOAD ARGUMENT ROUTINE                  BS102730
0274 00103   -0 01 00120     JMP*  LHIS+1           RETURN                                 BS102740
0275                    EJCT                                                               BS102750
```

```
0276                    *                                                                       BSI02760
0277                    *                                                                       BSI02770
0278                    *          DOUBLE STORE ROUTINE                                         BSI02780
0279                    *                                                                       BSI02790
0280                    *     CALLING SEQUENCE:                                                 BSI02800
0281                    *                                                                       BSI02810
0282                    *          JSI    H$22          ARGUMENT IN A AND B REGISTERS           BSI02820
0283                    *          DAC    ARG           FLAG AND TAG MAY BE SET                 BSI02830
0284                    *          ......RETURN                                                 BSI02840
0285                    *                                                                       BSI02850
0286                    *                                                                       BSI02860
0287                    *          THE ADDRESS OF THE DAC OF THE ARGUMENT IS LOADED IN THE A    BSI02870
0288                    *     REGISTER, AND THEN ADDR IS CALLED TO OBTAIN THE EFFECTIVE ADDRESS. BSI02880
0289                    *     THE ORIGINAL CONTENTS OF THE A REGISTER ARE STORED IN THE LOCATION BSI02890
0290                    *     SPECIFIED BY THE EFFECTIVE ADDRESS, AND THE CONTENTS OF THE B     BSI02900
0291                    *     REGISTER ARE STORED IN THE NEXT SEQUENTIAL LOCATION.              BSI02910
0292                    *                                                                       BSI02920
0293                    *                                                                       BSI02930
0294 00104   0 000000   H$22 DAC   **          FLOATING POINT STORE ENTRY                       BSI02940
0295 00105   0 04 00121      STA   LHIS+2       SAVE HIGH                                        BSI02950
0296 00106   0 02 00104      LDA   H$22         LOAD ADDRESS OF DAC                              BSI02960
0297 00107   0 10 00046      JSI   ADDR         EFFECTIVE ADDRESS ROUTINE                        BSI02970
0298 00110   0 02 00121      LDA   LHIS+2       LOAD HIGH                                        BSI02980
0299 00111  -0 04 00117      STA*  LHIS         STORE HIGH                                       BSI02990
0300 00112   000201          IAB                                                                BSI03000
0301 00113   0 12 00117      IRS   LHIS         INCREMENT FOR ADDRESS OF LOW                     BSI03010
0302 00114  -0 04 00117      STA*  LHIS         STORE LOW                                        BSI03020
0303 00115   000201          IAB                REPOSITION                                       BSI03030
0304 00116  -0 01 00120      JMP*  LHIS+1       RETURN                                           BSI03040
0305                    *                                                                       BSI03050
0306                    *                                                                       BSI03060
0307 00117   000000     LHIS BSZ   3                                                            BSI03070
0308                         EJCT                                                               BSI03080
```

```
0309                    *                                                                   BSI03090
0310                    *                                                                   BSI03100
0311                    *          TWO'S COMPLIMENT A FLOATING POINT NUMBER                 BSI03110
0312                    *                                                                   BSI03120
0313                    *     CALLING SEQUENCE:                                             BSI03130
0314                    *                                                                   BSI03140
0315                    *          JST    N$22          FLOATING POINT IN A AND B REGISTERS BSI03150
0316                    *          ......RETURN.        TWOS COMPLIMENTED FLOATING POINT IN A AND BSI03160
0317                    *                               B REGISTERS                         BSI03170
0318                    *                                                                   BSI03180
0319                    *                                                                   BSI03190
0320                    *          THE C BIT IS SET AFTER ENTRANCE TO THIS ROUTINE TO PROVIDE A BSI03200
0321                    *     TRUE TWO'S COMPLIMENT IF THE LOW ORDER WORD IS ZERO.  IF IT IS BSI03210
0322                    *     FOUND TO BE NON-ZERO, THE C BIT IS RESET.  THE LOW ORDER WORD IS BSI03220
0323                    *     TWO'S COMPLIMENTED.  THE HIGH ORDER WORD IS ONE'S COMPLIMENTED, BSI03230
0324                    *     AND THE C BIT IS ADDED.                                       BSI03240
0325                    *                                                                   BSI03250
0326                    *                                                                   BSI03260
0327 00122   0 000000   N$22 DAC   **              FLOATING POINT TWO'S COMPLIMENT ENTRY     BSI03270
0328 00123   140600          SCB                   SET CARRY INDICATOR                       BSI03280
0329 00124   000201          LAB                   LOAD LOW                                  BSI03290
0330 00125   100040          SZE                   TEST ZERO                                 BSI03300
0331 00126   140200          RCB                   IF NOT, RESET CARRY INDICATOR             BSI03310
0332 00127   140407          CMA                   TWO'S COMPLIMENT                          BSI03320
0333 00130   000201          LAB                   LOAD HIGH                                 BSI03330
0334 00131   140401          CMA                   ONE'S COMPLIMENT                          BSI03340
0335 00132   141216          ACA                   ADD CARRY                                 BSI03350
0336 00133  -0 01 00122       JMP*  N$22            RETURN                                    BSI03360
0337                    *EJCT                                                               BSI03370
```

```
0338                    *                                                                   BSI03380
0339                    *                                                                   BSI03390
0340                    *          UNPACK A FLOATING POINT                                  BSI03400
0341                    *                                                                   BSI03410
0342                    *     CALLING SEQUENCE:                                             BSI03420
0343                    *                                                                   BSI03430
0344                    *          JSI    UNPK          FLOATING POINT IN A AND B REGISTERS BSI03440
0345                    *          ......RETURN         MANTISSA IN A AND B REGISTERS, EXPONENT IN BSI03450
0346                    *                               SPECIFIED TEMPORARY STORAGE LOCATION BSI03460
0347                    *                                                                   BSI03470
0348                    *                                                                   BSI03480
0349                    *          THE EXPONENT IS RIGHT JUSTIFIED, AND IF THE FLOATING POINT BSI03490
0350                    *     NUMBER IS NEGATIVE,IT IS COMPLIMENTED.  THE MANTISSA IS RIGHT JUS- BSI03500
0351                    *     TIFIED SO THAT THE BINARY POINT IS BETWEEN THE FIRST TWO BITS OF BSI03510
0352                    *     THE A REGISTER.  THE FIRST BIT OF THE B REGISTER IS CLEARED TO  BSI03520
0353                    *     LEAVE THE MANTISSA IN DOUBLE WORD FIXED POINT FORMAT.         BSI03530
0354                    *                                                                   BSI03540
0355                    *                                                                   BSI03550
0356 00134  0 000000    UNPK DAC   **                  ENTRY                                BSI03560
0357 00135  0 04 00152       STA   EXPT                SAVE THE HIGH WORD                   BSI03570
0358                    *                                                                   BSI03580
0359                    *     EXTRACT THE EXPONENT AND SAVE IT                              BSI03590
0360                    *                                                                   BSI03600
0361 00136  0405 71          ARS   7                   RIGHT JUSTIFY THE EXPONENT           BSI03610
0362 00137  100400           SPL                       SKIP IF POSITIVE                     BSI03620
0363 00140  140401           CMA                       ONE'S COMPLIMENT IF NEGATIVE         BSI03630
0364 00141  0 13 00152       IMA   EXPT                SAVE THE EXPONENT, RECOVER THE HIGH WORD BSI03640
0365                    *                                                                   BSI03650
0366                    *     PUT THE MANTISSA IN DOUBLE WORD FIXED POINT FORMAT            BSI03660
0367                    *                                                                   BSI03670
0368 00142  0 03 00153       ANA   MSEX                STRIP THE EXPONENT                   BSI03680
0369 00143  100400           SPL                       SKIP IF POSITIVE                     BSI03690
0370 00144  0 05 00154       ERA   MES                 MOVE THE SIGN IF NEGATIVE            BSI03700
0371 00145  0410 70          LLL   8                   LEFT JUSTIFY THE MANTISSA            BSI03710
0372 00146  000201           IAB                       CLEAR BIT ONE OF THE LOW MANTISSA    BSI03720
0373 00147  0404 77          LGR   1                   X                                    BSI03730
0374 00150  000201           IAB                       X                                    BSI03740
```

```
0375 00151  -0 01 00134       JMP*  UNPK        RETURN                    BSI03750
0376                          *                                           BSI03760
0377                          *                                           BSI03770
0378 00152   000000          EXPT  BSZ   1                                 BSI03780
0379 00153   100177          MSEX  OCT   100177                            BSI03790
0380 00154   100200          MES   OCT   100200                            BSI03800
0381                               EJCT                                   BSI03810
```

```
0382                    *                                                                   BS103820
0383                    *                                                                   BS103830
0384                    *          DOUBLE WORD ADDITION ROUTINE                             BS103840
0385                    *                                                                   BS103850
0386                    *     CALLING SEQUENCE:                                             BS103860
0387                    *                                                                   BS103870
0388                    *          JST    DADD          FIRST ARGUMENT IN A AND B REGISTERS, SECONDBS103880
0389                    *                               ARGUMENT IN SPECIFIED TEMPORARY STORAGE BS103890
0390                    *                               LOCATIONS                           BS103900
0391                    *          ......RETURN         RESULT IN A AND B REGISTERS         BS103910
0392                    *                                                                   BS103920
0393                    *                                                                   BS103930
0394                    *          THIS ROUTINE ASSUMES THE ARGUMENTS ARE IN DOUBLE WORD FIXED BS103940
0395                    *     POINT FORMAT.  THE LOW PARTS OF THE ARGUMENTS ARE ADDED, AND BIT BS103950
0396                    *     ONE IS COPIED INTO THE C BIT AND THEN SET TO ZERO.  THE C BIT AND BS103960
0397                    *     THE HIGH PARTS OF THE TWO ARGUMENTS ARE ADDED TOGETHER.  IF THERE BS103970
0398                    *     IS NO OVERFLOW THE ROUTINE EXITS.  OTHERWISE, THE RESULT IS SHIFTEDBS103980
0399                    *     RIGHT AND THE EXPONENT IS INCREMENTED.                        BS103990
0400                    *                                                                   BS104000
0401                    *                                                                   BS104010
0402 00155   0 000000   DADD  DAC   **          DOUBLE ADDITION ROUTINE ENTRY               BS104020
0403 00156   000201           TAB               LOAD LOW X                                  BS104030
0404 00157   0 06 00201       ADD   LOW         ADD LOW Y                                   BS104040
0405 00160   140320           CSA               CARRY INTO C BIT                            BS104050
0406 00161   000201           TAB               LOAD HIGH X                                 BS104060
0407 00162   141210           ACA               ADD CARRY                                   BS104070
0408 00163   100001           SRC               SKIP IF NO OVERFLOW                         BS104080
0409 00164   0 01 00174       JMP   DA01        JUMP IF OVERFLOW                            BS104090
0410 00165   0 06 00200       ADD   HIGH        ADD HIGH Y                                  BS104100
0411 00166   101001           SSC               CHECK FOR OVERFLOW                          BS104110
0412 00167  -0 01 00155       JMP*  DADD        NO OVERFLOW - RETURN                        BS104120
0413                    *                                                                   BS104130
0414                    *     HERE IF OVERFLOW                                              BS104140
0415                    *                                                                   BS104150
0416 00170   0401 77     DA02  LRS   1           OVERFLOW - SHIFT RIGHT                      BS104160
0417 00171   140024           CHS               RESTORE THE SIGN                            BS104170
0418 00172   0 12 00152       IRS   EXPI        STEP THE EXPONENT                           BS104180
```

```
0419 00173  -0 01 00155       JMP*  DADD              RETURN                                    BS104190
0420                          *                                                               BS104200
0421                          *     CHECK HERE FOR COMPENSATING OVERFLOW                      BS104210
0422                          *                                                               BS104220
0423 00174   0 06 00200 DA01  ADD   HIGH              ADD HIGH Y                                BS104230
0424 00175   100001           SRC                     CHECK FOR COMPENSATING OVERFLOW           BS104240
0425 00176  -0 01 00155       JMP*  DADD              YES - RETURN                              BS104250
0426 00177   0 01 00170       JMP   DA02              NO - OVERFLOW                             BS104260
0427                          *                                                               BS104270
0428                          *                                                               BS104280
0429 00200   000000     HIGH  BSZ   1                                                         BS104290
0430 00201   000000     LOW   BSZ   1                                                         BS104300
0431                          EJCT                                                            BS104310
```

```
0432                    *                                                                   BS104320
0433                    *                                                                   BS104330
0434                    *          NORMALIZE AND REPACK INTO FLOATING POINT FORMAT           BS104340
0435                    *                                                                   BS104350
0436                    *     CALLING SEQUENCE:                                             BS104360
0437                    *                                                                   BS104370
0438                    *          JST    NORM        MANTISSA IN A AND B REGISTERS, EXPONENT IN BS104380
0439                    *                             SPECIFIED TEMPORARY STORAGE LOCATION  BS104390
0440                    *          ......RETURN       FLOATING POINT IN A AND B REGISTERS    BS104400
0441                    *                                                                   BS104410
0442                    *                                                                   BS104420
0443                    *          THE MANTISSA IS EXPECTED IN DOUBLE WORD FIXED POINT FORMAT BS104430
0444                    *     WITH THE BINARY POINT BETWEEN FIRST TWO BITS OF THE A REGISTER. BS104440
0445                    *     THE EXPONENT IS ASSUMED TO BE BIASED AND UNCOMPLIMENTED.  IF THE BS104450
0446                    *     MANTISSA IS ZERO, THE ROUTINE EXITS WITH ZERO.  OTHERWISE THE MAN-BS104460
0447                    *     TISSA IS NORMALIZED AND ROUNDED.  THEN IT IS SHIFTED LEFT TO  BS104470
0448                    *     MAKE ROOM FOR THE EXPONENT, AND BIT ONE OF THE SECOND WORD IS BS104480
0449                    *     FILLED.  THE EXPONENT IS DECREMENTED BY THE SHIFT COUNT OF THE BS104490
0450                    *     NORMALIZE, AND THEN TESTED FOR OVERFLOW AND UNDERFLOW.  UNDERFLOW BS104500
0451                    *     AND OVERFLOW ARE FLAGGED BY NO AND NO RESPECTIVELY.  OTHERWISE THE BS104510
0452                    *     TOTAL WORD IS FORMED, AND THE ROUTINE EXITS.                  BS104520
0453                    *                                                                   BS104530
0454                    *                                                                   BS104540
0455 00202   0 000000   NORM  DAC    **              ENTRY                                  BS104550
0456                    *                                                                   BS104560
0457                    *     TEST FOR ZERO MANTISSA                                        BS104570
0458                    *                                                                   BS104580
0459 00203   100040           SZE                    TEST HIGH MANTISSA EQUAL TO ZERO       BS104590
0460 00204   0 01 00211       JMP    NZ              NO-JUMP TO NORMALIZE                   BS104600
0461 00205   000201           TAB                    YES-LOAD LOW MANTISSA                  BS104610
0462 00206   101040           SNZ                    TEST EQUAL TO ZERO                     BS104620
0463 00207  -0 01 00202       JMP*   NORM            YES-RETURN WITH ZERO                   BS104630
0464 00210   000201           TAB                    NO-REPOSITION                          BS104640
0465                    *                                                                   BS104650
0466                    *     NORMALIZE THE MANTISSA                                        BS104660
0467                    *                                                                   BS104670
0468 00211   0 04 00272 NZ    STA    CNTR            INITIALIZE THE SHIFT COUNTER TO ZERO   BS104680
```

```
0469 00212   140040          CRA                    X                                  BS104690
0470 00213   0 13 00272       IMA    CNTR            X                                  BS104700
0471 00214   100000          SKP                    DON'T INCREMENT SHIFT COUNTER THE FIRST  BS104710
0472                 *                              TIME THROUGH                       BS104720
0473 00215   0 12 00272       IRS    CNTR            INCREMENT THE SHIFT COUNTER BY ONE  BS104730
0474 00216   0411 77          LLS    1               SHIFT LEFT TO NORMALIZE            BS104740
0475 00217   101001          SSC                    TEST FOR NORMAL RESULT             BS104750
0476 00220   0 01 00215       JMP    *-3             NO-LOOP TO INCREMENT THE SHIFT COUNTER  BS104760
0477 00221   0401 77          LRS    1               RESTORE WITH WRONG SIGN            BS104770
0478 00222   140024          CHS                    RESTORE THE SIGN                   BS104780
0479                 *                                                                 BS104790
0480                 *    ADD A ROUNDING FACTOR TO THE RIGHT OF THE LEAST SIGNIFICANT BIT  BS104800
0481                 *    OF THE MANTISSA                                              BS104810
0482                 *                                                                 BS104820
0483 00223   0 04 00200       STA    HIGH            STORE THE HIGH MANTISSA            BS104830
0484 00224   0 02 00273       LDA    C100            LOAD THE LOW ROUNDING FACTOR       BS104840
0485 00225   000201          TAB                    STORE THE LOW MANTISSA             BS104850
0486 00226   0 04 00201       STA    LOW             X                                  BS104860
0487 00227   140040          CRA                    LOAD THE HIGH ROUNDING FACTOR      BS104870
0488 00230   0 10 00155       JSI    DADD                                               BS104880
0489                 *                                                                 BS104890
0490                 *    TEST THE MANTISSA EQUAL TO NEGATIVE ONE                      BS104900
0491                 *                                                                 BS104910
0492 00231   0 11 00276       CAS    MLNN            TEST HIGH MANTISSA EQUAL TO -1     BS104920
0493 00232   0 01 00244       JMP    N3              NO-JUMP TO PACK THE MANTISSA       BS104930
0494 00233   000201          TAB                    YES-LOAD LOW MANTISSA              BS104940
0495 00234   0 03 00275       ANA    MLMA            MASK SIGNIFICANT BITS OF THE LOW MANTISSA  BS104950
0496 00235   100040          SZE                    TEST LOW MANTISSA EQUAL TO ZERO    BS104960
0497 00236   0 01 00243       JMP    N4              NO-JUMP TO PACK MANTISSA           BS104970
0498 00237   0 12 00152       IRS    EXPT            YES-MANTISSA EQUALS NEGATIVE ONE-INCREMENT  BS104980
0499                 *                              THE EXPONENT BY ONE                BS104990
0500 00240   000201          TAB                    AND GENERATE A MANTISSA OF NEGATIVE ONE  BS105000
0501 00241   0405 77          ARS    1               HALF                               BS105010
0502 00242   100000          SKP                                                       BS105020
0503                 *                                                                 BS105030
0504                 *    FILL BIT ONE OF LOW MANTISSA AND MAKE ROOM FOR THE EXPONENT  BS105040
0505                 *                                                                 BS105050
```

```
0506 00243   000201      N4   IAB                                                    BS105060
0507 00244   0401 11      N3   LRS   1              MAKE ROOM FOR THE EXPONENT      BS105070
0508 00245   000201            IAB                 AND FILL BIT ONE OF THE LOW MANTISSA BS105080
0509 00246   0414 11           LGL   1             X                               BS105090
0510 00247   000201            IAB                 X                               BS105100
0511 00250   0400 11           LRL   1                                               BS105110
0512 00251   0414 11           LGL   1             BUT SAVE THE SIGN OF THE MANTISSA BS105120
0513 00252   0405 11           ARS   1                                               BS105130
0514                      *                                                        BS105140
0515                      *    CHECK FOR OVERFLOW AND UNDERFLOW                    BS105150
0516                      *                                                        BS105160
0517 00253   0 13 00152        IMA   EXPI          LOAD THE EXPONENT               BS105170
0518 00254   0 07 00272        SUB   CNTR          SUBTRACT THE NORMALIZING COUNT  BS105180
0519 00255   100400            SPL                 TEST FOR UNDERFLOW              BS105190
0520 00256   0 01 00266        JMP   N6            YES-JUMP TO FLAG UNDERFLOW      BS105200
0521 00257   0 07 00274        SUB   C400          NO-TEST FOR OVERFLOW            BS105210
0522 00260   101400            SMI                 X                               BS105220
0523 00261   0 01 00270        JMP   N7            YES-JUMP TO FLAG OVERFLOW       BS105230
0524 00262   0 06 00274        ADD   C400          NO-RESTORE THE EXPONENT         BS105240
0525                      *                                                        BS105250
0526                      *    FORM TOTAL WORD AND RETURN                          BS105260
0527                      *                                                        BS105270
0528 00263   0414 11           LGL   1             POSITION THE EXPONENT           BS105280
0529 00264   0 05 00152        ERA   EXPI          FORM TOTAL WORD                 BS105290
0530 00265  -0 01 00202        JMP*  NORM          RETURN                          BS105300
0531                      *                                                        BS105310
0532                      *    HERE TO FLAG UNDERFLOW AND OVERFLOW                 BS105320
0533                      *                                                        BS105330
0534 00266   0 10 00000  N6   JSI   ERR                                            BS105340
0535 00267   147325            BCI   1,NU          FLAG UNDERFLOW                  BS105350
0536 00270   0 10 00000  N7   JSI   ERR                                            BS105360
0537 00271   147317            BCI   1,NU          FLAG OVERFLOW                   BS105370
0538                      *                                                        BS105380
0539                      *                                                        BS105390
0540 00272   000000     CNTR BSZ   1                                               BS105400
0541 00273   000100     C100 OCT   100                                             BS105410
0542 00274   000400     C400 OCT   400                                             BS105420
```

```
0543 00275    077600      MLMA UCI    77600                          BS105430
0544 00276    100000      MLNN UCI    100000                         BS105440
0545                           EJCI                                   BS105450
```

```
0546                        *                                                                   BSI05460
0547                        *                                                                   BSI05470
0548                        *          FLOATING POINT ADDITION AND SUBTRACTION                  BSI05480
0549                        *                                                                   BSI05490
0550                        *     CALLING SEQUENCE:                                             BSI05500
0551                        *                                                                   BSI05510
0552                        *          JSI    A$22(S$22)  FIRST ARGUMENT IN A AND B REGISTERS   BSI05520
0553                        *          DAC    ARG2        POINTER TO SECOND ARGUMENT            BSI05530
0554                        *          ......RETURN       RESULT IN A AND B REGISTERS           BSI05540
0555                        *                                                                   BSI05550
0556                        *                                                                   BSI05560
0557                        *          IF THE ARGUMENTS ARE TO BE SUBTRACTED A FLAG IS SET, AND THEBSI05570
0558                        *     SECOND ARGUMENT IS TWO'S COMPLIMENTED AFTER IT IS LOADED.  THE BSI05580
0559                        *     ARGUMENTS ARE UNPACKED, AND THE MANTISSA OF THE ARGUMENT OF SMALLERBSI05590
0560                        *     MAGNITUDE IS SHIFTED RIGHT THE DIFFERENCE OF THE EXPONETS OF THE BSI05600
0561                        *     ARGUMENTS.  IF THE DIFFERENCE OF THE EXPONENTS IS GREATER THAN THEBSI05610
0562                        *     NUMBER OF SIGNIFICANT BITS, ALL BITS OF THE SMALLER ARGUMENT EXCEPTBSI05620
0563                        *     FOR THE SIGN BIT ARE SHIFTED OUT.  THE MANTISSAS ARE ADDED.  THE BSI05630
0564                        *     RESULT IS PACKED INTO FLOATING POINT FORMAT WITH AN EXPONENT      BSI05640
0565                        *     EQUAL TO THE EXPONENT OF THE LARGER ARGUMENT.                     BSI05650
0566                        *                                                                   BSI05660
0567                        *                                                                   BSI05670
0568 00277   0 000000   S$22 DAC    **          SUBTRACTION ENTRY                                BSI05680
0569 00300   0 04 00355      STA    TEMP        SAVE HIGH PART                                   BSI05690
0570 00301   0 02 00277      LDA    S$22        MOVE RETURN ADDRESS                              BSI05700
0571 00302   0 04 00304      STA    A$22        TO THE ADDITION ROUTINE                          BSI05710
0572 00303   0 01 00307      JMP    *+4         JUMP INTO THE ADDITION ROUTINE                   BSI05720
0573 00304   0 000000   A$22 DAC    **          ADDITION ENTRY                                   BSI05730
0574 00305   0 04 00355      STA    TEMP        SAVE HIGH PART                                   BSI05740
0575 00306   0 02 00000      LDA    M1          LOAD MINUS ONE                                   BSI05750
0576 00307   0 13 00355      IMA    TEMP        SAVE FLAG, RECOVER HIGH PART                     BSI05760
0577                        *                                                                   BSI05770
0578                        *     UNPACK AND SAVE THE FIRST ARGUMENT                            BSI05780
0579                        *                                                                   BSI05790
0580 00310   0 10 00134      JSI    UNPK        UNPACK THE FIRST ARGUMENT                        BSI05800
0581 00311   0 10 00104      JSI    H$22        STORE IT                                         BSI05810
0582 00312   0 000200        DAC    HIGH                                                         BSI05820
```

```
0583 00313   0 02 00152       LDA   EXPT                                          BSI05830
0584 00314   0 04 00272       STA   CNTR          SAVE THE EXPONENT               BSI05840
0585                     *                                                       BSI05850
0586                     *     LOAD AND UNPACK THE SECOND ARGUMENT               BSI05860
0587                     *                                                       BSI05870
0588 00315   0 02 00304       LDA   A$22          LOAD DAC TO ARGUMENT ADDRESS    BSI05880
0589 00316   0 10 00070       JSI   LARG          LOAD ARGUMENT ROUTINE           BSI05890
0590 00317   0 12 00355       IRS   TEMP          SKIP IF ADDITION                BSI05900
0591 00320   0 10 00122       JSI   N$22          COMPLIMENT IF SUBTRACTION       BSI05910
0592 00321   0 10 00134       JSI   UNPK          UNPACK THE SECOND ARGUMENT      BSI05920
0593 00322   0 04 00355       STA   TEMP          SAVE HIGH SECOND ARGUMENT       BSI05930
0594                     *                                                       BSI05940
0595                     *     TAKE THE DIFFERENCE OF THE EXPONENTS              BSI05950
0596                     *                                                       BSI05960
0597 00323   0 02 00272       LDA   CNTR          LOAD THE EXPONENT OF THE FIRST ARGUMENT  BSI05970
0598 00324   0 07 00152       SUB   EXPT          SUBTRACT EXPONENT OF SECOND ARGUMENT     BSI05980
0599 00325   100400           SPL                 TEST SIGN                       BSI05990
0600 00326   0 01 00334       JMP   A6            SECOND ARGUMENT LARGER - JUMP   BSI06000
0601                     *                                                       BSI06010
0602                     *     IF THE FIRST ARGUMENT IS LARGER, FORM THE SHIFT COUNT AND SAVE  BSI06020
0603                     *     THE FIRST ARGUMENT'S EXPONENT AS THE EXPONENT OF THE RESULT     BSI06030
0604                     *                                                       BSI06040
0605 00327   140407           CA                  FIRST ARGUMENT LARGER - COMPLIMENT TO FORM BSI06050
0606                     *                       SHIFT COUNT                     BSI06060
0607 00330   0 04 00152       STA   EXPT          SAVE                            BSI06070
0608 00331   0 02 00272       LDA   CNTR          LOAD LARGER ARGUMENT EXPONENT   BSI06080
0609 00332   0 13 00152       IMA   EXPT          STORE IN EXPT, RECOVER SHIFT COUNT  BSI06090
0610 00333   0 01 00342       JMP   A1            JUMP TO FORM SHIFT INSTRUCTION  BSI06100
0611                     *                                                       BSI06110
0612                     *     IF THE SECOND ARGUMENT IS LARGER, SAVE THE SECOND ARGUMENT'S  BSI06120
0613                     *     MANTISSA AND LOAD THE FIRST ARGUMENT'S MANTISSA   BSI06130
0614                     *                                                       BSI06140
0615 00334   0 13 00355  A6   IMA   TEMP          SAVE SHIFT COUNT, RECOVER HIGH SECOND  BSI06150
0616                     *                       ARGUMENT                        BSI06160
0617 00335   0 13 00200       IMA   HIGH          EXCHANGE SO THAT LARGER ARGUMENT IS IN  BSI06170
0618 00336   000201           IAB                 HIGH AND LOW, AND SMALLER ARGUMENT IN  BSI06180
0619 00337   0 13 00201       IMA   LOW           THE A AND B REGISTERS           BSI06190
```

```
0620 00340   000201          IAB                                                  BS106200
0621 00341   0 13 00355       IMA   TEMP          RECOVER SHIFT COUNT            BS106210
0622                      *                                                      BS106220
0623                      *    FORM THE SHIFT INSRUCTION AND SHIFT THE SMALLER ARGUMENT'S MANTISSABS106230
0624                      *                                                      BS106240
0625 00342   0 11 00356 A1    CAS   M31           TEST IF DIFFERENCE OF EXPONENTS IS LESS BS106250
0626                      *                       THAN THE NUMBER OF SIGNIFICANT BITS PLUS 1 BS106260
0627 00343   0 03 00357       ANA   C77           YES - MASK SHIFT COUNT AND     BS106270
0628 00344   0 05 00360       ERA   LRS           FORM SHIFT INSTRUCTION         BS106280
0629 00345   100400           SPL                                                  BS106290
0630 00346   0 02 00712       LDA   LRSM          NO-FORM SHIFT INSTRUCTION TO SHIFT OUT ALL BS106300
0631                      *                       BITS EXCEPT THE SIGN           BS106310
0632 00347   0 04 00351       STA   A3                                           BS106320
0633 00350   0 02 00355       LDA   TEMP          LOAD HIGH OF SMALLER ARGUMENT  BS106330
0634 00351   0 00 00000 A3    PZE   **            SHIFT                          BS106340
0635                      *                                                      BS106350
0636                      *    ADD THE LARGER ARGUMENT'S MANTISSA AND PACK THE RESULT INTO BS106360
0637                      *    FLOATING POINT FORMAT                             BS106370
0638                      *                                                      BS106380
0639 00352   0 10 00155       JSI   DADD          ADD THE LARGER ARGUMENT MANTISSA BS106390
0640 00353   0 10 00202       JSI   NORM          PACK INTO FLOATING POINT FORMAT BS106400
0641 00354   -0 01 00120      JMP*  LHIS+1        ADDITION, SUBTRACTION RETURN   BS106410
0642                      *                                                      BS106420
0643                      *                                                      BS106430
0644 00355   000000       TEMP BSZ   1                                          BS106440
0645 00356   177741       M31  OCT   -31                                        BS106450
0646 00357   000077       C77  OCT   77                                         BS106460
0647 00360   040100       LRS  OCT   40100                                      BS106470
0648                           EJCT                                             BS106480
```

```
0649                    *                                                                       BSI06490
0650                    *                                                                       BSI06500
0651                    *          FLOATING POINT DIVISION                                      BSI06510
0652                    *                                                                       BSI06520
0653                    *     CALLING SEQUENCE:                                                 BSI06530
0654                    *                                                                       BSI06540
0655                    *          JSI    D$22          DIVIDEND IN A AND B REGISTERS           BSI06550
0656                    *          DAC    ARG2          POINTER TO DIVISOR                      BSI06560
0657                    *          ......RETURN        QUOTIENT IN A AND B REGISTER            BSI06570
0658                    *                                                                       BSI06580
0659                    *                                                                       BSI06590
0660                    *          THE ARGUMENTS ARE SET POSITIVE AND UNPACKED.  THE RESULTING  BSI06600
0661                    *     SIGN OF THE QUOTIENT IS SAVED.  THE EXPONENT OF THE DIVISOR IS SUB-BSI06610
0662                    *     TRACTED FROM THE EXPONENT OF THE DIVIDEND, AND THE BIAS IS        BSI06620
0663                    *     ADJUSTED TO FORM THE QUOTIENT EXPONENT.  THE MANTISSA OF THE      BSI06630
0664                    *     DIVIDEND IS COMPARED WITH THE MANTISSA OF THE DIVISOR.  IF THE    BSI06640
0665                    *     DIVIDEND MANTISSA IS GREATER OR EQUAL, THE TWO'S COMPLIMENT OF    BSI06650
0666                    *     THE DIVISOR MANTISSA IS ADDED TO THE DIVIDEND MANTISSA, AND A ONE BSI06660
0667                    *     QUOTIENT BIT IS GENERATED.  IF THE DIVIDEND MANTISSA IS LESS, A   BSI06670
0668                    *     ZERO QUOTIENT BIT IS GENERATED.  THE MANTISSAS OF THE DIVIDEND    BSI06680
0669                    *     AND THE QUOTIENT ARE THEN SHIFTED LEFT ONE PLACE.  THIS PROCESS   BSI06690
0670                    *     IS REPEATED UNTIL 24 QUOTIENT BITS HAVE BEEN GENERATED.  THE      BSI06700
0671                    *     QUOTIENT IS PACKED INTO FLOATING POINT FORMAT, AND THE SIGN IS    BSI06710
0672                    *     ADJUSTED.  IF THE SECOND ARGUMENT IS FOUND TO BE ZERO, A DZ ERROR BSI06720
0673                    *     IS FLAGGED.                                                       BSI06730
0674                    *                                                                       BSI06740
0675                    *                                                                       BSI06750
0676 00361  0 000000  D$22 DAC  **          DIVISION ENTRY                                      BSI06760
0677 00362  0 10 00455     JSI  MDAH        JUMP TO ARGUMENT HANDLING ROUTINE                   BSI06770
0678 00363  101040        SNZ              SKIP IF DIVISOR IS NON-ZERO                         BSI06780
0679 00364  0 01 00446     JMP  D6          DIVISOR IS ZERO-JUMP TO FLAG ERROR                  BSI06790
0680 00365  0401 77        LRS  1           POSITION DIVISOR MANTISSA                           BSI06800
0681 00366  0 10 00104     JSI  H$22        AND SAVE                                            BSI06810
0682 00367  0 000450      DAC  HGHC        X                                                   BSI06820
0683 00370  0 10 00122     JSI  N$22        TWO'S COMPLIMENT DIVISOR MANTISSA                   BSI06830
0684 00371  0 04 00200     STA  HIGH        SAVE HIGH PART                                      BSI06840
0685 00372  000201         IAB              CLEAR BIT ONE OF LOW MANTISSA BEFORE                BSI06850
```

```
0686 00373   140100          SSP                     STORING                                  BS106860
0687                    *                                                                    BS106870
0688                    *    QUOTIENT EXPONENT DETERMINATION                                 BS106880
0689                    *                                                                    BS106890
0690 00374   0 13 00201     IMA   LOW               SAVE LOW, AND RECOVER DIVIDEND EXPONENT  BS106900
0691 00375   0 07 00152     SUB   EXP1              SUBTRACT EXPONENT OF DIVISOR             BS106910
0692 00376   0 06 00454     ADD   C205              ADD BIAS PLUS SCALING FACTOR             BS106920
0693 00377   0 04 00152     STA   EXP1              SAVE EXPONENT OF QUOTIENT                BS106930
0694 00400   0 02 00356     LDA   M31               INITIALIZE LOOP COUNTER SO THAT NUMBER   BS106940
0695 00401   0 04 00272     STA   CNTR              OF QUOTIENT BITS GENERATED EQUALS NUMBER BS106950
0696                    *                           OF SIGNIFICANTS BITS PLUS ONE            BS106960
0697                    *                                                                    BS106970
0698                    *    INITIALIZE THE QUOTIENT MANTISSA TO ZERO AND LOAD THE DIVIDEND  BS106980
0699                    *    MANTISSA                                                        BS106990
0700                    *                                                                    BS107000
0701 00402   0 10 00100     JST   L$22              LOAD FLOATING POINT ZERO                 BS107010
0702 00403   0 001060       DAC   F0                X                                        BS107020
0703 00404   0 13 00453     IMA   RSLT+1            AND SAVE AS QUOTIENT, RECOVER DIVIDEND   BS107030
0704 00405   000201         IAB                     MANTISSA                                 BS107040
0705 00406   0 13 00452     IMA   RSLT              X                                        BS107050
0706 00407   0401 77        LRS   1                 POSITION DIVIDEND MANTISSA               BS107060
0707 00410   0 01 00414     JMP   *+4               JUMP TO COMPARE DIVIDEND AND DIVISOR     BS107070
0708                    *                           MANTISSAS                                BS107080
0709                    *                                                                    BS107090
0710                    *    LOAD THE DIVIDEND (REMAINDER) MANTISSA                          BS107100
0711                    *                                                                    BS107110
0712 00411   0 13 00453 D5  IMA   RSLT+1            SAVE QUOTIENT MANTISSA, RECOVER DIVIDEND BS107120
0713 00412   000201         IAB                     (REMAINDER) MANTISSA                     BS107130
0714 00413   0 13 00452     IMA   RSLT              X                                        BS107140
0715                    *                                                                    BS107150
0716                    *    COMPARE DIVISOR MANTISSA WITH DIVIDEND(REMAINDER) MANTISSA      BS107160
0717                    *                                                                    BS107170
0718 00414   0 11 00450     CAS   HGHC              COMPARE HIGH MANTISSAS                   BS107180
0719 00415   0 01 00441     JMP   D2+1               DIVIDEND(REMAINDER) GREATER              BS107190
0720 00416   100000         SKP                     EQUAL-SKIP TO COMPARE LOW MANTISSAS      BS107200
0721 00417   0 01 00425     JMP   D3                DIVISOR MANTISSA GREATER                 BS107210
0722 00420   000201         IAB                     LOAD LOW DIVIDEND(REMAINDER) MANTISSA    BS107220
```

```
0723 00421   0 11 00451       CAS   LOWC              COMPARE LOW MANTISSAS                          BS107230
0724 00422   101000           NOP                     X                                              BS107240
0725 00423   0 01 00440       JMP   D2                DIVIDEND MANTISSA GREATER OR EQUAL              BS107250
0726 00424   000201           IAB                     DIVISOR GREATER-REPOSITION                     BS107260
0727                  *                                                                              BS107270
0728                  *   HERE IF DIVIDEND(REMAINDER) MANTISSA IS LESS THAN DIVISOR MANTISSA          BS107280
0729                  *                                                                              BS107290
0730 00425   0411 77     D3    LLS   1                 SHIFT DIVIDEND(REMAINDER) MANTISSA              BS107300
0731 00426   0 13 00452       IMA   RSLT                LOAD LOW QUOTIENT MANTISSA                     BS107310
0732                  *                                                                              BS107320
0733                  *   SHIFT QUOTIENT MANTISSA AND INCREMENT COUNTER                               BS107330
0734                  *                                                                              BS107340
0735 00427   000201      D4    IAB                     LOAD LOW QUOTIENT MANTISSA IN B REGISTER       BS107350
0736 00430   0 13 00453       IMA   RSLT+1              LOAD HIGH QUOTIENT MANTISSA                    BS107360
0737 00431   0411 77          LLS   1                 SHIFT THE QUOTIENT MANTISSA                    BS107370
0738 00432   0 12 00272       IRS   CNTR                INCREMENT THE COUNTER BY ONE                   BS107380
0739 00433   0 01 00411       JMP   D5                LOOP TO GENERATE ANOTHER QUOTIENT BIT          BS107390
0740                  *                                 IF COUNTER IS NON-ZERO                         BS107400
0741                  *                                                                              BS107410
0742                  *   PACK AND SET SIGN OF QUOTIENT                                              BS107420
0743                  *                                                                              BS107430
0744 00434   0 10 00202       JSI   NORM                PACK INTO FLOATING POINT FORMAT (IF COUNTER BS107440
0745                  *                                 EQUALS ZERO)                                   BS107450
0746 00435   0 12 00355       IRS   TEMP                SKIP IF SIGN OF QUOTIENT IS POSITIVE           BS107460
0747 00436   0 10 00122       JSI   N$22                COMPLIMENT IF SIGN OF QUOTIENT IS NEGATIVE     BS107470
0748 00437  -0 01 00361       JMP*  D$22                DIVISION RETURN                                BS107480
0749                  *                                                                              BS107490
0750                  *   HERE IF DIVISOR MANTISSA IS LESS THAN OR EQUAL TO DIVIDEND                  BS107500
0751                  *   (REMAINDER) MANTISSA                                                       BS107510
0752                  *                                                                              BS107520
0753 00440   000201      D2    IAB                     REPOSITION                                     BS107530
0754 00441   0 10 00155       JSI   DADD                SUBTRACT DIVISOR MANTISSA FROM DIVIDEND         BS107540
0755                  *                                 (REMAINDER) MANTISSA                           BS107550
0756 00442   0411 77          LLS   1                 SHIFT REMAINDER MANTISSA                       BS107560
0757 00443   0 13 00452       IMA   RSLT                LOAD LOW QUOTIENT MANTISSA                     BS107570
0758 00444   141206           AOA                     GENERATE A ONE QUOTIENT BIT                    BS107580
0759 00445   0 01 00427       JMP   D4                JUMP TO SHIFT QUOTIENT                         BS107590
```

```
0760                        *                                                                   BSI07600
0761                        *     FLAG ERROR - DIVISION BY ZERO                                 BSI07610
0762                        *                                                                   BSI07620
0763 00446   0 10 00000 D6  JSI   ERR              JUMP TO ERROR REPORTING ROUTINE            BSI07630
0764 00447   142332         BCI   1,DZ             ID THE ERROR                               BSI07640
0765                        *                                                                   BSI07650
0766                        *                                                                   BSI07660
0767 00450   000000     HGHC BSZ   1                                                           BSI07670
0768 00451   000000     LOWC BSZ   1                                                           BSI07680
0769 00452   000000     RSLT BSZ   2                                                           BSI07690
0770 00454   000205     C205 DCI   205                                                         BSI07700
0771                        EJCT                                                              BSI07710
```

```
0772                    *                                                                   BSI07720
0773                    *                                                                   BSI07730
0774                    *          MULTIPLICATION AND DIVISION ARGUMENT HANDLING ROUTINE    BSI07740
0775                    *                                                                   BSI07750
0776                    *     CALLING SEQUENCE:                                             BSI07760
0777                    *                                                                   BSI07770
0778                    *          JSI    MDAH          CALL TO ROUTINE IMMEDIATELY FOLLOWS ENTRY, BSI07780
0779                    *                               FIRST ARGUMENT IN THE A AND B REGISTERS BSI07790
0780                    *          ......RETURN         MANTISSA OF ABSOLUTE VALUE OF FIRST ARGU- BSI07800
0781                    *                               MENT IN THE A AND B REGISTERS, ABSOLUTE BSI07810
0782                    *                               VALUE OF UNPACKED FIRST ARGUMENT AND FLAG BSI07820
0783                    *                               FOR SIGN OF RESULTANT IN TEMPORARY STORAGE BSI07830
0784                    *                               LOCATIONS                           BSI07840
0785                    *                                                                   BSI07850
0786                    *                                                                   BSI07860
0787                    *          THE SIGN OF THE FIRST ARGUMENT IS SAVED.  THE FIRST ARGUMENT BSI07870
0788                    *     IS SET POSITIVE, UNPACKED, AND SAVED.  THE SECOND ARGUMENT IS  BSI07880
0789                    *     LOADED, AND ITS SIGN XORED WITH THE SIGN OF THE FIRST ARGUMENT. BSI07890
0790                    *     IF THE RESULTANT SIGN IS POSITIVE, A FLAG IS SET TO NEGATIVE ONE. BSI07900
0791                    *     IF THE RESULTANT SIGN IS NEGATIVE, THE FLAG IS SET TO A POSITVE BSI07910
0792                    *     NUMBER.  THE SECOND ARGUMENT IS SET POSITIVE AND UNPACKED.    BSI07920
0793                    *                                                                   BSI07930
0794                    *                                                                   BSI07940
0795 00455  0 000000  MDAH DAC   **              ENTRY                                      BSI07950
0796 00456  0 04 00355      STA   TEMP            SAVE SIGN OF FIRST ARGUMENT                BSI07960
0797                    *                                                                   BSI07970
0798                    *     SET FIRST ARGUMENT POSITIVE, UNPACK, AND SAVE IT              BSI07980
0799                    *                                                                   BSI07990
0800 00457  100400           SPL                   SKIP IF POSITIVE                        BSI08000
0801 00460  0 10 00122       JSI   N$22            COMPLIMENT IF NEGATIVE                  BSI08010
0802 00461  0 10 00134       JSI   UNPK            UNPACK THE FIRST ARGUMENT               BSI08020
0803 00462  0 10 00104       JSI   H$22            SAVE THE MANTISSA                       BSI08030
0804 00463  0 000452         DAC   RSLT            X                                       BSI08040
0805 00464  0 02 00152       LDA   EXPT            AND THE EXPONENT                        BSI08050
0806 00465  0 04 00201       STA   LOA             X                                       BSI08060
0807                    *                                                                   BSI08070
0808                    *     LOAD SECOND ARGUMENT                                          BSI08080
```

```
0809                        *                                                          BS108090
0810 00466   0 02 00455       LDA    MDAH          LOAD ADDRESS PLUS ONE OF CALL       BS108100
0811 00467   0 07 00506       SUB    C2            SUBTRACT TWO AND                    BS108110
0812 00470   0 04 00272       STA    CNTR          SAVE ADDRESS OF CALLING ROUTINE ENTRY BS108120
0813 00471  -0 02 00272       LDA*   CNTR          LOAD ADDRESS OF DAC TO SECOND ARGUMENT BS108130
0814 00472  -0 12 00272       IRS*   CNTR          INCREMENT TO SET UP RETURN ADDRESS FOR BS108140
0815                        *                       CALLING ROUTINE                    BS108150
0816 00473   0 10 00070       JSI    LARG          LOAD SECOND ARGUMENT                BS108160
0817                        *                                                          BS108170
0818                        *     DETERMINE SIGN OF RESULT AND SET FLAG                BS108180
0819                        *                                                          BS108190
0820 00474   0 13 00355       IMA    TEMP          RECOVER SIGN OF FIRST ARGUMENT      BS108200
0821 00475   0 05 00355       ERA    TEMP          XOR SIGN OF SECOND ARGUMENT         BS108210
0822 00476   140320           CSA                  CLEAR A1 SO THAT NEGATIVE ONE ISN'T BS108220
0823                        *                       GENERATED INADVERTENTLY            BS108230
0824 00477   101001           SSC                  SKIP IF RESULTANT SIGN IS NEGATIVE  BS108240
0825 00500   0 02 00000       LDA    M1            LOAD NEGATIVE ONE IF RESULTANT SIGN IS BS108250
0826                        *                       POSITIVE                           BS108260
0827 00501   0 13 00355       IMA    TEMP          SAVE RESULTANT SIGN FLAG, RECOVER HIGH BS108270
0828                        *                       WORD OF SECOND ARGUMENT            BS108280
0829                        *                                                          BS108290
0830                        *     SET SECOND ARGUMENT POSITIVE AND UNPACK IT           BS108300
0831                        *                                                          BS108310
0832 00502   100400           SPL                  SKIP IF POSITIVE                    BS108320
0833 00503   0 10 00122       JSI    N$22          COMPLIMENT IF NEGATIVE              BS108330
0834 00504   0 10 00134       JSI    UNPK          UNPACK THE SECOND ARGUMENT          BS108340
0835 00505  -0 01 00455       JMP*   MDAH          RETURN                              BS108350
0836                        *                                                          BS108360
0837 00506   000002     C2    OCT    2                                                 BS108370
0838                        EJCT                                                       BS108380
```

```
0839                    *                                                                       BSI08390
0840                    *                                                                       BSI08400
0841                    *          SINGLE PRECISION MULTIPLIPLICATION                           BSI08410
0842                    *                                                                       BSI08420
0843                    *     CALLING SEQUENCE:                                                 BSI08430
0844                    *                                                                       BSI08440
0845                    *          JSI    M$11          MULTIPLICAND IN THE A REGISTER          BSI08450
0846                    *          DAC    ARG2          POINTER TO MULTIPLIER                   BSI08460
0847                    *          ......RETURN         PRODUCT TRANSPOSED SO THAT LEAST SIGNIFI-BSI08470
0848                    *                                CANT PART IS IN A REGISTER              BSI08480
0849                    *                                                                       BSI08490
0850                    *                                                                       BSI08500
0851                    *          THE ROUTINE EXPECTS THE MULTIPLICAND AND MULTIPLIER TO BE IN BSI08510
0852                    *     SINGLE PRECISION FIXED POINT FORMAT WITH POSITIVE SIGNS.  THE LEASTBSI08520
0853                    *     SIGNIFICANT BIT OF THE MULTIPLIER IS TESTED.  IF IT IS ONE, THE   BSI08530
0854                    *     MULTIPLICAND IS ADDED TO THE PRODUCT WHICH IS INITIALLY ZERO.  THE BSI08540
0855                    *     MULTIPLIER AND THE PRODUCT ARE SHIFTED RIGHT, AND THE NEXT LEAST  BSI08550
0856                    *     SIGNIFICANT BIT OF THE MULTIPLIER IS TESTED.  THE PROCESS IS      BSI08560
0857                    *     REPEATED FOR THE 15 MAGNITUDE BITS OF THE MULTIPLIER.  THE RESULT BSI08570
0858                    *     IS THEN SHIFTED SO THAT IT IS DOUBLE PRECISION FIXED POINT FORMAT. BSI08580
0859                    *     THE ANSWER IS TRANSPOSED SO THAT THE LEAST SIGNIFICANT PART IS IN BSI08590
0860                    *     THE A REGISTER, AND THE ROUTINE EXITS.                            BSI08600
0861                    *                                                                       BSI08610
0862                    *                                                                       BSI08620
0863 00507   0 000000   M$11 DAC    **                 INTEGER MULTIPLY ENTRY                   BSI08630
0864 00510   0 04 00200      STA    HIGH               SAVE THE MULTIPLICAND                    BSI08640
0865                    *                                                                       BSI08650
0866                    *     INITIALIZE THE LOOP COUNTER AND POSITION THE MULTIPLIER           BSI08660
0867                    *                                                                       BSI08670
0868 00511   0 02 00527      LDA    M17                INITIALIZE LOOP COUNTER TO THE NUMBER OF BSI08680
0869                    *                               MAGNITUDE BITS IN MULTIPLIER            BSI08690
0870 00512   0 04 00272      STA    CNTR                                                        BSI08700
0871 00513   0 02 00507      LDA    M$11               LOAD MULTIPLIER                          BSI08710
0872 00514   0 10 00070      JSI    LARG               LOAD ARGUMENT ROUTINE                    BSI08720
0873 00515   0400 57         LRL    17                 CLEAR A, SHIFT LSB OF MULTIPLIER INTO C BIBSI08730
0874                    *                                                                       BSI08740
0875                    *     MULTIPLY LOOP                                                     BSI08750
```

```
0876                        *                                                          BSI08760
0877 00516   100001         M101 SRC                  TEST LSB                         BSI08770
0878 00517   0 06 00200          ADD    HIGH          LSB IS ONE-ADD THE MULTIPLICAND TO THE PRO-BSI08780
0879                        *                         DUCT                             BSI08790
0880 00520   0400 11             LRL    1             SHIFT LSB OF MULTIPLIER INTO C BIT BSI08800
0881 00521   0 12 00272         IRS    CNTR          BUMP COUNTER                     BSI08810
0882 00522   0 01 00516         JMP    M101          LOOP TO TEST LSB OF MULTIPLIER IF COUNTER BSI08820
0883                        *                         IS NON-ZERO                      BSI08830
0884                        *                                                          BSI08840
0885                        *    FORMAT THE RESULT                                     BSI08850
0886                        *                                                          BSI08860
0887 00523   0414 11             LGL    1             IF THE COUNTER IS ZERO, PUT THE RESULT IN BSI08870
0888 00524   0400 11             LRL    1             DOUBLE PRECISION FORMAT BY CLEARING BIT ONEBSI08880
0889                        *                         OF THE SECOND WORD               BSI08890
0890 00525   000201             IAB                  TRANSPOSE THE HIGH AND LOW WORDS BSI08900
0891 00526   -0 01 00120        JMP*   LHIS+1        INTEGER MULTIPLY EXIT            BSI08910
0892                        *                                                          BSI08920
0893                        *                                                          BSI08930
0894 00527   177761         M17  OCT    -17                                           BSI08940
0895                             EJCT                                                  BSI08950
```

```
0896                    *                                                                       BS108960
0897                    *                                                                       BS108970
0898                    *          FLOATING POINT MULTIPLICATION                                BS108980
0899                    *                                                                       BS108990
0900                    *     CALLING SEQUENCE:                                                 BS109000
0901                    *                                                                       BS109010
0902                    *          JSI    M$22          MULTIPLICAND IN A AND B REGISTERS       BS109020
0903                    *          DAC    ARG2          POINTER TO MULTIPLIER                   BS109030
0904                    *          ......RETURN         PRODUCT IN A AND B REGISTER             BS109040
0905                    *                                                                       BS109050
0906                    *                                                                       BS109060
0907                    *          THE ARGUMENTS ARE UNPACKED, THEIR SIGNS ARE SET POSITIVE, ANDBS109070
0908                    *     THE SIGN OF THE PRODUCT IS SAVED.  THEIR EXPONENTS ARE ADDED, THE BS109080
0909                    *     BIAS IS ADJUSTED, AND THE RESULT IS SAVED AS THE EXPONENT OF THE  BS109090
0910                    *     PRODUCT.  THE HIGH MULTIPLICAND MANTISSA AND THE LOW MULTIPLIER   BS109100
0911                    *     MANTISSA ARE MULTIPLIED, AS ARE THE HIGH MULTIPLIER MANTISSA AND  BS109110
0912                    *     LOW MULTIPLICAND MANTISSA.  THE TWO HIGH CROSS PRODUCT ARE SAVED. BS109120
0913                    *     THE HIGH MULTIPLICAND MANTISSA AND THE HIGH MULTIPLIER MANTISSA AREBS109130
0914                    *     MULTIPLIED, AND THE HIGH CROSS PRODUCTS ARE ADDED TO THE LOW PRO- BS109140
0915                    *     DUCT.  IF THERE IS OVERFLOW, THE HIGH PRODUCT IS INCREMENTED.     BS109150
0916                    *     THE PRODUCT IS PACKED INTO FLOATING POINT FORMAT, AND THE SIGN IS  BS109160
0917                    *     ADJUSTED.                                                         BS109170
0918                    *                                                                       BS109180
0919                    *                                                                       BS109190
0920 00530   0 000000   M$22 DAC    **                 MULTIPLICATION ENTRY                    BS109200
0921 00531   0 10 00455      JSI    MDAH               JUMP TO ARGUMENT HANDLING ROUTINE       BS109210
0922                    *                                                                       BS109220
0923                    *     PRODUCT EXPONENT DETERMINATION                                    BS109230
0924                    *                                                                       BS109240
0925 00532   0 13 00201      IMA    LOW                SAVE HIGH MULTIPLIER MANTISSA, RECOVER  BS109250
0926                    *                               EXPONENT OF MULTIPLICAND                BS109260
0927 00533   0 06 00152      ADD    EXPI               ADD EXPONENT OF MULTIPLIER              BS109270
0928 00534   0 07 00572      SUB    C200               SUBTRACT BIAS                           BS109280
0929 00535   0 04 00152      STA    EXPI               SAVE RESULTANT EXPONENT                 BS109290
0930                    *                                                                       BS109300
0931                    *     PRODUCT MANTISSA DETERMINATION                                    BS109310
0932                    *                                                                       BS109320
```

```
0933 00536   000201          IAB                LOAD LOW MULTIPLIER MANTISSA                    BSI09330
0934 00537   0 10 00507      JSI    M$11        MULTIPLY BY HIGH MULTIPLICAND MANTISSA          BSI09340
0935 00540   0 000452        DAC    RSL1        X                                               BSI09350
0936 00541   000201          IAB                LOAD HIGH CROSS PRODUCT                         BSI09360
0937 00542   0 13 00453      IMA    RSL1+1      AND SAVE, RECOVER LOW MULTIPLICAND MANTISSA     BSI09370
0938 00543   0 10 00507      JSI    M$11        MULTIPLY BY HIGH MULTIPLIER MANTISSA            BSI09380
0939 00544   0 000201        DAC    LOW         X                                               BSI09390
0940 00545   000201          IAB                LOAD HIGH CROSS PRODUCT                         BSI09400
0941 00546   0 13 00201      IMA    LOW         AND SAVE, RECOVER HIGH MULTIPLIER MANTISSA      BSI09410
0942 00547   0 10 00507      JSI    M$11        MULTIPLY BY HIGH MULTIPLICAND MANTISSA          BSI09420
0943 00550   0 000452        DAC    RSL1        X                                               BSI09430
0944 00551   000201          IAB                SAVE HIGH PRODUCT MANTISSA                      BSI09440
0945 00552   0 04 00452      STA    RSL1        X                                               BSI09450
0946 00553   000201          IAB                LOAD LOW PRODUCT MANTISSA                       BSI09460
0947 00554   0 06 00201      ADD    LOW         ADD CROSS PRODUCT                               BSI09470
0948 00555   140320          CSA                OVERFLOW INTO C BIT                             BSI09480
0949 00556   100001          SRC                SKIP IF C BIT RESET                             BSI09490
0950 00557   0 12 00452      IRS    RSL1        INCREMENT HIGH IF C BIT IS SET                  BSI09500
0951 00560   0 06 00453      ADD    RSL1+1      ADD IN THE OTHER CROSS PRODUCT                  BSI09510
0952 00561   140320          CSA                OVERFLOW INTO C BIT                             BSI09520
0953 00562   100001          SRC                SKIP IF C BIT RESET                             BSI09530
0954 00563   0 12 00452      IRS    RSL1        INCREMENT HIGH IF C BIT IS SET                  BSI09540
0955 00564   000201          IAB                LOAD LOW PRODUCT MANTISSA IN THE B REGISTER     BSI09550
0956 00565   0 02 00452      LDA    RSL1        LOAD HIGH PRODUCT MANTISSA                      BSI09560
0957                 *                                                                          BSI09570
0958                 *     PACK AND SET SIGN OF PRODUCT                                         BSI09580
0959                 *                                                                          BSI09590
0960 00566   0 10 00202      JSI    NORM        PACK PRODUCT INTO FLOATING POINT FORMAT         BSI09600
0961 00567   0 12 00355      IRS    TEMP        SKIP IF SIGN OF PRODUCT IS POSITIVE             BSI09610
0962 00570   0 10 00122      JSI    N$22        COMPLIMENT IF SIGN OF PRODUCT IS NEGATIVE       BSI09620
0963 00571   -0 01 00530     JMP*   M$22        MULTIPLICATION RETURN                           BSI09630
0964                 *                                                                          BSI09640
0965                 *                                                                          BSI09650
0966 00572   000200     C200 OCT    200                                                         BSI09660
0967                      EJCT                                                                  BSI09670
```

```
0968                        *                                                                   BS109680
0969                        *          INTEGER TO FLOATING POINT CONVERSION                     BS109690
0970                        *                                                                   BS109700
0971                        *    CALLING SEQUENCE:                                              BS109710
0972                        *                                                                   BS109720
0973                        *          JSI   FINI          ASSUMES INTEGER IN A REGISTER       BS109730
0974                        *          ......RETURN        FLOATING POINT IN A AND B REGISTERS BS109740
0975                        *                                                                   BS109750
0976                        *                                                                   BS109760
0977                        *          THE INTEGER IS PUT IN DOUBLE PERCISION FIXED POINT FORM BY BS109770
0978                        *    CLEARING THE B REGISTER.  THE EXPONENT IS INITIALIZED AT '217 - BS109780
0979                        *    '200 FOR BIAS AND '17 TO MOVE THE BINARY POINT BETWEEN BITS 1 AND BS109790
0980                        *    2 OF THE A REGISTER.  THEN THE FLOATING POINT NORMALIZING ROUTINE BS109800
0981                        *    IS USED TO PACK IT INTO FLOATING POINT FORMAT.                  BS109810
0982                        *                                                                   BS109820
0983                        *                                                                   BS109830
0984 00573   0 000000   FINI DAC   **                  ENTRY                                    BS109840
0985 00574   0 04 00152      STA   EXPI                SAVE THE INTEGER                         BS109850
0986 00575   140040          CRA                       CLEAR THE B REGISTER                     BS109860
0987 00576   000201          IAB                                                                BS109870
0988 00577   0 02 00603      LDA   C217                INITIALIZE EXPONENT TO BIAS PLUS FACTOR   BS109880
0989                        *                         TO MOVE BINARY POINT TO THE END OF THE   BS109890
0990                        *                         A REGISTER                               BS109900
0991 00600   0 13 00152      IMA   EXPI                RECOVER THE INTEGER                      BS109910
0992 00601   0 10 00202      JSI   NORM                NORMALIZE TO CONVERT                     BS109920
0993 00602  -0 01 00573      JMP*  FINI                EXIT                                     BS109930
0994                        *                                                                   BS109940
0995                        *                                                                   BS109950
0996 00603   000217     C217 OCT   217                                                         BS109960
0997                        EJCT                                                               BS109970
```

```
0998                        *                                                                   BSI09980
0999                        *                                                                   BSI09990
1000                        *          FLOATING POINT TO INTGER CONVERSION                      BSI10000
1001                        *                                                                   BSI10010
1002                        *     CALLING SEQUENCE:                                             BSI10020
1003                        *                                                                   BSI10030
1004                        *          JSI    IFLI          ASSUMES FLOATING POINT IN A AND B REGISTERSBSI10040
1005                        *          ......OUT OF RANGE RETURN                                BSI10050
1006                        *          ......NORMAL RETURN - INTEGER IN THE A REGISTER          BSI10060
1007                        *                                                                   BSI10070
1008                        *                                                                   BSI10080
1009                        *          THE FLOATING POINT IS UNPACKED.  '217 IS SUBTRACTED FROM THE BSI10090
1010                        *     EXPONENT - '200 TO REMOVE THE BIAS,  AND '17 TO MOVE THE BINARY  BSI10100
1011                        *     POINT TO THE END OF THE A REGISTER.  IF THE RESULT IS GREATER THAN BSI10110
1012                        *     ZERO, THE FLOATING POINT IS OUT OF INTEGER RANGE, AND THE OUT OF  BSI10120
1013                        *     RANGE RETURN IS TAKEN.  IF THE RESULT IS LESS THAN OR EQUAL TO    BSI10130
1014                        *     ZERO, THE MANTISSA IS SHIFTED RIGHT TO FORM THE TRUNCATED INTEGER,BSI10140
1015                        *     AND THE NORMAL RETURN IS TAKEN                                BSI10150
1016                        *                                                                   BSI10160
1017                        *                                                                   BSI10170
1018 00604   0 000000   IFLI DAC   **          ENTRY                                            BSI10180
1019 00605   0 10 00134      JSI   UNPK        UNPACK THE FLOATING POINT                        BSI10190
1020 00606   0 13 00152      IMA   EXPI        LOAD THE EXPONENT                                BSI10200
1021 00607   0 07 00603      SUB   C217        SUBTRACT BIAS PLUS FACTOR TO MOVE THE            BSI10210
1022                        *                 BINARY POINT BETWEEN FIRST TWO BITS OF           BSI10220
1023                        *                 THE A REGISTER                                   BSI10230
1024 00610   0 11 01060      CAS   FO          TEST FOR OUT OF INTEGER RANGE                    BSI10240
1025 00611  -0 01 00604      JMP*  IFLI        OUT OF RANGE RETURN                              BSI10250
1026 00612   101000         NOP                                                                BSI10260
1027 00613   0 11 00527      CAS   M17         TEST FOR ALL FRACTIONAL BITS IN MANTISSA         BSI10270
1028 00614   0 03 00357      ANA   C77         NO - MASK SHIFT COUNT AND                        BSI10280
1029 00615   0 05 00625      ERA   ARS         FORM SHIFT INSTRUCTION                           BSI10290
1030 00616   100400         SPL               SKIP IF GREATER THAN '-17                        BSI10300
1031 00617   0 02 00626      LDA   ARSM        YES - SHIFT OUT ALL BITS EXCEPT SIGN             BSI10310
1032 00620   0 04 00622      STA   IF01                                                         BSI10320
1033 00621   0 02 00152      LDA   EXPI        RECOVER HIGH MANTISSA                            BSI10330
1034 00622   0405 00    IF01 ARS   **          SHIFT TO INTEGER FORMAT                          BSI10340
```

```
1035 00623   0 12 00604        IRS   IFLT                NORMAL RETURN                BSI10350
1036 00624  -0 01 00604        JMP*  IFLT                                             BSI10360
1037                      *                                                          BSI10370
1038                      *                                                          BSI10380
1039 00625   040500         ARS   OCT   40500                                        BSI10390
1040 00626   040517         ARSM  OCT   40517                                        BSI10400
1041                           EJCT                                                  BSI10410
```

```
1042                    *                                                                       BSI10420
1043                    *                                                                       BSI10430
1044                    *          TEST FOR INTEGER ROUTINE                                     BSI10440
1045                    *                                                                       BSI10450
1046                    *     CALLING SEQUENCE:                                                 BSI10460
1047                    *                                                                       BSI10470
1048                    *          JSI    TINT          ASSUMES FLOATING POINT IN A AND B REGISTERSBSI10480
1049                    *          ......INTEGER RETURN - INTEGER IN A REGISTER                 BSI10490
1050                    *          ......FLOATING POINT RETURN - FLOATING POINT IN A AND B      BSI10500
1051                    *                                 REGISTERS.  CONVERSION NOT MADE BECAUSE BSI10510
1052                    *                                 FLOATING POINT IS OUT OF INTEGER RANGE, BSI10520
1053                    *                                 OR CONTAINED FRACTIONAL BITS.         BSI10530
1054                    *                                                                       BSI10540
1055                    *                                                                       BSI10550
1056                    *          THIS ROUTINE CALLS UPON THE FLOATING POINT TO INTEGER CON-    BSI10560
1057                    *     VERSION.  IF OUT OF RANGE RETURN IS TAKEN, THE FLOATING POINT     BSI10570
1058                    *     RETURN IS MADE.  IF THE NORMAL RETURN IS TAKEN, THE INTEGER IS CON-BSI10580
1059                    *     VERTED TO FLOATING POINT AND SUBTRACTED FROM THE ORIGINAL FLOATING BSI10590
1060                    *     POINT.  IF THE RESULT IS ZERO, THE INTEGER RETURN IS MADE.  OTHER- BSI10600
1061                    *     WISE, THE FLOATING POINT RETURN IS MADE.                          BSI10610
1062                    *                                                                       BSI10620
1063                    *     NOTE: E$22 MAKES USE OF THE FACT THAT THE INTEGER WILL ALSO BE IN BSI10630
1064                    *     CTMP, AND THE FLOATING POINT WILL ALSO BE IN FTMP.                BSI10640
1065                    *                                                                       BSI10650
1066                    *                                                                       BSI10660
1067 00627  0 000000   TINT DAC   **              ENTRY                                         BSI10670
1068 00630  0 10 00104      JSI   H$22            SAVE THE FLOATING POINT ARGUMENT              BSI10680
1069 00631  0 001017       DAC   FTMP                                                          BSI10690
1070 00632  0 10 00604      JSI   IFLI            FLOATING POINT TO INTEGER CONVERSION          BSI10700
1071 00633  0 01 00644      JMP   I101            OUT OF RANGE RETURN - TAKE FLOATING POINT     BSI10710
1072                    *                         RETURN                                        BSI10720
1073 00634  0 04 00650      STA   CTMP            INTEGER RETURN-SAVE THE INTEGER               BSI10730
1074 00635  0 10 00573      JSI   FINI            CONVERT IT TO FLOATING POINT FORMAT           BSI10740
1075 00636  0 10 00277      JSI   S$22            SUBTRACT IT FROM ORIGINAL FLOATING POINT      BSI10750
1076 00637  0 001017       DAC   FTMP                                                          BSI10760
1077 00640  100040         SZE                   TEST ZERO DIFFERENCE                          BSI10770
1078 00641  0 01 00644      JMP   I101            NO - TAKE FLOATING POINT RETURN               BSI10780
```

```
1079 00642   0 02 00650       LDA   CIMP     YES-LOAD THE INTEGER                    BSI10790
1080 00643  -0 01 00627       JMP*  IINT     INTEGER RETURN                         BSI10800
1081 00644   0 10 00100 I101  JSI   L$22     LOAD ORIGINAL FLOATING POINT           BSI10810
1082 00645   0 001017        DAC   FIMP                                             BSI10820
1083 00646   0 12 00627       IRS   IINT     FLOATING POINT RETURN                  BSI10830
1084 00647  -0 01 00627       JMP*  IINT                                            BSI10840
1085                         *                                                      BSI10850
1086                         *                                                      BSI10860
1087 00650   000000      CIMP BSZ   1                                               BSI10870
1088                       EJCT                                                     BSI10880
```

```
                 *
                 *
                 *         GREATEST INTEGER FUNCTION
                 *
                 *    CALLING SEQUENCE:
                 *
                 *         JST    INTF
                 *         DAC    ARG           POINTER TO THE ARGUMENT
                 *         ******RETURN         RESULT IN A AND B REGISTERS
                 *
                 *
                 *         THE ARGUMENT IS LOADED AND UNPACKED.  THE BIAS PLUS THE NUM-
                 *    BER OF SIGNIFICANT BITS IS SUBTRACTED FROM THE EXPONENT TO COMPUTE
                 *    THE NUMBER OF FRACTIONAL BITS IN THE MANTISSA.  IF THERE ARE NO
                 *    FRACTIONAL BITS THE NUMBER IS REPACKED, AND THE ROUTINE EXITS.
                 *    OTHERWISE, THE FRACTIONAL BITS ARE SHIFTED OUT OF THE MANTISSA.
                 *    ZEROS ARE THEN SHIFTED IN TO REPLACE THE FRACTIONAL BITS.  THERE IS
                 *    A SPECIAL TEST SO THAT MINUS ONE IS GENERATED FOR ARGUMENT IN THE
                 *    RANGE OF MINUS ONE AND ZERO.
                 *
                 *
00651  0 000000  INTF DAC   **
00652  0 02 00651      LDA   INTF
00653  0 10 00070      JST   LARG               LOAD ARGUMENT ROUTINE
00654  0 10 00134      JST   UNPK               UNPACK THE FLOATING POINT
00655  0 04 00677      STA   IN03               SAVE HIGH
                 *
                 *    TEST FOR FRACTIONAL BITS
                 *
00656  0 02 00152      LDA   EXPT               LOAD THE EXPONENT
00657  0 06 00707      ADD   M227               SUBTRACT BIAS PLUS NUMBER OF SIGNIFICANT
                 *                              BITS IN THE MANTISSA
00660  100400          SPL                      SKIP IF POSITIVE
00661  0 01 00665      JMP   IN01               JUMP TO SHIFT OUT FRACTIONAL BITS
00662  0 02 00677      LDA   IN03               RECOVER HIGH
                 *
                 *    REPACK AND RETURN
```

```
1126                        *                                                                   BSI11260
1127 00663   0 10 00202 IN04 JST   NORM            REPACK                                      BSI11270
1128 00664  -0 01 00120       JMP*  LHIS+1         GREATEST INTEGER FUNCTION EXIT              BSI11280
1129                        *                                                                   BSI11290
1130                        *    FORM THE SHIFT INSTRUCTIONS AND SHIFT OUT THE MANTISSA'S        BSI11300
1131                        *    FRACTIONAL BITS                                                BSI11310
1132                        *                                                                   BSI11320
1133 00665   0 06 01165 IN01 ADD   M7              ADD -7 TO THE SHIFT COUNT TO RIGHT          BSI11330
1134                        *                       JUSTIFY MANTISSA                           BSI11340
1135 00666   0 11 00710       CAS   M36            TEST IF ALL MANTISSA BITS ARE FRACTIONAL    BSI11350
1136 00667   0 03 00357       ANA   C77            NO- MASK SHIFT COUNT AND                    BSI11360
1137 00670   0 05 00360       ERA   LRS            FORM SHIFT INSTRUCTION                      BSI11370
1138 00671   100400           SPL                                                              BSI11380
1139 00672   0 02 00712       LDA   LRSM           YES - SHIFT OUT ALL BITS EXCEPT SIGN        BSI11390
1140 00673   0 04 00676       STA   IN02           STORE RIGHT SHIFT                           BSI11400
1141 00674   0 05 00711       ERA   C1K            FORM LEFT SHIFT INSTRUCTION                 BSI11410
1142 00675   0 13 00677       IMA   IN03           AND STORE, RECOVER HIGH                     BSI11420
1143 00676   0401 00     IN02 LRS   **             SHIFT OUT FRACTIONAL BITS                   BSI11430
1144 00677   0411 00     IN03 LLS   **             RESTORE FORMAT                              BSI11440
1145                        *                                                                   BSI11450
1146                        *    BE SURE THAT NEGATIVE ONE IS RETURNED FOR ARGUMENTS IN THE     BSI11460
1147                        *    RANGE OF NEGATIVE ONE AND ZERO                                 BSI11470
1148                        *                                                                   BSI11480
1149 00700   0 11 00276       CAS   MLNN           SPECIAL CHECK SO THAT -1 IS GENERATED FOR   BSI11490
1150                        *                       NUMBERS BETWEEN -1 AND 0                   BSI11500
1151 00701   0 01 00663       JMP   IN04           JUMP TO REPACK                              BSI11510
1152 00702   0 13 00677       IMA   IN03           LOAD THE SHIFT INSTRUCTION                  BSI11520
1153 00703   0 11 00713       CAS   LLSM           TEST IF ALL MANTISSA BITS WERE SHIFTED OUT  BSI11530
1154 00704   0 01 00662       JMP   IN04-1         NO - JUMP TO REPACK                         BSI11540
1155 00705   0 02 00000       LDA   FM1            YES - LOAD -1                               BSI11550
1156 00706  -0 01 00120       JMP*  LHIS+1         AND RETURN                                  BSI11560
1157                        *                                                                   BSI11570
1158                        *                                                                   BSI11580
1159 00707   177551     M227 OCT   -227                                                        BSI11590
1160 00710   177742     M36  OCT   -36                                                         BSI11600
1161 00711   001000     C1K  OCT   1000                                                        BSI11610
1162 00712   040142     LRSM OCT   40142                                                       BSI11620
```

```
1163 00713    041142      LLSM UCI    41142                          BS111630
1164                            EJCI                                 BS111640
```

```
1165                    *                                                                       BSI11650
1166                    *                                                                       BSI11660
1167                    *          FLOATING POINT EXPONENTIATION                                BSI11670
1168                    *                                                                       BSI11680
1169                    *     CALLING SEQUENCE:                                                 BSI11690
1170                    *                                                                       BSI11700
1171                    *          JSI    E$22          FIRST ARGUMENT IN A AND B REGISTERS     BSI11710
1172                    *          DAC    ARG2          POINTER TO SECOND ARGUMENT              BSI11720
1173                    *          ......RETURN         FIRST ARGUMENT RAISED TO THE SECOND ARGU- BSI11730
1174                    *                               MENT POWER IN THE A AND B REGISTERS     BSI11740
1175                    *                                                                       BSI11750
1176                    *                                                                       BSI11760
1177                    *          THE FIRST ARGUMENT IS SAVED, AND THE RESULT IS INITIALIZED TOBSI11770
1178                    *     ONE.  THE SECOND ARGUMENT IS LOADED, AND IF IT IS AN INTEGER WHOSE BSI11780
1179                    *     ABSOLUTE VALUE IS LESS THAN 64, THE SIGN OF THE SECOND ARGUMENT IS BSI11790
1180                    *     SAVED, AND THE LEAST SIGNIFICANT BIT OF THE ABSOLUTE VALUE OF THE  BSI11800
1181                    *     SECOND ARGUMENT IS TESTED.  IF IT IS ONE, THE RESULT IS MULTIPLIED BSI11810
1182                    *     BY THE FIRST ARGUMENT.  THE ABSOLUTE VALUE OF THE SECOND ARGUMENT  BSI11820
1183                    *     IS SHIFTED RIGHT ONE PLACE AND TESTED EQUAL TO ZERO.  IF NON-ZERO, BSI11830
1184                    *     FIRST ARGUMENT IS REPLACED BY ITS SQUARE, AND THE ROUTINE LOOPS TO BSI11840
1185                    *     TEST THE LEAST SIGNIFICANT BIT OF THE SECOND ARGUMENT.  IF ZERO    BSI11850
1186                    *     THE SIGN OF SECOND ARGUMENT IS TESTED, AND IF NEGATIVE, THE RESULT BSI11860
1187                    *     IS REPLACED BY ITS RECIPROCAL.  THE ROUTINE EXITS.  IF THE SECOND  BSI11870
1188                    *     ARGUMENT IS NOT AN INTEGER OR AN INTEGER WHOSE ABSOLUTE VALUE IS   BSI11880
1189                    *     GREATER THAN 63, THE EXPONENTIATION IS ACCOMPLISHED BY TAKING THE  BSI11890
1190                    *     NATURAL ANTILOGARITHM OF THE PRODUCT OF THE SECOND ARGUMENT AND    BSI11900
1191                    *     NATURAL LOGARITHM OF THE FIRST ARGUMENT.  (EXP(ARG2*LN(ARG1)))     BSI11910
1192                    *     THERE IS A SPECIAL CHECK SO THAT IF THE FIRST ARGUMENT IS ZERO AND BSI11920
1193                    *     SECOND ARGUMENT IS GREATER THAN ZERO, A ZERO IS RETURNED; AND IF   BSI11930
1194                    *     THE SECOND ARGUMENT IS LESS THAN ZERO, A DZ ERROR IS FLAGGED.      BSI11940
1195                    *                                                                       BSI11950
1196                    *                                                                       BSI11960
1197 00714   0 000000   E$22 DAC    **                ENTRY                                     BSI11970
1198 00715   0 10 00104       JSI    H$22              STORE FIRST ARGUMENT                      BSI11980
1199 00716   0 001021         DAC    EIMP                                                        BSI11990
1200 00717   0 10 00100       JSI    L$22              INITALIZE THE RESULT TO ONE               BSI12000
1201 00720   0 001144         DAC    F1                                                          BSI12010
```

```
1202 00721    0 10 00104        JSI   H$22                                    BSI12020
1203 00722    0 001221          DAC   Z0                                      BSI12030
1204 00723    0 02 00714        LDA   E$22       LOAD THE EXPONENT            BSI12040
1205 00724    0 10 00070        JSI   LARG                                    BSI12050
1206 00725    0 12 00714        IRS   E$22       BUMP FOR RETURN              BSI12060
1207 00726    0 10 00627        JSI   IINT       TEST FOR INTEGER             BSI12070
1208 00727    0 01 00751        JMP   E$01       INTEGER - JUMP               BSI12080
1209                    *                                                     BSI12090
1210                    *    HERE IF THE EXPONENT IS A NON-INTEGER NUMBER, OR AN INTEGER  BSI12100
1211                    *    WHOSE ABSOLUTE VALUE IS GREATER THAN 63          BSI12110
1212                    *                                                     BSI12120
1213 00730    0 10 00100 E$03   JSI   L$22       LOAD THE FIRST ARGUMENT      BSI12130
1214 00731    0 001021          DAC   ETMP                                    BSI12140
1215 00732    100040            SZE             TEST ZERO                    BSI12150
1216 00733    0 01 00740        JMP   *+5        IF NOT, JUMP OVER NEXT FEW LINES  BSI12160
1217 00734    0 11 01017        CAS   FTMP       IF ZERO, COMPARE WITH EXPONENT  BSI12170
1218 00735    0 10 00000        JSI   ERR        ZERO TO NEGATIVE POWER - FLAG DIVISION  BSI12180
1219 00736    142332            BCI   1,DZ       BY ZERO                      BSI12190
1220 00737   -0 01 00714        JMP*  E$22       ZERO TO POSITIVE POWER - EXIT WITH ZERO  BSI12200
1221                    *                                                     BSI12210
1222                    *    COMPUTE EXP(ARG2*LN(ARG1))                       BSI12220
1223                    *                                                     BSI12230
1224 00740    0 10 01023        JSI   LOGF       COMPUTE NATURAL LOG OF FIRST ARGUMENT  BSI12240
1225 00741    0 001021          DAC   ETMP                                    BSI12250
1226 00742    0 10 00530        JSI   M$22       MULTIPLY BY THE EXPONENT     BSI12260
1227 00743    0 001017          DAC   FTMP                                    BSI12270
1228 00744    0 10 00104        JSI   H$22       STORE IT                     BSI12280
1229 00745    0 001017          DAC   FTMP                                    BSI12290
1230 00746    0 10 01107        JSI   EXPF       COMPUTE EXP(B*LN(A))         BSI12300
1231 00747    0 001017          DAC   FTMP                                    BSI12310
1232 00750   -0 01 00714        JMP*  E$22       EXIT                         BSI12320
1233                    *                                                     BSI12330
1234                    *    TEST IF THE ABSOLUTE VALUE OF THE INTEGER IS LESS THAN 64  BSI12340
1235                    *                                                     BSI12350
1236 00751    100400      E$01  SPL             SKIP IF POSITIVE             BSI12360
1237 00752    140407            CA              COMPLIMENT IF NEGATIVE       BSI12370
1238 00753    0 11 00273        CAS   C100       TEST FOR LESS THAN 64        BSI12380
```

```
1239 00754   101000          NOP                                                    BSI12390
1240 00755   0 01 00730       JMP   E$03          JUMP IF NOT                      BSI12400
1241                  *                                                           BSI12410
1242                  *    HERE IF EXPONENT IS AN INTEGER WHOSE ABSOLUTE VALUE IS LESS THAN 64BSI12420
1243                  *                                                           BSI12430
1244 00756   101100    E$06 SLN           TEST LSB OF EXPONENT                    BSI12440
1245 00757   0 01 00770     JMP   E$04          JUMP IF ZERO                     BSI12450
1246 00760   0 13 01222     IMA   Z0+1          IF ONE, LOAD THE INTERMEDIATE RESULT BSI12460
1247 00761   000201         IAB                                                   BSI12470
1248 00762   0 02 01221     LDA   Z0                                              BSI12480
1249 00763   0 10 00530     JSI   M$22          MULTIPLY BY POWER OF FIRST ARGUMENT BSI12490
1250 00764   0 001021       DAC   ETMP                                           BSI12500
1251 00765   0 04 01221     STA   Z0            SAVE THE RESULT                  BSI12510
1252 00766   000201         IAB                                                   BSI12520
1253 00767   0 13 01222     IMA   Z0+1          AND RECOVER THE EXPONENT         BSI12530
1254 00770   0404 77   E$04 LGR   1            SHIFT EXPONENT                   BSI12540
1255 00771   101040         SNZ                 TEST IF THE EXPONENT EQUALS ZERO BSI12550
1256 00772   0 01 01004     JMP   E$05          IF ZERO, JUMP TO TEST SIGN OF EXPONENT BSI12560
1257 00773   0 04 00450     STA   HGHC          IF NON-ZERO, SAVE THE EXPONENT   BSI12570
1258 00774   0 10 00100     JSI   L$22          LOAD POWER OF FIRST ARGUMENT     BSI12580
1259 00775   0 001021       DAC   ETMP                                           BSI12590
1260 00776   0 10 00530     JSI   M$22          MULTIPLY BY ITSELF               BSI12600
1261 00777   0 001021       DAC   ETMP                                           BSI12610
1262 01000   0 10 00104     JSI   H$22          SAVE AS THE FIRST ARGUMENT       BSI12620
1263 01001   0 001021       DAC   ETMP                                           BSI12630
1264 01002   0 02 00450     LDA   HGHC          LOAD THE EXPONENT                BSI12640
1265 01003   0 01 00756     JMP   E$06          LOOP TO TEST LSB OF THE EXPONENT BSI12650
1266                  *                                                           BSI12660
1267                  *    TEST SIGN OF EXPONENT, INVERT RESULT IF SIGN IS NEGATIVE, AND BSI12670
1268                  *    RETURN                                                 BSI12680
1269                  *                                                           BSI12690
1270 01004   0 02 00650 E$05 LDA   CTMP          LOAD SIGNED EXPONENT             BSI12700
1271 01005   100400         SPL                 TEST THE SIGN                    BSI12710
1272 01006   0 01 01012     JMP   E$07          JUMP IF NEGATIVE                 BSI12720
1273 01007   0 10 00100     JSI   L$22          LOAD RESULT                      BSI12730
1274 01010   0 001221       DAC   Z0                                            BSI12740
1275 01011  -0 01 00714     JMP*  E$22          EXIT                             BSI12750
```

```
1276 01012   0 10 00100 E$07 JSI   L$22    NEGATIVE EXPONENT - LOAD ONE          BSI12760
1277 01013   0 001144       DAC   F1                                            BSI12770
1278 01014   0 10 00361     JSI   D$22    DIVIDE BY RESULT TO INVERT            BSI12780
1279 01015   0 001221       DAC   Z0                                            BSI12790
1280 01016  -0 01 00714     JMP*  E$22    EXIT                                  BSI12800
1281                  *                                                        BSI12810
1282                  *                                                        BSI12820
1283 01017   000000   FIMP BSZ   2                                             BSI12830
1284 01021   000000   EIMP BSZ   2                                             BSI12840
1285                  *                                                        BSI12850
1286                  *                                                        BSI12860
1287                  *                                                        BSI12870
1288                       FIN                                                 BSI12880
1289                       EJCT                                                BSI12890
```

```
1290                    *                                                                       BSI12900
1291                    *                                                                       BSI12910
1292                    *          NATURAL LOGARITHMIC FUNCTION                                 BSI12920
1293                    *                                                                       BSI12930
1294                    *     CALLING SEQUENCE:                                                 BSI12940
1295                    *                                                                       BSI12950
1296                    *          JST    LOGF                                                  BSI12960
1297                    *          DAC    ARG          POINTER TO ARGUMENT                      BSI12970
1298                    *          ......RETURN        LN(ARG) IN A AND B REGISTERS             BSI12980
1299                    *                                                                       BSI12990
1300                    *                                                                       BSI13000
1301                    *          THE ARGUMENT IS LOADED, AND IF IT IS LESS THAN OR EQUAL TO   BSI13010
1302                    *     ZERO, AN LG ERROR IS FLAGGED.  THE HIGH WORD OF THE ARGUMENT IS   BSI13020
1303                    *     SAVED, AND THE EXPONENT OF THE ARGUMENT IS SET TO ONE PLUS THE    BSI13030
1304                    *     BIAS.  ONE IS SUBTRACTED FROM THE ARGUMENT, AND A POLYNOMIAL IS   BSI13040
1305                    *     EVALUATED IN THE DIFFERENCE WITH THE FOLLOWING COEFFICIENTS: C0=0, BSI13050
1306                    *     C1=.9999964, C2=-.4998741, C3=.331799, C4=-.2407338, C5=.1676541, BSI13060
1307                    *     C6=-.09532939, C7=.03608849, C8=-.006453544.  THE EXPONENT IS     BSI13070
1308                    *     EXTRACTED FROM THE HIGH WORD OF THE ORIGINAL ARGUMENT, AND THE    BSI13080
1309                    *     BIAS PLUS ONE IS SUBTRACTED FROM IT.  IT IS CONVERTED TO A FLOATINGBSI13090
1310                    *     POINT NUMBER AND THEN DIVIDED BY THE LOGARITHM OF E TO THE BASE   BSI13100
1311                    *     TWO.  THE POLYNOMIAL RESULT IS ADDED TO THE QUOTIENT FOR THE      BSI13110
1312                    *     FUNCTION RESULT.                                                  BSI13120
1313                    *                                                                       BSI13130
1314                    *                                                                       BSI13140
1315 01023  0 000000  LOGF DAC  **          ENTRY                                               BSI13150
1316 01024  0 02 01023     LDA  LOGF                                                           BSI13160
1317 01025  0 10 00070     JST  LARG        LOAD THE ARGUMENT                                  BSI13170
1318 01026  0 11 01060     CAS  FO          COMPARE WITH ZERO                                  BSI13180
1319 01027  0 01 01033     JMP  *+4         IF GREATER SKIP NEXT FEW LINES                     BSI13190
1320 01030  101000         NOP                                                                 BSI13200
1321 01031  0 10 00000     JST  ERR         IF NOT, FLAG AN LG ERROR                           BSI13210
1322 01032  146307         BCI  1,LG                                                           BSI13220
1323 01033  0 04 01170     STA  Z2          SAVE THE HIGH WORD OF THE ARGUMENT                 BSI13230
1324 01034  0414 77        LGL  1           SHIFT EXPONENT OF ARGUMENT INTO FIRST 8 BIT        BSI13240
1325 01035  141050         CAL              CLEAR THE EXPONENT                                 BSI13250
1326 01036  0404 77        LGR  1           REFORMAT                                           BSI13260
```

```
1327 01037   0 05 01106      ERA   LMSK    SET THE EXPONENT TO ONE PLUS THE BIAS      BSI13270
1328 01040   0 10 00277      JSI   S$22    SUBTRACT ONE                               BSI13280
1329 01041   0 001144        DAC   F1                                                 BSI13290
1330 01042   0 10 01172      JSI   FPLY    EVALUATE THE POLYNOMIAL                    BSI13300
1331 01043   0 001102        DAC   LG01                                               BSI13310
1332 01044   0 10 00104      JSI   H$22    SAVE THE RESULT AS Z0                      BSI13320
1333 01045   0 001221        DAC   Z0                                                 BSI13330
1334 01046   0 02 01170      LDA   Z2      LOAD HIGH WORD OF ORIGINAL ARGUMENT        BSI13340
1335 01047   0 07 01106      SUB   LMSK    SUBTRACT BIAS PLUS ONE FROM THE EXPONENT   BSI13350
1336 01050   0405 71         ARS   7       RIGHT JUSTIFY THE EXPONENT                 BSI13360
1337 01051   0 10 00573      JSI   FINT    CONVERT IT TO A FLOATING POINT NUMBER      BSI13370
1338 01052   0 10 00361      JSI   D$22    DIVIDE IT BY LOG OF E TO                   BSI13380
1339 01053   0 001104        DAC   LG2E    THE BASE 2                                 BSI13390
1340 01054   0 10 00304      JSI   A$22    ADD Z0                                     BSI13400
1341 01055   0 001221        DAC   Z0                                                 BSI13410
1342 01056   0 12 01023      IRS   LOGF    SET-UP RETURN ADDRESS                      BSI13420
1343 01057  -0 01 01023      JMP*  LOGF    RETURN                                     BSI13430
1344                 *                                                              BSI13440
1345                 *                                                              BSI13450
1346 01060   000000    LOG0  DEC   0.0                                              BSI13460
     01061   000000
1347 01062   040177    LOG1  OCT   40177,177742                                     BSI13470
     01063   177742
1348 01064   140000    LOG2  OCT   140000,4100                                      BSI13480
     01065   004100
1349 01066   037724    LOG3  OCT   37724,170310                                     BSI13490
     01067   170310
1350 01070   140204    LOG4  OCT   140204,137212                                    BSI13500
     01071   137212
1351 01072   037525    LOG5  OCT   37525,153301                                     BSI13510
     01073   153301
1352 01074   140436    LOG6  OCT   140436,60771                                     BSI13520
     01075   060771
1353 01076   037111    LOG7  OCT   37111,164304                                     BSI13530
     01077   164304
1354 01100   141426    LOG8  OCT   141426,41740                                     BSI13540
     01101   041740
```

```
1355 01102   0 001100   LG01 DAC   LOG8                                    BSI13550
1356 01103   177770          DEC   -8                                      BSI13560
1357         001060     FO    EQU   LOG0      FLOATING POINT ZERO           BSI13570
1358 01104   040334     LG2E  OCT   40334,52436                            BSI13580
     01105   052436
1359 01106   040200     LMSK  OCT   40200                                   BSI13590
1360                          EJCT                                          BSI13600
```

```
1361                    *                                                                       BSI13610
1362                    *                                                                       BSI13620
1363                    *          EXPONENTIAL FUNCTION                                         BSI13630
1364                    *                                                                       BSI13640
1365                    *     CALLING SEQUENCE:                                                 BSI13650
1366                    *                                                                       BSI13660
1367                    *          JSI    EXPF                                                  BSI13670
1368                    *          DAC    ARG          POINTER TO ARGUMENT                      BSI13680
1369                    *          ......RETURN       EE(ARG) IN THE A AND B REGISTERS         BSI13690
1370                    *                                                                       BSI13700
1371                    *                                                                       BSI13710
1372                    *          THE ARGUMENT IS LOADED, SAVED, AND THEN DIVIDED BY THE       BSI13720
1373                    *     NATURAL LOGARITM OF TWO.  THE QUOTIENT IS CONVERTED TO AN INTEGER, BSI13730
1374                    *     ONE IS ADDED TO IT, AND THE RESULT IS SAVED AS Z2.  IF THE QUOTIENTBSI13740
1375                    *     IS OUT OF INTEGER RANGE, NUMERIC UNDERFLOW (NU) OR NUMERIC OVERFLOWBSI13750
1376                    *     (NO) IS FLAGGED THROUGH NORM , DEPENDING ON THE ARGUMENT'S SIGN. Z2BSI13760
1377                    *     IS THEN MULTIPLIED BY THE NATURAL LOGARITHM OF TWO, AND THE       BSI13770
1378                    *     ORIGINAL ARGUMENT IS SUBTRACTED FROM IT.  A POLYNOMIAL IS EVALUATEDBSI13780
1379                    *     IN THE DIFFERENCE WITH FOLLOWING COEFFICIENTS:C0=1.0, C1=-1.0,    BSI13790
1380                    *     C2=.4999999,C3=-.1666653, C4=.04165735, C5=-.00830136,            BSI13800
1381                    *     C6=.001329882, C7=-.0001413161.  Z2 IS ADDED TO THE EXPONENT OF THEBSI13810
1382                    *     POLYNOMIAL RESULT FOR THE FUNCTION RESULT.                        BSI13820
1383                    *                                                                       BSI13830
1384                    *                                                                       BSI13840
1385 01107   0 000000   EXPF DAC    **           ENTRY                                          BSI13850
1386 01110   0 02 01107      LDA    EXPF                                                        BSI13860
1387 01111   0 10 00070      JSI    LARG         LOAD THE ARGUMENT                              BSI13870
1388 01112   0 10 00104      JSI    H$Z2         SAVE IT AS Z0                                  BSI13880
1389 01113   0 001221        DAC    Z0                                                          BSI13890
1390 01114   0 10 00361      JSI    D$Z2         DIVIDE THE ARGUMENT BY THE                     BSI13900
1391 01115   0 001166        DAC    LN2          NATURAL LOGARITHM OF TWO                       BSI13910
1392 01116   0 10 00604      JSI    IFLI         CONVERT IT TO AN INTEGER                       BSI13920
1393 01117   0 01 01140      JMP    EX01         OUT OF INTEGER RANGE                           BSI13930
1394 01120   141206          AOA                 ADD ONE                                        BSI13940
1395 01121   0 04 01170      STA    Z2           AND SAVE                                       BSI13950
1396 01122   0 10 00573      JSI    FINI         FLOAT IT                                       BSI13960
1397 01123   0 10 00530      JSI    M$Z2         MULTIPLY BY THE NATURAL LOG OF TWO             BSI13970
```

```
1398 01124   0 001166        DAC   LN2                                                  BSI13980
1399 01125   0 10 00277       JSI   S$22    SUBTRACT THE ORIGINAL ARGUMENT             BSI13990
1400 01126   0 001221        DAC   Z0                                                   BSI14000
1401 01127   0 10 01172       JSI   EPLY    EVALUATE POLYNOMIAL                        BSI14010
1402 01130   0 001164        DAC   EX02                                                 BSI14020
1403 01131   0 10 00134       JSI   UNPK    UNPACK THE RESULT                          BSI14030
1404 01132   0 13 00152       IMA   EXPT    LOAD THE EXPONENT                          BSI14040
1405 01133   0 06 01170       ADD   Z2      ADD Z2                                     BSI14050
1406 01134   0 13 00152       IMA   EXPT    RESTORE THE FORMAT                         BSI14060
1407 01135   0 10 00202       JSI   NORM    REPACK                                     BSI14070
1408 01136   0 12 01107       IRS   EXPF    SET-UP RETURN ADDRESS                      BSI14080
1409 01137  -0 01 01107       JMP*  EXPF    RETURN                                     BSI14090
1410 01140   0 02 01221 EX01  LDA   Z0      LOAD THE HIGH WORD OF THE ARGUMENT         BSI14100
1411 01141   101400          SMI           TEST ITS SIGN                              BSI14110
1412 01142   0 01 00270       JMP   N7      POSITIVE - JUMP TO FLAG NUMERIC OVERFLOW   BSI14120
1413 01143   0 01 00266       JMP   N6      NEGATIVE - JUMP TO FLAG NUMERIC UNDERFLOW  BSI14130
1414                  *                                                               BSI14140
1415                  *                                                               BSI14150
1416 01144   040300      EXP0 DEC   1.0                                               BSI14160
     01145   000000
1417 01146   137600      EXP1 OCT   137600,0                                          BSI14170
     01147   000000
1418 01150   037777      EXP2 OCT   37777,177777                                      BSI14180
     01151   177777
1419 01152   140252      EXP3 OCT   140252,125330                                     BSI14190
     01153   125330
1420 01154   037125      EXP4 OCT   37125,50163                                       BSI14200
     01155   050163
1421 01156   141273      EXP5 OCT   141273,177311                                     BSI14210
     01157   177311
1422 01160   035727      EXP6 OCT   35727,23670                                      BSI14220
     01161   023670
1423 01162   142665      EXP7 OCT   142665,164340                                     BSI14230
     01163   164340
1424 01164   0 001162    EX02 DAC   EXP7                                             BSI14240
1425 01165   177777      M7   DEC   -1                                               BSI14250
1426         001144      F1   EQU   EXP0    FLOATING POINT ONE                       BSI14260
```

```
1427 01166  040130      LN2  OCT   40130,134414                          BSI14270
     01167  134414
1428 01170  000000      ZZ   BSZ   Z                                     BSI14280
1429                          EJCT                                       BSI14290
```

```
1430  *                                                                      BSI14300
1431  *                                                                      BSI14310
1432  *          POLYNOMIAL EVALUATION ROUTINE                               BSI14320
1433  *                                                                      BSI14330
1434  *     CALLING SEQUENCE:                                                BSI14340
1435  *                                                                      BSI14350
1436  *          JST    EPLY        ARGUMENT IN THE A AND B REGISTERS        BSI14360
1437  *          DAC    CPLC        ADDRESS OF COEFFICIENT POINTER AND TWO'S  BSI14370
1438  *                             COMPLIMENT OF THE DEGREE OF THE POLYNOMIAL BSI14380
1439  *          ......RETURN       RESULT RETURNED IN THE A AND B REGISTERS  BSI14390
1440  *          .                                                           BSI14400
1441  *          .                                                           BSI14410
1442  *          .                                                           BSI14420
1443  *          DEC    C0                                                   BSI14430
1444  *          DEC    C1                                                   BSI14440
1445  *          .                                                           BSI14450
1446  *          .                                                           BSI14460
1447  *          .                                                           BSI14470
1448  *     CNAD DEC    CN                                                   BSI14480
1449  *          .                                                           BSI14490
1450  *          .                                                           BSI14500
1451  *          .                                                           BSI14510
1452  *     CPLC DAC    CNAD                                                 BSI14520
1453  *          DEC    -N                                                   BSI14530
1454  *                                                                      BSI14540
1455  *                                                                      BSI14550
1456  *          THIS ROUTINE EVALUATES THE POLYNOMIAL OF THE FORM:           BSI14560
1457  *     C0+C1*X+C2*X**2+C3*X**3+...+CN*X**N.  THE ARGUMENT PASSED IN THE A ANDBSI14570
1458  *     B REGISTERS IS SAVED, AND THE COEFFICIENT POINTER AND THE TWO'S    BSI14580
1459  *     COMPLIMENT OF THE DEGREE OF THE POLYNOMIAL ARE LOADED.  THE COEF-  BSI14590
1460  *     FICIENT POINTER IS SAVED IN THE ADDITION CALL IN THE LOOP.  THE    BSI14600
1461  *     TWO'S COMPLIMENT OF THE DEGREE OF THE POLYNOMIAL IS SAVED AS A LOOPBSI14610
1462  *     COUNTER.  THE NTH COEFFICIENT IS LOADED, AND THE COEFFICIENT       BSI14620
1463  *     POINTER IS DECREMENTED TO POINT AT THE NEXT COEFFICIENT.  THE      BSI14630
1464  *     CURRENT RESULT IS MULTIPLIED BY THE ARGUMENT, AND THE NEXT COEF-   BSI14640
1465  *     FICIENT IS ADDED.  THE LOOP COUNTER IS INCREMENTED BY ONE, AND IF  BSI14650
1466  *     IT IS EQUAL TO ZERO RETURN IS MADE.  OTHERWISE THE ROUTINE LOOPS   BSI14660
```

```
1467                        *     TO DECREMENT THE COEFFICIENT POINTER.                                  BSI14670
1468                        *                                                                           BSI14680
1469                        *                                                                           BSI14690
1470 01172   0 000000  FPLY DAC   **          ENTRY                                                     BSI14700
1471 01173   0 10 00104       JSI   H$22      SAVE THE ARGUMENT                                         BSI14710
1472 01174   0 001221        DAC   Z0                                                                  BSI14720
1473 01175   0 02 01172      LDA   FPLY       LOAD THE COEFFICIENT POINTER                              BSI14730
1474 01176   0 10 00070      JSI   LARG       AND THE LOOP COUNTER                                      BSI14740
1475 01177   0 04 01213      STA   FP01       SAVE COEDDICIENT POINTER IN ADDITION CALL                  BSI14750
1476 01200   0 04 01204      STA   FP03       SAVE TO LOAD THE NTH COEFFICIENT                          BSI14760
1477 01201   000201          IAB              LOAD LOOP COUNTER                                         BSI14770
1478 01202   0 04 01220      STA   FPCR       AND SAVE                                                  BSI14780
1479 01203   0 10 00100      JSI   L$22       LOAD THE NTH COEFFICIENT                                  BSI14790
1480 01204   0 000000  FP03 DAC   **                                                                   BSI14800
1481 01205   0 13 01213 FP02 IMA   FP01       DECREMENT THE COEFFICIENT POINTER IN THE                  BSI14810
1482 01206   0 07 00506      SUB   C2         ADDITION CALL SO THAT IT POINTS TO THE                    BSI14820
1483 01207   0 13 01213      IMA   FP01       NEXT COEFFICIENT                                          BSI14830
1484 01210   0 10 00530      JSI   M$22       MULTIPLY BY THE ARGJMENT                                  BSI14840
1485 01211   0 001221        DAC   Z0         X                                                         BSI14850
1486 01212   0 10 00304      JSI   A$22       ADD THE COEFFICIENT                                       BSI14860
1487 01213   0 000000  FP01 DAC   **         X                                                         BSI14870
1488 01214   0 12 01220      IRS   FPCR       INCREMENT THE COUNTER                                     BSI14880
1489 01215   0 01 01205      JMP   FP02       LOOP UNTIL COUNTER EQUALS ZERO                            BSI14890
1490 01216   0 12 01172      IRS   FPLY       SET-UP RETURN ADDRESS                                     BSI14900
1491 01217  -0 01 01172      JMP*  FPLY       RETURN                                                    BSI14910
1492                        *                                                                           BSI14920
1493                        *                                                                           BSI14930
1494 01220   000000    FPCR BSZ   1                                                                     BSI14940
1495 01221   000000    Z0   BSZ   2                                                                     BSI14950
1496                        EJCT                                                                        BSI14960
```

```
1497                    *                                                                       BSI14970
1498                    *                                                                       BSI14980
1499                    *          SQUARE ROOT FUNCTION                                         BSI14990
1500                    *                                                                       BSI15000
1501                    *    CALLING SEQUENCE:                                                  BSI15010
1502                    *                                                                       BSI15020
1503                    *          JSI    SQRF                                                  BSI15030
1504                    *          DAC    ARG          POINTER TO THE ARGUMENT                  BSI15040
1505                    *          ......RETURN        SQUARE ROOT OF THE ARGUMENT RETURNED IN A BSI15050
1506                    *                              AND B REGISTERS                          BSI15060
1507                    *                                                                       BSI15070
1508                    *                                                                       BSI15080
1509                    *          THE ARGUMENT IS LOADED AND THE FLOATING POINT IS UNPACKED.   BSI15090
1510                    *    A STRAIGHT LINE APPROXIMATION IS MADE TO THE SQUARE ROOT OF THE    BSI15100
1511                    *    ARGUMENT BY DIVIDING THE EXPONENT BY TWO AND REPLACING THE MANTISSABSI15110
1512                    *    WITH 5/8 TIMES THE MANTISSA PLUS 3/8.  THE FIRST GUESS IS PUT INTO BSI15120
1513                    *    FLOATING POINT FORMAT AND FOUR NEWTON-RAPHSON ITERATIONS ARE MADE  BSI15130
1514                    *    TO OBTAIN FULL FLOATING POINT ACCURACY                             BSI15140
1515                    *                                                                       BSI15150
1516                    *                                                                       BSI15160
1517 01223    0 000000  SQRF  DAC    **           SQUARE ROOT FUNCTION ENTRY                   BSI15170
1518 01224    0 02 01272      LDA    M4           INITIALIZE N-R COUNTER TO -4                 BSI15180
1519 01225    0 04 00650      STA    CTMP         COUNTER                                      BSI15190
1520 01226    0 02 01223      LDA    SQRF                                                      BSI15200
1521 01227    0 10 00070      JSI    LARG         LOAD ARGUMENT ROUTINE                        BSI15210
1522 01230    0 11 01060      CAS    FO           TEST FOR ZERO AND NEGATIVE ARGUMENT          BSI15220
1523 01231    0 01 01235      JMP    *+4          ARGUMENT IS POSITIVE NUMBER, CONTINUE        BSI15230
1524 01232   -0 01 00120      JMP*   EXIT+1       ARGUMENT IS ZERO-RETURN WITH ZERO            BSI15240
1525 01233    0 10 00000      JSI    ERR          NEGATIVE NUMBER, FLAG ERROR                  BSI15250
1526 01234    151721          BCI    1,SQ                                                      BSI15260
1527                    *                                                                       BSI15270
1528                    *    SET EXPONENT OF FIRST GUESS EQUAL TO HALF THE ARGUMENT EXPONENT    BSI15280
1529                    *                                                                       BSI15290
1530 01235    0 10 00134      JSI    UNPK         UNPACK THE FLOATING POINT                    BSI15300
1531 01236    0 13 00152      IMA    EXPT         LOAD THE EXPONENT                            BSI15310
1532 01237    0404 77         LGR    1            DIVIDE IT BY TWO                             BSI15320
1533 01240    0 06 00273      ADD    C100         ADD BIAS/2                                   BSI15330
```

```
1534 01241   0 13 00152        IMA   EXPT                SAVE EXPONENT, RECOVER HIGH MANTISSA      BSI15340
1535 01242   101001           SSC                       SKIP IF EXPONENT IS ODD                   BSI15350
1536 01243   0 01 01246       JMP   *+3                 JUMP IF EXPONENT EVEN                     BSI15360
1537 01244   0 12 00152       IRS   EXPT                INCREMENT THE EXPONENT                    BSI15370
1538 01245   0401 11          LRS   1                   AND SHIFT THE MANTISSA                    BSI15380
1539                  *                                                                           BSI15390
1540                  *    SET MANTISSA OF FIRST GUESS EQUAL TO 5/8 TIMES ARGUMENT MANTISSA       BSI15400
1541                  *    PLUS 3/8                                                               BSI15410
1542                  *                                                                           BSI15420
1543 01246   0401 11          LRS   1                   1/2 * MANTISSA                            BSI15430
1544 01247   0 10 00104       JSI   H$22                STORE                                     BSI15440
1545 01250   0 000200         DAC   HIGH                                                          BSI15450
1546 01251   0401 16          LRS   2                   1/8 * MANTISSA                            BSI15460
1547 01252   0 10 00155       JSI   DADD                1/2 + 1/8 = 5/8 * MANTISSA                BSI15470
1548 01253   0 06 01273       ADD   C30K                ADD 3/8                                   BSI15480
1549 01254   0 10 00202       JSI   NORM                REPACK THE FLOATING POINT                 BSI15490
1550 01255   0 10 00104 SQ01  JSI   H$22                SAVE AS XOLD                              BSI15500
1551 01256   0 001017         DAC   FTMP                                                          BSI15510
1552                  *                                                                           BSI15520
1553                  *    NEWTON-RAPHSON METHOD: XNEW=1/2*(ARG/XOLD+XOLD)                        BSI15530
1554                  *                                                                           BSI15540
1555 01257   0 02 01263       LDA   SQRF                LOAD THE ARGUMENT                         BSI15550
1556 01260   0 10 00070       JSI   LARG                                                          BSI15560
1557 01261   0 10 00361       JSI   D$22                ARG/XOLD                                  BSI15570
1558 01262   0 001017         DAC   FTMP                                                          BSI15580
1559 01263   0 10 00304       JSI   A$22                ARG/XOLD + XOLD                           BSI15590
1560 01264   0 001017         DAC   FTMP                                                          BSI15600
1561 01265   0 07 00572       SUB   C200               1/2*(ARG/XOLD+XOLD)                       BSI15610
1562 01266   0 12 00650       IRS   CTMP                INCREMENT THE ITERATION COUNTER           BSI15620
1563 01267   0 01 01255       JMP   SQ01                LOOP IF COUNTER IS NON-ZERO               BSI15630
1564 01270   0 12 01263       IRS   SQRF                INCREMENT FOR RETURN                      BSI15640
1565 01271  -0 01 01263       JMP*  SQRF                RETURN IF COUNTER EQUALS ZERO             BSI15650
1566                  *                                                                           BSI15660
1567                  *                                                                           BSI15670
1568 01272   177774     M4    OCT   -4                                                            BSI15680
1569 01273   030000     C30K  OCT   30000                                                         BSI15690
1570                         FIN                                                                 BSI15700
```

```
1571                    EJCT                                    BSI15710
```

```
1572                    *                                                                       BSI15720
1573                    *                                                                       BSI15730
1574                    *          COSINE FUNCTION ROUTINE                                      BSI15740
1575                    *                                                                       BSI15750
1576                    *    CALLING SEQUENCE:                                                  BSI15760
1577                    *                                                                       BSI15770
1578                    *          JSI   COSF                                                   BSI15780
1579                    *          DAC   ARG          POINTER TO THE ARGUMENT                   BSI15790
1580                    *          ......RETURN       COSINE OF THE ARGUMENT IN A AND B REGISTERSBSI15800
1581                    *                                                                       BSI15810
1582                    *                                                                       BSI15820
1583                    *          THE ROUTINE MOVES THE RETURN ADDRESS TO THE SINE ROUTINE.    BSI15830
1584                    *    THE ARGUMENT IS LOADED, AND ONE-HALF PI IS ADDED TO IT.  THE ROUTINBSI15840
1585                    *    THEN JUMPS INTO THE SINE FUNCTION ROUTINE.  NOTE THE IDENTITY:     BSI15850
1586                    *          COSINE(ARG)=SINE(ARG+PI/2)                                   BSI15860
1587                    *                                                                       BSI15870
1588                    *                                                                       BSI15880
1589 01274   0 000000   COSF DAC   **             ENTRY                                        BSI15890
1590 01275   0 02 01274      LDA   COSF           MOVE RETURN ADDRESS TO SINF ENTRY            BSI15900
1591 01276   0 04 01303      STA   SINF                                                        BSI15910
1592 01277   0 10 00070      JSI   LARG           LOAD THE ARGUMENT                            BSI15920
1593 01300   0 10 00304      JSI   A$22          ADD PI/2                                     BSI15930
1594 01301   0 001364        DAC   HFPI                                                       BSI15940
1595 01302   0 01 01306      JMP   *+4            JUMP INTO SINE FUNCTION                      BSI15950
1596                         EJCT                                                            BSI15960
```

```
1597                    *                                                                       BSI15970
1598                    *                                                                       BSI15980
1599                    *          SINE FUNCTION ROUTINE                                        BSI15990
1600                    *                                                                       BSI16000
1601                    *    CALLING SEQUENCE:                                                  BSI16010
1602                    *                                                                       BSI16020
1603                    *          JSI   SINF                                                   BSI16030
1604                    *          DAC   ARG          POINTER TO THE ARGUMENT                   BSI16040
1605                    *          ......RETURN       SINE OF THE ARGUMENT IN THE A AND B REGIS- BSI16050
1606                    *                             TERS                                      BSI16060
1607                    *                                                                       BSI16070
1608                    *                                                                       BSI16080
1609                    *          THE ARGUMENT IS LOADED AND DIVIDED BY 2*PI TO CONVERT RADIANSBSI16090
1610                    *    TO CIRCLE REVOLUTIONS.  THE INTEGRAL PART OF THE RESULT IS DIS-    BSI16100
1611                    *    CARDED SINCE SIN(X)=SIN(X+2*PI).  THE FRACTIONAL PART OF THE RESULTBSI16110
1612                    *    IS REDUCED TO AN ANGLE IN THE FIRST AND FOURTH QUADRANT            BSI16120
1613                    *    (-1/4<=F.P.<=1/4), USING THE IDENTITIES: SIN(X)=SINC(X-2*PI) AND   BSI16130
1614                    *    SIN(X)=SIN(PI-X).  THE REDUCED FRACTIONAL PART IS SAVED AND THEN   BSI16140
1615                    *    SQUARED.  THE POLYNOMIAL IS EVALUATED IN THE REDUCED FRACTIONAL    BSI16150
1616                    *    PART SQUARED WITH THE FOLLOWING COEFFICIENTS  C0=6.283185,         BSI16160
1617                    *    C1=41.34168, C2=81.60223, C3=-76.57498, C4=39.701067.  THE POLY-   BSI16170
1618                    *    NOMIAL RESULT IS MULTIPLIED BY THE REDUCED FRACTIONAL PART TO OB-  BSI16180
1619                    *    TAIN THE FUNCTION RESULT.                                          BSI16190
1620                    *                                                                       BSI16200
1621                    *                                                                       BSI16210
1622 01303   0 000000   SINF DAC   **                 ENTRY                                     BSI16220
1623 01304   0 02 01303      LDA   SINF                                                         BSI16230
1624 01305   0 10 00070      JSI   LARG               LOAD THE ARGUMENT                         BSI16240
1625 01306   0 10 00361      JSI   D$22               DIVIDE IT BY 2*PI                         BSI16250
1626 01307   0 001350        DAC   TWPI                                                        BSI16260
1627 01310   0 10 00104      JSI   H$22               SAVE THE RESULT                           BSI16270
1628 01311   0 001221        DAC   Z0                                                          BSI16280
1629 01312   0 10 00651      JSI   INTF               TAKE THE GREATEST INTEGER                 BSI16290
1630 01313   0 001221        DAC   Z0                 FUNCTION                                  BSI16300
1631 01314   0 10 00277      JSI   S$22               SUBTRACT TO FIND NEGATIVE                 BSI16310
1632 01315   0 001221        DAC   Z0                 FRACTIONAL PART                           BSI16320
1633 01316   0 10 00142      JSI   N$22               TWO'S COMPLIMENT IT                       BSI16330
```

```
1634 01317   0 10 00104       JSI   H$22          SAVE FRACTIONAL PART AS Z2                    BSI16340
1635 01320   0 001170          DAC   Z2                                                          BSI16350
1636 01321   0 10 00277        JSI   S$22          SUBTRACT 1/4 TO TEST WHETHER                  BSI16360
1637 01322   0 001366          DAC   FFRH          ANGLE IS IN FIRST QUADRANT                    BSI16370
1638 01323   100400            SPL                                                             BSI16380
1639 01324   0 01 01345        JMP   SN02          JUMP TO RELOAD THE FRACTIONAL PART            BSI16390
1640 01325   0 10 00277        JSI   S$22          SUBTRACT 1/2 FROM RESULT TO                   BSI16400
1641 01326   0 001370          DAC   FHLF          TEST WHETHER IT'S IN FOURTH QUADRANT          BSI16410
1642 01327   100400            SPL                 IF IT IS, SKIP                              BSI16420
1643 01330   0 10 00122        JSI   N$22          IF NOT, COMPLIMENT                            BSI16430
1644 01331   0 10 00277        JSI   S$22          SUBTRACT 1/4                                  BSI16440
1645 01332   0 001366          DAC   FFRH                                                        BSI16450
1646 01333   0 10 00104        JSI   H$22          SAVE THE RESULT AS THE REDUCED ARGUMENT       BSI16460
1647 01334   0 001170          DAC   Z2                                                          BSI16470
1648 01335   0 10 00530  SN01  JSI   M$22          SQUARE THE REDUCED ARGUMENT                   BSI16480
1649 01336   0 001170          DAC   Z2                                                          BSI16490
1650 01337   0 10 01172        JSI   FPLY          EVALUATE POLYNOMIAL IN                        BSI16500
1651 01340   0 001362          DAC   SN03          THE REDUCED ARGUMENT SQUARED                  BSI16510
1652 01341   0 10 00530        JSI   M$22          MULTIPLY REDUCED ARGUMENT                     BSI16520
1653 01342   0 001170          DAC   Z2                                                          BSI16530
1654 01343   0 12 01303        IRS   SINF          SET-UP RETURN ADDRESS                         BSI16540
1655 01344  -0 01 01303        JMP*  SINF          RETURN                                        BSI16550
1656 01345   0 10 00100  SN02  JSI   L$22          RELOAD THE FRACTIONAL PART OF THE ARGUMENT     BSI16560
1657 01346   0 001170          DAC   Z2                                                          BSI16570
1658 01347   0 01 01335        JMP   SN01          JUMP TO THE POLYNOMIAL EVALUATION             BSI16580
1659                     *                                                                      BSI16590
1660                     *                                                                      BSI16600
1661 01350   040744      SIN0  OCT   40744,103755                                              BSI16610
     01351   103755
1662 01352   136255      SIN1  OCT   136255,50420                                              BSI16620
     01353   050420
1663 01354   041721      SIN2  OCT   41721,115054                                              BSI16630
     01355   115054
1664 01356   136063      SIN3  OCT   136063,66316                                              BSI16640
     01357   066316
1665 01360   041517      SIN4  OCT   41517,65735                                               BSI16650
     01361   065735
```

```
1666 01362   0 001360   SN03 DAC   SIN4                                          BSI16660
1667 01363   177774          DEC   -4                                           BSI16670
1668 01364   040344     HPPI OCT   40344,103755                                 BSI16680
     01365   103755
1669         001350     TWPI EQU   SIN0        PI MULTIPLIED BY TWO             BSI16690
1670 01366   037700     FFRH DEC   0.25                                         BSI16700
     01367   000000
1671 01370   040100     FHLF DEC   0.5                                          BSI16710
     01371   000000
1672                          EJCT                                              BSI16720
```

```
1673                    *                                                                       BSI16730
1674                    *                                                                       BSI16740
1675                    *          TANGENT FUNCTION ROUTINE                                     BSI16750
1676                    *                                                                       BSI16760
1677                    *    CALLING SEQUENCE:                                                  BSI16770
1678                    *                                                                       BSI16780
1679                    *          JSI   TANF                                                   BSI16790
1680                    *          DAC   ARG          POINTER TO THE ARGUMENT                   BSI16800
1681                    *          ......RETURN       TANGENT OF THE ARGUMENT IN THE A AND B    BSI16810
1682                    *                             REGISTERS                                 BSI16820
1683                    *                                                                       BSI16830
1684                    *                                                                       BSI16840
1685                    *          THE ARGUMENT IS LOADED AND SAVED.  THE COSINE OF THE ARGUMENTBSI16850
1686                    *    IS TAKEN AND SAVED.  THE SINE OF THE ARGUMENT IS TAKEN AND DIVIDED BSI16860
1687                    *    BY THE COSINE TO OBTAIN THE TANGENT.                               BSI16870
1688                    *                                                                       BSI16880
1689                    *                                                                       BSI16890
1690 01372   0 000000   TANF DAC   **                 ENTRY                                     BSI16900
1691 01373   0 02 01372      LDA   TANF                                                         BSI16910
1692 01374   0 10 00070      JSI   LARG               LOAD THE ARGUMENT                         BSI16920
1693 01375   0 10 00104      JSI   H$22               STORE IT                                  BSI16930
1694 01376   0 001021        DAC   ETMP                                                        BSI16940
1695 01377   0 10 01274      JSI   COSF               TAKE COSINE OF ARGUMENT                   BSI16950
1696 01400   0 001021        DAC   ETMP                                                        BSI16960
1697 01401   0 10 00104      JSI   H$22               SAVE COSINE                               BSI16970
1698 01402   0 001017        DAC   FTMP                                                        BSI16980
1699 01403   0 10 01303      JSI   SINF               TAKE SINE OF THE ARGUMENT                 BSI16990
1700 01404   0 001021        DAC   ETMP                                                        BSI17000
1701 01405   0 10 00361      JSI   D$22               DIVIDE SINE BY COSINE FOR TANGENT         BSI17010
1702 01406   0 001017        DAC   FTMP                                                        BSI17020
1703 01407   0 12 01372      IRS   TANF               INCREMENT FOR RETURN                      BSI17030
1704 01410  -0 01 01372      JMP*  TANF               RETURN                                    BSI17040
1705                         FIN                                                             BSI17050
1706                         EJCT                                                            BSI17060
```

```
1707                    *                                                                       BSI17070
1708                    *                                                                       BSI17080
1709                    *          ARCTANGENT FUNCTION ROUTINE                                  BSI17090
1710                    *                                                                       BSI17100
1711                    *     CALLING SEQUENCE:                                                 BSI17110
1712                    *                                                                       BSI17120
1713                    *          JST    ATNF                                                  BSI17130
1714                    *          DAC    ARG          POINTER TO THE ARGUMENT                  BSI17140
1715                    *          ......RETURN       ARCTANGENT OF ARGUMENT IN A AND B REGISTERSBSI17150
1716                    *                                                                       BSI17160
1717                    *                                                                       BSI17170
1718                    *          A SIGN FLAG AND A SCALE FACTOR FLAG ARE INITIALZED TO MINUS  BSI17180
1719                    *     TWO.  THE ARGUMENT IS LOADED, AND IF IT IS POSITIVE, THE SIGN FLAG BSI17190
1720                    *     IS INCREMENTED BY ONE.  IF THE ARGUMENT IS NEGATIVE, THE ARGUMENT  BSI17200
1721                    *     IS TWO'S COMPLIMENTED.  IF THE ABSOLUTE VALUE OF THE ARGUMENT      BSI17210
1722                    *     IS GREATER THAN OR EQUAL TO FLOATING POINT ONE, THE IDENTITY,      BSI17220
1723                    *     ATN(X)=PI/2+ATN(-1/X), IS APPLIED.  THE REDUCED ARGUMENT IS SAVED  BSI17230
1724                    *     AND THEN SQUARED.  THE POLYNOMIAL IS EVALUATED IN THE REDUCED ARGU-BSI17240
1725                    *     MENT SQUARED WITH THE FOLLOWING COEFFICIENTS: C0=1, C1=-.3333315,  BSI17250
1726                    *     C2=.1999355, C3=-.142089, C4=.1065626, C5=-.07528964,              BSI17260
1727                    *     C6=.04296961, C7=-.016165/4, C8=.002866226.  THE POLYNOMIAL RESULT BSI17270
1728                    *     IS MULTIPLIED BY THE REDUCED ARGUMENT.  IF THE ABSOLUTE VALUE OF   BSI17280
1729                    *     THE ORIGINAL ARGUMENT WAS GREATER THAN OR EQUAL TO ZERO, THE SCALE BSI17290
1730                    *     FACTOR FLAG WILL BE MINUS ONE.  WHEN IT IS 'IRS'ED A SKIP WILL     BSI17300
1731                    *     OCCUR CAUSING PI/2 TO BE ADDED TO THE RESULT.  THE SIGN FLAG IS    BSI17310
1732                    *     THEN 'IRS'ED, AND IF THE ORIGINAL ARGUMENT WAS POSITIVE, THERE WILLBSI17320
1733                    *     A SKIP.  OTHERWISE THE RESULT IS TWO'S COMPLIMENTED.               BSI17330
1734                    *                                                                       BSI17340
1735                    *                                                                       BSI17350
1736 01411   0 000000   ATNF DAC    **           ENTRY                                          BSI17360
1737 01412   0 02 01503      LDA    M2                                                          BSI17370
1738 01413   0 04 00650      STA    CIMP         INITIALIZE SIGN FLAG AND                       BSI17380
1739 01414   0 04 01502      STA    SFAC         SCALE FACTOR FLAG TO MINUS TWO                 BSI17390
1740 01415   0 02 01411      LDA    ATNF                                                        BSI17400
1741 01416   0 10 00070      JST    LARG         LOAD THE ARGUMENT                              BSI17410
1742 01417   101400          SMI                                                               BSI17420
1743 01420   0 12 00650      IRS    CIMP         IF POSITIVE, INCREMENT THE SIGN FLAG           BSI17430
```

```
1744 01421   100400          SPL                                                    BSI17440
1745 01422   0 10 00122       JSI   N$ZZ      IF NEGATIVE, TWO'S COMPLIMENT        BSI17450
1746 01423   0 11 01144       CAS   F1        COMPARE WITH ONE                     BSI17460
1747 01424   101000          NOP                                                    BSI17470
1748 01425   0 01 01446       JMP   A101      IF GREATER OR EQUAL, JUMP TO REDUCE   BSI17480
1749                  *                       THE ARGUMENT                         BSI17490
1750 01426   0 10 00104 A102  JSI   H$ZZ      SAVE THE REDUCED ARGUMENT AS Z2      BSI17500
1751 01427   0 001170        DAC   ZZ                                               BSI17510
1752 01430   0 10 00530       JSI   M$ZZ      SQUARE IT                            BSI17520
1753 01431   0 001170        DAC   ZZ                                               BSI17530
1754 01432   0 10 01172       JSI   EPLY      EVALUATE THE POLYNOMIAL IN THE REDICED BSI17540
1755 01433   0 001500        DAC   A103      ARGUMENT SQUARED                     BSI17550
1756 01434   0 10 00530       JSI   M$ZZ      MULTIPLY RESULT BY THE REDUCED ARGUMENT BSI17560
1757 01435   0 001170        DAC   ZZ                                               BSI17570
1758 01436   0 12 01502       IRS   SFAC      INCREMENT THE SCALE FACTOR FLAG      BSI17580
1759 01437   0 01 01442       JMP   *+3       NO SKIP-DON'T ADD THE SCALE FACTOR   BSI17590
1760 01440   0 10 00304       JSI   A$ZZ      ADD SCALE FACTOR OF PI/2             BSI17600
1761 01441   0 001364        DAC   HPP1                                            BSI17610
1762 01442   0 12 00650       IRS   CIMP      INCREMENT THE SIGN FLAG              BSI17620
1763 01443   0 10 00122       JSI   N$ZZ      NO SKIP-TWO'S COMPLIMENT             BSI17630
1764 01444   0 12 01411       IRS   AINF      SET-UP RETURN ADDRESS                BSI17640
1765 01445  -0 01 01411       JMP*  AINF      RETURN                               BSI17650
1766 01446   0 10 00104 A101  JSI   H$ZZ      SAVE THE ARGUMENT AS ZZ              BSI17660
1767 01447   0 001170        DAC   ZZ                                               BSI17670
1768 01450   0 10 00100       JSI   L$ZZ      LOAD FLOATING POINT MINUS ONE        BSI17680
1769 01451   0 000000        DAC   FM1                                              BSI17690
1770 01452   0 10 00361       JSI   D$ZZ      -1/ARGUMENT                          BSI17700
1771 01453   0 001170        DAC   ZZ                                               BSI17710
1772 01454   0 12 01502       IRS   SFAC      SET SCALE FACTOR FLAG SO THAT SCALE FACTOR BSI17720
1773 01455   0 01 01426       JMP   A102                                           BSI17730
1774                  *                                                            BSI17740
1775                  *                                                            BSI17750
1776 01456   040300    AIN0  DEC   1.0                                            BSI17760
     01457   000000
1777 01460   140052    AIN1  OCT   140052,125312                                   BSI17770
     01461   125312
1778 01462   037546    AIN2  OCT   37546,26762                                     BSI17780
```

```
      01463   056762
1779  01464   140267      AIN3  OCT   140267,40034                    BSI17790
      01465   040034
1780  01466   037355      AIN4  OCT   37355,17302                     BSI17800
      01467   017302
1781  01470   140462      AIN5  OCT   140462,163506                   BSI17810
      01471   163506
1782  01472   037127      AIN6  OCT   37127,160377                    BSI17820
      01473   160377
1783  01474   141075      AIN7  OCT   141075,144377                   BSI17830
      01475   144377
1784  01476   036135      AIN8  OCT   36135,165645                    BSI17840
      01477   165645
1785  01500   0 001476    AIO3  DAC   AIN8                            BSI17850
1786  01501   177770            DEC   -8                              BSI17860
1787  01502   000000      SPAC  BSZ   1                               BSI17870
1788  01503   177776      MZ    OCT   -2                              BSI17880
1789          001503      PP12  EQU   *-1                            BSI17890
1790                            EJCT                                  BSI17900
```

```
1791                    *                                                       BS117910
1792                    *                                                       BS117920
1793                         END                                                BS117930
```

A$22	000304	A1	000342	A3	000351	A6	000334
ABSF	000014	AD1	000052	AD2	000065	AD3	000060
ADDR	000046	ARS	000625	ARSM	000626	AT01	001446
AT02	001426	AT03	001500	ATN0	001456	ATN1	001460
ATN2	001462	ATN3	001464	ATN4	001466	ATN5	001470
ATN6	001472	ATN7	001474	ATN8	001476	ATNF	001411
C100	000273	CIK	000711	C2	000506	C200	000572
C205	000454	C217	000603	C30K	001273	C400	000274
C77	000357	CNTR	000272	COSF	001274	CIMP	000650
D$22	000361	D2	000440	D3	000425	D4	000427
D5	000411	D6	000446	DA01	000174	DA02	000170
DADD	000155	E$01	000751	E$03	000730	E$04	000770
E$05	001004	E$06	000756	E$07	001012	E$22	000714
ERR	0000000E	EIMP	001021	EX01	001140	EX02	001164
EXP0	001144	EXP1	001146	EXP2	001150	EXP3	001152
EXP4	001154	EXP5	001156	EXP6	001160	EXP7	001162
EXPF	001107	EXPT	000152	F0	001060	F1	001144
FFRH	001366	FHLF	001370	FIN1	000573	FM1	0000000E
FP01	001213	FP02	001205	FP03	001204	FPCR	001220
FPLY	001172	FIMP	001017	H$22	000104	HFP1	001364
HGHC	000450	HIGH	000200	IF01	000622	IFL1	000604
IN01	000665	IN02	000676	IN03	000677	IN04	000663
INTF	000651	K5E5	000044	L$22	000100	LARG	000070
LG01	001102	LG2E	001104	LHIS	000117	LLSM	000713
LMSK	001106	LN2	001166	LOG0	001060	LOG1	001062
LOG2	001064	LOG3	001066	LOG4	001070	LOG5	001072
LOG6	001074	LOG7	001076	LOG8	001100	LOGF	001023
LOW	000201	LOWC	000451	LRS	000360	LRSM	000712
M$11	000507	M$22	000530	M1	0000000E	M17	000527
M2	001503	M227	000707	M31	000356	M36	000710
M4	001272	M7	001165	MDAH	000455	MES	000154
M101	000516	MLMA	000275	MLNN	000276	MSEX	000153
N$22	000122	N2	000211	N3	000244	N4	000243

N6	000266	N7	000270	NORM	000202	PP12	001503
RN03	000042	RND1	000045	RNDF	000022	RSL1	000452
S$22	000277	SPAC	001502	SGNF	000000	SIN0	001350
SIN1	001352	SIN2	001354	SIN3	001356	SIN4	001360
SINF	001303	SN01	001335	SN02	001345	SN03	001362
SQ01	001255	SQRF	001223	TANF	001372	TEMP	000355
T101	000644	TINT	000627	TWP1	001350	UNPK	000134
Z0	001221	Z2	001170				

0000 WARNING OR ERROR FLAGS